Innermost Termination of Context-Sensitive Rewriting

Beatriz Alarcón Salvador Lucas
DSIC, Universidad Politécnica de Valencia
Camino de Vera s/n, E-46022 Valencia, Spain
{balarcon , slucas}@dsic.upv.es

Abstract

Innermost context-sensitive rewriting (CSR) has been proved useful for modeling the computational behavior of programs of algebraic languages like Maude, OBJ, etc, which incorporate an innermost strategy which is used to break down the nondeterminism which is inherent to reduction relations. Furthermore, innermost termination of rewriting is often easier to prove than termination. Thus, under appropriate conditions, a useful strategy for proving termination of rewriting is trying to prove termination of innermost rewriting. This phenomenon has also been investigated for context-sensitive rewriting. Up to now, only few transformation-based methods have been proposed and used to (specifically) prove termination of innermost *CSR*. Powerful and efficient techniques for proving (innermost) termination of (unrestricted) rewriting like the dependency pair framework have not been considered yet. In this work, we investigate the adaptation of the dependency pair framework to innermost *CSR*. We provide a suitable notion of innermost context-sensitive dependency pair and show how to extend and adapt the main notions which conform the framework (chain, termination problem, processor, etc.). Thanks to the innermost context-sensitive dependency pairs, we can now use powerful techniques for proving termination of innermost *CSR*. This is made clear by means of some benchmarks showing that our techniques dramatically improve over previously existing transformational techniques, thus establishing the new state-of-the-art in the area. We have implemented them as part of the termination tool MU-TERM.

1 Introduction

Termination is one of the most interesting practical problems in computation and software engineering. A program or computational system is said to be *terminating* if it does not lead to any infinite computation for any possible call or input data. Ensuring termination is often a prerequisite for essential program properties like correctness. In the last years, many studies have been developed to analyze termination of programming languages, mainly of functional

[Gie95, LJB01, Xi02] and logic programming languages [CLS05, CT99, DD94, DL01, DS02, LMS03, Sma04]. In the case of imperative programming languages, it is becoming important in the last years [AAC+08, BMS05, CPR06, CS02, Tiw04]. Since most computational systems whose operational principle is based on reducing expressions can be described and analyzed by using notions and techniques coming from the abstract model of Term Rewriting Systems (TRSs [BN98, TeR03]), in many programming languages, it is possible to reduce the question of termination of programs to analyze termination of TRSs. For this reason, the development of techniques for proving termination of term rewrite systems becomes especially important since every improvement will have a positive impact on program verification of many programming languages. Following this approach, many powerful studies have been developed for both declarative and imperative programming languages. Regarding with termination of logic programs several works can be found: [AM93, KKS98, Mar94, Mar96, SGN09, SGST06]. Termination of the functional language Haskell [HPW92] has been developed quite recently [GSST06] and also termination of Java Bytecode [OBEG10]. Moreover, such computational systems (e.g., functional, algebraic, and equational programming languages as well as theorem provers based on rewriting techniques) often incorporate a predefined reduction strategy which is used to break down the nondeterminism which is inherent to reduction relations. Eventually, this can rise problems, as each kind of strategy only behaves properly (i.e., it is normalizing, optimal, etc.) for particular classes of programs. One of the most commonly used strategy is the *innermost* one, in which only 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 arguments of symbols) have been used in programming languages such as Clean [NSEP92], Haskell [HPW92], Lisp [McC60], Maude [CDE$^+$07], OBJ2 [FGJM85], OBJ3 [GWM+00], CafeOBJ [FN97], etc., to improve the termination and efficiency of computations. Lazy languages (e.g., Haskell, Clean) interpret them as *strictness annotations* in order to become 'more eager' and efficient. Eager languages (e.g., Lisp, Maude, OBJ2, OBJ3, CafeOBJ) use them as *replacement restrictions* to become 'more lazy' thus (hopefully) avoiding nontermination.

Context-sensitive rewriting (*CSR* [Luc98, Luc02]) is a restriction of rewriting that forbids reductions on some subexpressions and that has proved useful to model and analyze such programming language features at different levels, see, e.g., [BM06, DLM$^+$04, DLM$^+$08, GM04, Luc01b, LM09]. Such a restriction of the rewriting computations is formalized at a very simple syntactic level:

that of the arguments of function symbols f in the signature $\mathcal{F}$. As usual, by a *signature* we mean a set of function symbols $f_1, \ldots, f_n, \ldots$ together with an *arity* function $ar : \mathcal{F} \to \mathbb{N}$ which establishes the number of 'arguments' associated to each symbol. A *replacement map* is a mapping $\mu : \mathcal{F} \to \wp(\mathbb{N})$ satisfying $\mu(f) \subseteq \{1, \ldots, k\}$, for each k-ary symbol f in the signature $\mathcal{F}$ [Luc98]. We use them to discriminate the argument positions on which the rewriting steps are allowed. In *CSR* we only rewrite μ-replacing subterms: every term t (as a whole) is μ-replacing by definition; and t_i (as well as all its μ-replacing subterms) is a μ-replacing subterm of $f(t_1, \ldots, t_k)$ if $i \in \mu(f)$.

Example 1 *Consider the following orthogonal TRS $\mathcal{R}$ which is a variant of an example in [Bor03]:*

$$
\begin{array}{rcl}
\mathtt{from}(x) & \to & \mathtt{cons}(x, \mathtt{from}(\mathtt{s}(x))) \\
\mathtt{sel}(\mathtt{0}, \mathtt{cons}(x, xs)) & \to & x \\
\mathtt{sel}(\mathtt{s}(y), \mathtt{cons}(x, xs)) & \to & \mathtt{sel}(y, xs) \\
\mathtt{minus}(x, \mathtt{0}) & \to & x \\
\mathtt{minus}(\mathtt{s}(x), \mathtt{s}(y)) & \to & \mathtt{minus}(x, y) \\
\mathtt{quot}(\mathtt{0}, \mathtt{s}(y)) & \to & \mathtt{0} \\
\mathtt{quot}(\mathtt{s}(x), \mathtt{s}(y)) & \to & \mathtt{s}(\mathtt{quot}(\mathtt{minus}(x, y), \mathtt{s}(y))) \\
\mathtt{zWquot}(\mathtt{nil}, \mathtt{nil}) & \to & \mathtt{nil} \\
\mathtt{zWquot}(\mathtt{cons}(x, xs), \mathtt{nil}) & \to & \mathtt{nil} \\
\mathtt{zWquot}(\mathtt{nil}, \mathtt{cons}(x, xs)) & \to & \mathtt{nil} \\
\mathtt{zWquot}(\mathtt{cons}(x, xs), \mathtt{cons}(y, ys)) & \to & \mathtt{cons}(\mathtt{quot}(x, y), \mathtt{zWquot}(xs, ys))
\end{array}
$$

together with $\mu(\mathtt{cons}) = \{1\}$ and $\mu(f) = \{1, \ldots, ar(f)\}$ for all other symbols f. According to [GM02a], innermost μ-termination of $\mathcal{R}$ implies its μ-termination as well. We will show how $\mathcal{R}$ can easily be proved innermost μ-terminating (and hence μ-terminating) by using the results in this paper.

The replacement map in Example 1 exemplifies one of the most typical applications of context-sensitive rewriting as a computational mechanism. The declaration $\mu(\mathtt{cons}) = \{1\}$ disallows reductions on the *list* part of the list constructor `cons`, thus making possible a kind of *lazy evaluation* of lists. We can still use projection operators as `sel` to continue the evaluation when needed. The other typical application is the declaration $\mu(\mathtt{if}) = \{1\}$ which allows us to forbid reductions on the two *alternatives* s and t of *if-then-else* expressions $\mathtt{if}(b, s, t)$ whereas it is still possible to perform reductions on the *boolean* part b, as required to implement the usual semantics of the operator.

Termination is also one of the most interesting problems when dealing with *CSR*. With *CSR* we can *achieve* a terminating behavior with nonterminating TRSs by pruning (all) infinite rewrite sequences.

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). Termination of innermost context-sensitive rewriting has been proved useful for proving termination of programs in programming languages like Maude and

OBJ* which permit to control the program execution by means of such context-sensitive annotations [Luc01a, Luc01b]. Techniques for proving termination of innermost *CSR* were first investigated in [GM02b, Luc01a]. These papers, though, only consider *transformational* techniques, where the original CS-TRS $(\mathcal{R}, \mu)$ is transformed into a TRS $\mathcal{R}^{\mu}_{\Theta}$ (where Θ represents the transformation which has been used) whose *innermost* termination implies the innermost termination of *CSR* for $(\mathcal{R}, \mu)$. The *dependency pairs method* [AG00, GAO02, GTS04, GTSF06, HM04, HM05], one of the most powerful techniques for proving termination of rewriting, had not been investigated in connection with proofs of termination of *CSR* until [AGL06]. As shown in [AGL07], proofs of termination using context-sensitive dependency pairs (CSDPs) are much more powerful and faster than any other technique for proving termination of *CSR*. As we show here, dealing with innermost *CSR*, we have a similar situation.

Proving innermost termination of rewriting is often easier than proving termination of rewriting [AG00] and, for some relevant classes of TRSs, innermost termination of rewriting is even equivalent to termination of rewriting [Gra95, Gra96]. In [GM02b, GL02a] it is proved that the equivalence between termination of innermost *CSR* and termination of *CSR* holds in some interesting cases (e.g., for *orthogonal* CS-TRSs).

During the last years, we have investigated in deep how to prove termination of context-sensitive rewriting by using dependency pairs, since they have proven to be one of the most powerful techniques for proving termination of unrestricted rewriting. In [AGL10], we define the notion of context-sensitive dependency pairs following the approach of [HM04] which consists of considering the structure of the infinite rewrite sequences starting from minimal non-terminating terms. Therefore, all the advantages and improvements over this research can also be taken into account in innermost context-sensitive rewriting, improving our previous results on this field in [AL07].

1.1 Plan of the paper

After some preliminaries in Section 2, we develop the material in the paper in three main parts:

1. We investigate the structure of infinite innermost context-sensitive rewrite sequences. This analysis is essential to provide an appropriate definition of innermost context-sensitive dependency pair, and the related notions of innermost chains, graph, etc. Section 3 provides appropriate notions of *minimal* innermost non-μ-terminating terms and introduces the main properties of such terms. It also recalls the notion of *hidden term* in a CS-TRS. This notion turns to be essential for the appropriate treatment of our dependency pairs. We investigate the structure of infinite innermost context-sensitive rewrite sequences starting from strongly minimal innermost non-μ-terminating terms.

2. We define the notions of *innermost context-sensitive dependency pair* and *innermost context-sensitive chain of pairs* and show how to use them to

characterize innermost termination of *CSR*. Sections 4 introduces the general framework to compute and use innermost context-sensitive dependency pairs for proving innermost termination of *CSR*. The introduction of a new kind of dependency pairs (the *collapsing* dependency pairs) leads to a notion of innermost context-sensitive dependency *chain*, which is quite different from the standard one. We prove that our *innermost context-sensitive dependency pair approach* fully characterizes termination of innermost *CSR*.

3. We describe a suitable *framework* for dealing with proofs of termination of innermost *CSR* by using the previous results. Section 5 provides an adaptation of the *dependency pair framework* [GTS04, GTSF06] to innermost *CSR* by defining appropriate notions of *CS problem* and *CS processor* which rely in the notions and results investigated in the second part of the paper. Section 6 introduces several basic processors for proving innermost termination of *CSR*. Section 7 introduces the notion of *innermost context-sensitive (dependency) graph* and the associated CS processor which formalizes the usual practice of analyzing the absence of infinite (minimal) innermost chains by considering the (maximal) cycles in the dependency graph. As in the standard case, the ICS-dependency graph is not computable, so we show how to obtain the *estimated* ICS-dependency graph which is a computable overestimation of it. Section 8 adapts the notion of usable rules to deal with proofs of innermost *CSR* by using term orderings. We introduce the notion of *μ-reduction pair*, which is the straightforward adaptation of reduction pairs used for dealing with dependency pairs in the standard case. Section 9 adapts to the context-sensitive setting, the notion of *usable argument* introduced by Férnandez [Fer05] to prove innermost termination of rewriting by proving termination of *CSR*. In this way, we can prove innermost termination of *CSR* by proving innermost termination of *CSR* using a *more restrictive* replacement map. We also include this criterion as a processor in the innermost context-sensitive dependency pair framework. Section 10 adapts *narrowing transformation* of pairs in [GTSF06] to innermost *CSR* and the new framework.

The paper ends with an experimental evaluation of our techniques in Section 11. Section 12 concludes.

2 Preliminaries

This section collects a number of definitions and notations about term rewriting. More details and missing notions can be found in [BN98, Ohl02, TeR03].

Let A be a set and $\mathsf{R} \subseteq \mathsf{A} \times \mathsf{A}$ be a binary relation on A. We denote the transitive closure of R by R^+ and its reflexive and transitive closure by R^*. We say that R is *terminating* (*strongly normalizing*) if there is no infinite sequence $a_1 \;\mathsf{R}\; a_2 \;\mathsf{R}\; a_3 \cdots$. A reflexive and transitive relation R is a quasi-ordering.

2.1 Signatures, Terms, and Positions

Throughout the paper, $\mathcal{X}$ denotes a countable set of variables and $\mathcal{F}$ denotes a signature, i.e., a set of function symbols $\{\mathtt{f}, \mathtt{g}, \ldots\}$, each having a fixed arity given by a mapping $ar : \mathcal{F} \to \mathbb{N}$. The set of terms built from $\mathcal{F}$ and $\mathcal{X}$ is $\mathcal{T}(\mathcal{F}, \mathcal{X})$. A term is *ground* if it contains no variable. 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. Given positions p, q, we denote their concatenation as $p.q$. Positions are ordered by the standard prefix ordering: $p \leq q$ if $\exists q'$ such that $q = p.q'$ If p is a position, and Q is a set of positions, $p.Q = \{p.q \mid q \in Q\}$. The set of positions of a term t is $\mathcal{P}os(t)$. Positions of nonvariable 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.

We write $t \trianglerighteq s$, read *s is a subterm of t*, if $s = t|_p$ for some $p \in \mathcal{P}os(t)$ and $t \triangleright s$ if $t \trianglerighteq s$ and $t \neq s$. We write $t \ntrianglerighteq s$ and $t \ntriangleright s$ for the negation of the corresponding properties. The symbol labeling the root of t is denoted as $root(t)$. A *context* is a term $C \in \mathcal{T}(\mathcal{F} \cup \{\Box\}, \mathcal{X})$ with a 'hole' $\Box$ (a fresh constant symbol). We write $C[\,]_p$ to denote that there is a (usually single) hole $\Box$ at position p of C. Generally, we write $C[\,]$ to denote an arbitrary context and make explicit the position of the hole only if necessary. $C[\,] = \Box$ is called the *empty* context.

2.2 Substitutions

A substitution is a mapping $\sigma : \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X})$. Denote as ε the 'identity' substitution: $\varepsilon(x) = x$ for all $x \in \mathcal{X}$. The set $\mathcal{D}om(\sigma) = \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$ is called the *domain* of σ.

Remark 1 *In this paper, we do* not *impose that the domain of the substitutions is finite. This is usual practice in the dependency pair approach, where a single substitution is used to instantiate an infinite number of variables coming from renamed versions of the dependency pairs (see below).*

Whenever $\mathcal{D}om(\sigma) \cap \mathcal{D}om(\sigma') = \varnothing$, for substitutions σ, σ', we denote by $\sigma \cup \sigma'$, a substitution such that $(\sigma \cup \sigma')(x) = \sigma(x)$ if $x \in \mathcal{D}om(\sigma)$ and $(\sigma \cup \sigma')(x) = \sigma'(x)$ if $x \in \mathcal{D}om(\sigma')$.

2.3 Renamings and unifiers

A *renaming* is an injective substitution ρ such that $\rho(x) \in \mathcal{X}$ for all $x \in \mathcal{X}$. For renamings, we assume that $\mathcal{V}ar(\rho)$ is finite (which is the usual practice) and also idempotency, i.e., $\rho(\rho(x)) = \rho(x)$ for all $x \in \mathcal{X}$.

The quasi-ordering of subsumption $\leq$ over $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is $t \leq t' \Leftrightarrow \exists \sigma.\ t' = \sigma(t)$. We denote as $\sigma \leq \sigma'$ the fact that $\sigma(x) \leq \sigma'(x)$ for all $x \in \mathcal{X}$, thus extending the quasi-ordering to substitutions.

A substitution σ such that $\sigma(s) = \sigma(t)$ for two terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ is called a *unifier* of s and t; we also say that s and t unify (with substitution σ). If two terms s and t unify, then there is a unique (up to renaming of variables) *most general unifier (mgu)* θ which is *minimal* (w.r.t. the subsumption quasi-ordering $\leq$) among all other unifiers of s and t.

A relation $\mathsf{R} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X}) \times \mathcal{T}(\mathcal{F}, \mathcal{X})$ on terms is *stable* if for all terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, and substitutions σ, we have $\sigma(s) \,\mathsf{R}\, \sigma(t)$ whenever $s \,\mathsf{R}\, t$.

2.4 Rewrite Systems and Term Rewriting

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 rewrite rule $l \to r$ is said to be *collapsing* if $r \in \mathcal{X}$. A *Term Rewriting System* (TRS) is a pair $\mathcal{R} = (\mathcal{F}, R)$, where R is a set of rewrite rules. Given TRSs $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{R}' = (\mathcal{F}', R')$, we let $\mathcal{R} \cup \mathcal{R}'$ be the TRS $(\mathcal{F} \cup \mathcal{F}', R \cup R')$. An instance $\sigma(l)$ of a *lhs* l of a rule is called a *redex*. 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 $\mathcal{D} = \{root(l) \mid l \to r \in R\}$ and $\mathcal{C} = \mathcal{F} - \mathcal{D}$.

Example 2 *Consider again the TRS in Example 1. The symbols* `from`*,* `sel`*,* `minus`*,* `quot` *and* `zWquot` *are defined, and* `s`*,* `0`*,* `cons`*, and* `nil` *are constructors.*

For simplicity, we often write $l \to r \in \mathcal{R}$ instead of $l \to r \in R$ to express that the rule $l \to r$ is a rule of $\mathcal{R}$. The pair $\langle \sigma(l)[\sigma(r')]_p, \sigma(r) \rangle$ is called a critical pair and is also called an overlay if $p = \Lambda$. A critical pair $\langle t, s \rangle$ is trivial if $t = s$. The critical pairs of a TRS $\mathcal{R}$ are the critical pairs between any two of its (renamed) rules; this includes overlaps of a rule with a renamed variant of itself, except at the root, i.e., if $p = \Lambda$. A TRS $\mathcal{R}$ is left-linear if for all $l \to r \in R$, l is a linear term. A left-linear TRS without critical pairs is called orthogonal. 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 $t \to s$, or $t \to_{\mathcal{R}} s$), if $t|_p = \sigma(l)$ and $s = t[\sigma(r)]_p$, for some rule $l \to r \in R$, $p \in \mathcal{P}os(t)$ and substitution σ. We write $t \xrightarrow{>p}_{\mathcal{R}} s$ if $t \xrightarrow{q}_{\mathcal{R}} s$ for some $q > p$. A TRS $\mathcal{R}$ is terminating if its one step rewrite relation $\to_{\mathcal{R}}$ is terminating.

2.5 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 $x \in \mathcal{D}om(\sigma)$. A term $f(t_1, \ldots, t_k)$ is argument normalized if t_i is a normal form for all $1 \leq i \leq n$. An innermost redex is an argument normalized redex. A term s rewrites innermost to t, written $s \to_i t$, if $s \to t$ at position p and $s|_p$ is an innermost redex. Let $\mathcal{R}$ be a TRS. For any symbol f let *Rules*$(\mathcal{R}, f)$ be the set of rules $l \to r$ 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:

$$
\begin{array}{lcl}
\mathbf{U}(\mathcal{R},x) & = & \varnothing \\
\mathbf{U}(\mathcal{R},f(t_1,\ldots,t_n)) & = & \mathit{Rules}(\mathcal{R},f) \cup \bigcup_{i\in ar(f)} \mathbf{U}(\mathcal{R}',t_i) \cup \bigcup_{l\to r\in \mathit{Rules}(\mathcal{R},f)} \mathbf{U}(\mathcal{R}',r)
\end{array}
$$

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

2.6 (Innermost) Context-Sensitive Rewriting

A mapping $\mu : \mathcal{F} \to \wp(\mathbb{N})$ is a *replacement map* (or $\mathcal{F}$-map) if $\forall f \in \mathcal{F},\ \mu(f) \subseteq \{1,\ldots,ar(f)\}$ [Luc98]. Let $M_{\mathcal{F}}$ be the set of all $\mathcal{F}$-maps (or $M_{\mathcal{R}}$ for the $\mathcal{F}$-maps of a TRS $(\mathcal{F}, R)$). Let $\mu_\top$ be the replacement map given by $\mu_\top(f) = \{1,\ldots,ar(f)\}$ for all $f \in \mathcal{F}$ (i.e., no replacement restrictions are specified).

A binary relation R on terms is μ-monotonic if whenever $t \mathrel{\mathsf{R}} s$ we have that $f(t_1,\ldots,t_{i-1},t,\ldots,t_k) \mathrel{\mathsf{R}} f(t_1,\ldots,t_{i-1},s,\ldots,t_k)$ for all $f \in \mathcal{F}$, $i \in \mu(f)$, and $t, s, t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F},\mathcal{X})$. If R is $\mu_\top$-monotonic, we just say that R is *monotonic*.

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 $t \notin \mathcal{X}$. When no replacement map is made explicit, the μ-replacing positions are often called *active*; and the non-μ-replacing ones are often called *frozen*. The following result about *CSR* is often used without any explicit mention.

Proposition 1 [Luc98] *Let $t \in \mathcal{T}(\mathcal{F},\mathcal{X})$ and $p = q.q' \in \mathcal{P}os(t)$. Then $p \in \mathcal{P}os^\mu(t)$ iff $q \in \mathcal{P}os^\mu(t) \wedge q' \in \mathcal{P}os^\mu(t|_q)$*

The μ-replacing subterm relation $\trianglerighteq_\mu$ is given by $t \trianglerighteq_\mu s$ if there is $p \in \mathcal{P}os^\mu(t)$ such that $s = t|_p$. We write $t \rhd_\mu s$ if $t \trianglerighteq_\mu s$ and $t \neq s$. We write $t \rhd_{\not\mu} s$ to denote that s is a non-μ-replacing (hence strict) subterm of t: $t \rhd_{\not\mu} s$ if there is $p \in \mathcal{P}os(t) - \mathcal{P}os^\mu(t)$ such that $s = t|_p$. The set of *μ-replacing* variables of a term t, i.e., variables occurring at some μ-replacing position in t, is $\mathcal{V}ar^\mu(t) = \{x \in \mathcal{V}ar(t) \mid t \trianglerighteq_\mu x\}$. The set of non-$\mu$-replacing variables of t, i.e., variables occurring at some non-μ-replacing position in t, is $\mathcal{V}ar^{\not\mu}(t) = \{x \in \mathcal{V}ar(t) \mid t \rhd_{\not\mu} x\}$. Note that $\mathcal{V}ar^\mu(t)$ and $\mathcal{V}ar^{\not\mu}(t)$ do not need to be disjoint.

A pair $(\mathcal{R},\mu)$ where $\mathcal{R}$ is a TRS and $\mu \in M_{\mathcal{R}}$ is often called a CS-TRS. In *context-sensitive rewriting*, we (only) contract *μ-replacing* redexes: t μ-rewrites to s, written $t \hookrightarrow_\mu s$ (or $t \hookrightarrow_{\mathcal{R},\mu} s$ and even $t \hookrightarrow s$), if $t \xrightarrow{p}_{\mathcal{R}} s$ and $p \in \mathcal{P}os^\mu(t)$.

Example 3 *Consider $\mathcal{R}$ and μ as in Example 1. Then, we have:*

$$\underline{\mathtt{from(0)}} \hookrightarrow_\mu \mathtt{cons(0,\underline{from(s(0))})} \not\hookrightarrow_\mu \mathtt{cons(0,cons(s(0),from(s(s(0)))))}$$

Since the second argument of `cons` *is not μ-replacing, we have that $2 \notin \mathcal{P}os^\mu(\mathtt{cons(0,from(s(0)))})$, and the redex* `from(s(0))` *cannot be μ-rewritten.*

A term t is μ-terminating (or $(\mathcal{R},\mu)$-terminating, if we want an explicit reference to the involved TRS $\mathcal{R}$) if there is no infinite μ-rewrite sequence $t = t_1 \hookrightarrow_\mu t_2 \hookrightarrow_\mu \cdots \hookrightarrow_\mu t_n \hookrightarrow_\mu \cdots$ starting from t. A TRS $\mathcal{R}$ is μ-terminating if $\hookrightarrow_\mu$ is terminating.

A μ-normal form is a term which cannot be μ-rewritten. Let $\mathsf{NF}_\mu(\mathcal{R})$ (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 $x \in \mathcal{D}om(\sigma)$. A term $t = f(t_1, \ldots, t_k)$ 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, i.e., $t = \sigma(l)$ for some substitution σ and rule $l \to r \in \mathcal{R}$ and for all $p \in \mathcal{P}os^\mu(t - \Lambda)$, $t|_p \in \mathsf{NF}_\mu$. A term s innermost μ-rewrites to t, written $s \hookrightarrow_{\mathsf{i}} t$, if $s \xrightarrow{p}_{\mathcal{R}} t$, $p \in \mathcal{P}os^\mu(s)$, and $s|_p$ is a μ-innermost redex. Let innermost μ-rewriting below the root be $\xhookrightarrow{>\Lambda}_{\mathsf{i}} = (\xhookrightarrow{>\Lambda} \cap \hookrightarrow_{\mathsf{i}})$. Termination of *CSR* is fully captured by the so-called μ-reduction orderings, i.e., well-founded, stable orderings $\sqsupset$ which are μ-monotonic. A TRS $\mathcal{R}$ is innermost μ-terminating if $\hookrightarrow_{\mu,\mathsf{i}}$ is terminating. We write $s \xhookrightarrow{!}_{\mathcal{R},\mu,\mathsf{i}} t$ if $s \hookrightarrow^*_{\mathcal{R},\mu,\mathsf{i}} t$ and $t \in \mathsf{NF}_\mu$.

A term t μ-*narrows* to a term s (written $t \leadsto_{\mathcal{R},\mu,\theta} s$), if there is a nonvariable μ-replacing position $p \in \mathcal{P}os^\mu_{\mathcal{F}}(t)$ and a rule $l \to r$ in $\mathcal{R}$ (sharing no variable with t) such that $t|_p$ and l unify with most general unifier θ and $s = \theta(t[r]_p)$.

3 Minimal innermost non-μ-terminating terms and Infinite Innermost μ-rewrite Sequences

In the following, we show how to adapt our results about the structure of infinite context-sensitive rewrite sequences [AGL10, Section 3] to the innermost sequences. Most proofs are only slightly different from the original ones and therefore we comment on the differences only (for full proofs see [Ala08]). Major differences come from particularities of reductions under an innermost strategy. In some cases, they bring us some advantages over the case of 'free' reductions in *CSR*. In the following we discuss some of these peculiarities. In the innermost (context-sentive) setting, matching substitutions are always (μ-)normalized. According to the discussion in [AGL10], we introduce the following:

Definition 1 ((Strongly) minimal innermost non-μ-terminating term) *Let $\mathcal{M}_{\infty,\mu,\mathsf{i}}$ be the set of minimal innermost non-μ-terminating terms in the following sense: t belongs to $\mathcal{M}_{\infty,\mu,\mathsf{i}}$ if t is not innermost μ-terminating and every strict* μ-replacing *subterm s of t (i.e., $t \rhd_\mu s$) is innermost μ-terminating. Let $\mathcal{T}_{\infty,\mu,\mathsf{i}}$ be a set of strongly minimal innermost non-μ-terminating terms in the following sense: t belongs to $\mathcal{T}_{\infty,\mu,\mathsf{i}}$ if t is innermost non-μ-terminating and every strict subterm u (i.e., $t \rhd u$) is innermost μ-terminating. It is obvious that $root(t) \in \mathcal{D}$ for all $t \in \mathcal{T}_{\infty,\mu,\mathsf{i}}$ or $t \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$.*

Note that $\mathcal{T}_{\infty,\mu,\mathsf{i}} \subseteq \mathcal{M}_{\infty,\mu,\mathsf{i}}$. Before starting our discussion about minimal innermost non-μ-terminating terms, we provide auxiliary results about innermost μ-terminating terms (see [AGL10, Lemmata 1,2,3,4]).

Proposition 2 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, and $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$.*

1. *If s is innermost μ-terminating and $s \trianglerighteq_\mu t$ or $s \hookrightarrow^*_{\mathcal{R},\mu,\mathsf{i}} t$ then t is innermost μ-terminating.*

2. *If s is not innermost μ-terminating, then there is a subterm t of s ($s \trianglerighteq t$) such that $t \in \mathcal{T}_{\infty,\mu,\mathsf{i}}$. Furthermore, there is a μ-replacing subterm t of s ($s \trianglerighteq_\mu t$) such that $t \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$.*

3. *If $t \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$, $t \overset{>\Lambda}{\hookrightarrow}{}^*_{\mathsf{i}}\, u$ and u is not innermost μ-terminating, then $u \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$.*

The following result is the innermost context-sensitive version of Lemma 1 in [HM04] that uses previous results. This proposition establishes that, given a minimal not innermost μ-terminating term $t \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$, there are only two ways for an infinite innermost μ-rewrite sequence to proceed. The first one is by using 'visible' parts of the rules which correspond to μ-replacing nonvariable subterms in the right-hand sides which are rooted by a defined symbol. This would corresponds with the straightforward extension of the original result but taking into account the reemplacement restrictions. The second one is by showing up 'hidden' not innermost μ-terminating subterms which are activated by *migrating* variables in a rule $l \to r$, i.e., variables $x \in \mathcal{V}ar^\mu(r) \setminus \mathcal{V}ar^\mu(l)$ which are *not* μ-replacing in the left-hand side l but become μ-replacing in the right-hand side r.

Proposition 3 *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. Then for all $t \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$, there exist $l \to r \in R$, a substitution σ such that $\sigma(l)$ is argument μ-normalized and a term $u \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$ such that $t \overset{>\Lambda}{\hookrightarrow}{}^*_{\mathsf{i}}\, \sigma(l) \overset{\Lambda}{\hookrightarrow}_{\mathsf{i}} \sigma(r) \trianglerighteq_\mu u$ and either*

1. *there is a nonvariable μ-replacing subterm s of r, $r \trianglerighteq_\mu s$, such that $u = \sigma(s)$ and $\sigma(x) \in \mathsf{NF}_\mu(\mathcal{R})$ for all $x \in \mathcal{V}ar(s) \cap \mathcal{V}ar^\mu(l)$, or*

2. *there is $x \in \mathcal{V}ar^\mu(r) \setminus \mathcal{V}ar^\mu(l)$ such that $\sigma(x) \trianglerighteq_\mu u$, that is, $\sigma(x) = C[u]_p$ for some context $C[]_p$ with $p \in Pos^\mu(C[]_p)$.*

Proof.

Consider an infinite innermost μ-rewrite sequence starting from t. By definition of $\mathcal{M}_{\infty,\mu,\mathsf{i}}$, all proper μ-replacing subterms of t are innermost μ-terminating. Therefore, t has an inner reduction (of innermost μ-rewriting steps) to an instance $\sigma(l)$ of the left-hand side of a rule $l \to r$ of $\mathcal{R}$, such that no strict μ-replacing subterm of $\sigma(l)$ is a redex, i.e. $\sigma(l)$ is argument μ-normalized. Then we have $t \overset{>\Lambda}{\hookrightarrow}{}^*_{\mathsf{i}}\, \sigma(l) \overset{\Lambda}{\hookrightarrow}_{\mathsf{i}} \sigma(r)$ and $\sigma(r)$ is not innermost μ-terminating. Note that, $\sigma(l)$ must be argument μ-normalized; otherwise, the last step would not be an innermost μ-rewriting step. Thus, we can write $t = f(t_1, \ldots, t_k)$ and $\sigma(l) = f(l_1, \ldots, l_k)$ for some k-ary defined symbol f, and $t_i \hookrightarrow^*_{\mathsf{i}} \sigma(l_i)$ for all i, $1 \leq i \leq k$. More precisely, $t_i \overset{!}{\hookrightarrow}_{\mathsf{i}} \sigma(l_i)$ if $i \in \mu(f)$ Since $\sigma(l)$ is argument μ-normalized, $\sigma(x) \in \mathsf{NF}_\mu$ for all μ-replacing variables x in l: $x \in \mathcal{V}ar^\mu(l)$. Since $\sigma(r)$ is not innermost μ-terminating, by Proposition 2-2 it contains a μ-replacing subterm $u \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$: $\sigma(r) \trianglerighteq_\mu u$, i.e., there is a position $p \in \mathcal{P}os^\mu(\sigma(r))$ such that $\sigma(r)|_p = u$. We consider two cases:

1. If $p \in \mathcal{P}os_{\mathcal{F}}(r)$ is a nonvariable position of r, then there is a μ-replacing subterm s of r, such that $u = \sigma(s)$. Note that $\sigma(x) \in \mathsf{NF}_\mu$ for all $x \in \mathcal{V}ar(r) \setminus \mathcal{V}ar^\mu(l)$.

2. If $p \notin \mathcal{P}os_{\mathcal{F}}(r)$, then there is a μ-replacing variable position $q \in \mathcal{P}os^\mu(r) \cap \mathcal{P}os_{\mathcal{X}}(r)$ such that $q \leq p$. Let $x \in \mathcal{V}ar^\mu(r)$ be such that $r|_q = x$. Then, $\sigma(x) \trianglerighteq_\mu u$ and $\sigma(x)$ is not innermost μ-terminating (by assumption, $u \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$ is not innermost μ-terminating: by Proposition 2-1, $\sigma(x)$ cannot be innermost μ-terminating either). Since $\sigma(l_i)$ is innermost μ-terminating for all $i \in \mu(f)$, and $\sigma(x) \in \mathsf{NF}_\mu$ for all μ-replacing variables in l , we conclude that $x \in \mathcal{V}ar^\mu(r) \setminus \mathcal{V}ar^\mu(l)$.

$\square$

Proposition 3 entails the following result, which establishes some properties of infinite sequences starting from minimal innermost non-μ-terminating terms.

Corollary 1 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. For all $t \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$, there is an infinite sequence*

$$t \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \sigma_1(l_1) \overset{\Lambda}{\hookrightarrow}_{\mathsf{i}} \sigma_1(r_1) \trianglerighteq_\mu t_1 \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \sigma_2(l_2) \overset{\Lambda}{\hookrightarrow}_{\mathsf{i}} \sigma_2(r_2) \trianglerighteq_\mu t_2 \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \cdots$$

where, for all $i \geq 1$, $l_i \to r_i \in R$ are rewrite rules, σ_i are substitutions, $\sigma_i(l_i)$ is argument μ-normalized, and terms $t_i \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$ are minimal innermost non-μ-terminating terms such that either

1. *$t_i = \sigma_i(s_i)$ for some nonvariable subterm s_i such that $r_i \trianglerighteq_\mu s_i$ and $\sigma(x) \in \mathsf{NF}_\mu(\mathcal{R})$ for all $x \in \mathcal{V}ar(s_i) \cap \mathcal{V}ar^\mu(l_i)$, or*

2. *$\sigma_i(x_i) \trianglerighteq_\mu t_i$ which is equivalent to $\sigma(x) = C[t_i]_{p_i}$ for some $x_i \in \mathcal{V}ar^\mu(r_i) \setminus \mathcal{V}ar^\mu(l_i)$ and context $C[]_{p_i}$ with $p_i \in Pos^\mu(C[]_{p_i})$.*

Now we pay attention to Item 2 of Proposition 3. To analyze in deep infinite sequences starting from minimal innermost non-μ-terminating terms we need to go inside the instantiation of the migrating variable, $\sigma(x)$. Since in (innermost) context-sensitive rewriting, function calls can be delayed, terms that are (innermost) μ-terminating can generate future (innermost) non-μ-terminating subterms. By Lemma 2 we know that innermost μ-termination is preserved under μ-rewritings and extraction of μ-replacing subterms, therefore, these innermost non-μ-terminating subterms introduced by innermost μ-rewriting steps can only occur at frozen positions in the reducts. This is captured by the notion of *hidden term*.

Definition 2 (Hidden Term [AGL10]) *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 $l \to r \in R$ such that $r \rhd_{\not\mu} t$. 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 $(\mathcal{R}, \mu)$. We say that $f \in \mathcal{F}$ is a* hidden symbol *if it occurs in a hidden term. Let $\mathcal{H}(\mathcal{R}, \mu)$ (or just $\mathcal{H}$) be the set of all hidden symbols in $(\mathcal{R}, \mu)$. We also use $\mathcal{DHT}(\mathcal{R}, \mu) = \{t \in \mathcal{HT} \mid root(t) \in \mathcal{D}\}$ for the set of hidden terms which are rooted by a* defined *symbol.*

Example 4 *For $\mathcal{R}$ and μ as in Example 1, the hidden terms are* $\mathtt{from}(\mathtt{s}(x))$*,* $\mathtt{s}(x)$*, and* $\mathtt{zWquot}(zs, ys)$*. The hidden symbols are* from*,* s *and* zWquot*. Finally,* $\mathcal{DHT}(\mathcal{R}, \mu) = \{\mathtt{from}(\mathtt{s}(x)), \mathtt{zWquot}(zs, ys)\}$.

Innermost non-μ-terminating terms at frozen positions can be activated by some specific contexts. In Proposition 3 (2), the intended role of hidden terms in the binding of the migrating variable $\sigma(x) = C[u]_p$ is that u' is a hidden term such that $\theta(u') = u$ for some substitution θ and context $C[]_p$. This context can only be composed by symbols f contained in hidden terms $f(\ldots, r_i, \ldots)$ such that $r' \rhd_{\not\mu} f(\ldots, r_i, \ldots) \unrhd_\mu r_i$ for a rule $l' \to r' \in \mathcal{R}$ satisfying:

- r_i is a nonvariable term and $\sigma(r_i) = u$, or
- r_i is a variable at a frozen position in both, l and r.

These symbols conforms what is called as hiding context.

First notion of hiding context was found in [AEF+08] but it has been recently slightly redefined in [GL10]. We follow the last definition since it present some advantages.

Definition 3 (Hiding Context [GL10]) *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. A function symbol f* hides position *$i \in \mu(f)$ in the rule $l \to r \in \mathcal{R}$ if $r \rhd_{\not\mu} f(r_1, \ldots, r_n)$ for some terms $r_1, \ldots, r_n$, and r_i contains a μ-replacing defined symbol (i.e., $\mathcal{P}os^\mu_{\mathcal{D}}(r_i) \neq \varnothing$) or a variable $x \in (\mathcal{V}ar^{\not\mu}(l) \cap \mathcal{V}ar^{\not\mu}(r)) \setminus (\mathcal{V}ar^\mu(l) \cup \mathcal{V}ar^\mu(r))$ which is μ-replacing in r_i (i.e., $x \in \mathcal{V}ar^\mu(r_i)$). We say that f hides position i in $\mathcal{R}$ if there is a rule $l \to r$ such that f hides position i in $l \to r$. A context $C[\Box]$ is* hiding *if*

1. *$C[\Box] = \Box$, or*
2. *$C[\Box] = f(t_1, \ldots, t_{i-1}, C'[\Box], t_{i+1}, \ldots, t_k)$, where f hides position i and $C'[\Box]$ is a hiding context.*

These notions are used and combined to model infinite context-sensitive rewrite sequences starting from strongly minimal innermost non-μ-terminating, although, first, we need some previous results.

Definition 4 (Hiding Property [AEF+08]) *A term u has the* hiding property *iff*

- *$u \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$ and*
- *whenever $u \rhd_{\not\mu} s \unrhd_\mu t'$ for some terms s and t' with $t' \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$, then t' is an instance of a hidden term and $s = C[t']$ for some hiding context $C[\,]$.*

Lemma 1 ([AEF+08]) *Let u be a term with the hiding property and let $u \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} v \unrhd_\mu w$ with $w \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$. Then w also has the hiding property.*

The proof of the previous lemma differs from the one in [AEF+08] in the refinement done in the notion of hiding context mentioned in [GL10] and it is slightly different from the one in [GL10] since we are dealing with innermost rewriting and all μ-replacing variables of the instantiated left-hand sides of the rules applied in a innermost μ-rewrite sequence are in μ-normal form: no matter if they are in a nonactive position on the right-hand side, they cannot start any reduction. In [GL10] it is not necessary either since in a μ-rewrite sequence, these variables could start a reduction but due to minimality, these reductions would be finite.

In the following, we consider a function $\textsc{Ren}^{\mu}$ [AGL06, AGL10] which *independently* renames all *occurrences* of μ-replacing variables within a term t by using new fresh variables which are not in $\mathcal{V}ar(t)$. Note that $\textsc{Ren}^{\mu}(t)$ keeps variables at non-μ-replacing positions untouched.

Proposition 4 ([AGL10]) *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) - \mathcal{X}$ be a nonvariable term and σ be a substitution. If $\sigma(t) \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \sigma(l)$ for some (probably renamed) rule $l \to r \in R$, then $\textsc{Ren}^{\mu}(t)$ is μ-narrowable.*

Corollary 2 ([AGL10]) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) - \mathcal{X}$ be a nonvariable term and σ be a substitution such that $\sigma(t) \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$. Then, $\textsc{Ren}^{\mu}(t)$ is μ-narrowable.*

In the following, we write $\textsc{Narr}^{\mu}(t)$ [AGL10] to indicate that t is μ-narrowable (w.r.t. the intended TRS $\mathcal{R}$). We also let

$$\mathcal{NHT}(\mathcal{R}, \mu) = \{t \in \mathcal{DHT} \mid \textsc{Narr}^{\mu}(\textsc{Ren}^{\mu}(t))\}$$

be the set of *hidden terms* which are rooted by a *defined* symbol, and that, after applying $\textsc{Ren}^{\mu}$, become *μ-narrowable.*

As a consequence of the previous results, we have the following main result.

Theorem 1 (Minimal Innermost Sequence) *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. For all $t \in \mathcal{T}_{\infty,\mu,\mathsf{i}}$, there is an infinite sequence*

$$t = t_0 \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \sigma_1(l_1) \overset{\Lambda}{\hookrightarrow}_{\mathsf{i}} \sigma_1(r_1) \trianglerighteq_{\mu} t_1 \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \sigma_2(l_2) \overset{\Lambda}{\hookrightarrow}_{\mathsf{i}} \sigma_2(r_2) \trianglerighteq_{\mu} t_2 \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \cdots$$

where, for all $i \geq 1$, $l_i \to r_i \in R$, σ_i is a substitution, $\sigma_i(l_i)$ is argument μ-normalized, and $t_i \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$ is a minimal innermost non-μ-terminating term such that either

1. *$t_i = \sigma_i(s_i)$ for some nonvariable term s_i such that $r_i \trianglerighteq_{\mu} s_i$, or*
2. *$\sigma_i(x_i) = \theta_i(C_i[t'_i])$ and $t_i = \theta_i(t'_i)$ for some variable $x_i \in \mathcal{V}ar^{\mu}(r_i) \setminus \mathcal{V}ar^{\mu}(l_i)$, $t'_i \in \mathcal{NHT}(\mathcal{R}, \mu)$, hiding context $C_i[\Box]$, and substitution θ_i.*

Proof.

Since $\mathcal{T}_{\infty,\mu,\mathsf{i}} \subseteq \mathcal{M}_{\infty,\mu,\mathsf{i}}$, by Corollary 1, we have a sequence

$$t = t_0 \xhookrightarrow{>\Lambda}_{\mathsf{i}}^{*} \sigma_1(l_1) \xhookrightarrow{\Lambda}_{\mathsf{i}} \sigma_1(r_1) \trianglerighteq_\mu t_1 \xhookrightarrow{>\Lambda}_{\mathsf{i}}^{*} \sigma_2(l_2) \xhookrightarrow{\Lambda}_{\mathsf{i}} \sigma_2(r_2) \trianglerighteq_\mu t_2 \xhookrightarrow{>\Lambda}_{\mathsf{i}}^{*} \cdots$$

where, for all $i \geq 1$, $l_i \to r_i \in R$, σ_i is a substitution such that $\sigma(l_i)$ is argument μ-normalized, $t_i \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$, and either (1) $t_i = \sigma_i(s_i)$ for some s_i such that $r_i \trianglerighteq_\mu s_i$ or (2) $\sigma_i(x_i) \trianglerighteq_\mu t_i$ for some $x_i \in \mathcal{V}ar^\mu(r_i) \setminus \mathcal{V}ar^\mu(l_i)$ (and hence $\sigma(l_i) \rhd_{\not\mu} t_i$ and $\sigma(r_i) \trianglerighteq_\mu t_i$ as well). If $\sigma_i(x_i) \trianglerighteq_\mu t_i$ for some $x_i \in \mathcal{V}ar^\mu(r_i) \setminus \mathcal{V}ar^\mu(l_i)$, it means that $\sigma(l_i) \rhd_{\not\mu} C_i[t_i]$. Since $t \in \mathcal{T}_{\infty,\mu,\mathsf{i}}$, it has the hiding property and, by Lemma 1, all $\sigma(l_i)$ satisfies the *hiding* property. Hence, $C_i[t_i] = \theta_i(C'_i[t'_i])$ where $t'_i \in \mathcal{DHT}(\mathcal{R}, \mu)$ and $C'_i[\,]$ is a hiding context. By Corollary 2 we have $t'_i \in \mathcal{NHT}$. □

4 Innermost Context-Sensitive Dependency Pairs and Chains

An essential property of the dependency pairs method is that it provides a *characterization* of termination of TRSs $\mathcal{R}$ as the absence of infinite (minimal) *chains of dependency pairs* [AG00, GTSF06]. As we prove here this is also true for innermost *CSR*. First, we have to introduce a suitable notion of innermost dependency pair and chain which can be used for this purpose.

In innermost *CSR*, we only perform reduction steps on *innermost μ-replacing redexes*. Therefore, we have to restrict the definition of chains in order to obtain an appropriate notion corresponding to innermost *CSR*, which, obviously, is an adaptation of the one for standard *CSR*(see [AGL10]). Regarding innermost reductions, arguments of a redex should be in *normal form* before the redex is contracted and, regarding *CSR*, the redex to be contracted has to be in a *μ-replacing* position.

Given a signature $\mathcal{F}$ and $f \in \mathcal{F}$, we let $f^\sharp$ be a new fresh symbol (often called *tuple* symbol or DP-symbol) associated to a symbol f [AG00]. Let $\mathcal{F}^\sharp$ be the set of tuple symbols associated to symbols in $\mathcal{F}$. As usual, for $t = f(t_1, \ldots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, we write $t^\sharp$ to denote the *marked* term $f^\sharp(t_1, \ldots, t_k)$. Conversely, given a marked term $t = f^\sharp(t_1, \ldots, t_k)$, where $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, we write $t^\natural$ to denote the term $f(t_1, \ldots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Let $\mathcal{T}^\sharp(\mathcal{F}, \mathcal{X}) = \{t^\sharp \mid t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) - \mathcal{X}\}$ be the set of marked terms.

Definition 5 (Innermost Context-Sensitive Dependency Pairs) *Let* $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ *be a TRS and* $\mu \in M_\mathcal{F}$. *We define* $\mathsf{iDP}(\mathcal{R}, \mu) = \mathsf{iDP}_\mathcal{F}(\mathcal{R}, \mu) \cup \mathsf{iDP}_\mathcal{X}(\mathcal{R}, \mu)$ *to be the set of* innermost context-sensitive *dependency pairs (ICS-DPs) where:*

$$\begin{array}{lcl} \mathsf{iDP}_\mathcal{F}(\mathcal{R}, \mu) & = & \{l^\sharp \to s^\sharp \mid l \to r \in R, l^\sharp \in \mathsf{NF}_\mu(\mathcal{R}), r \trianglerighteq_\mu s, root(s) \in \mathcal{D}, l \ntrianglerighteq_\mu s, \textsc{Narr}^\mu(\textsc{Ren}^\mu(s))\} \\ \mathsf{iDP}_\mathcal{X}(\mathcal{R}, \mu) & = & \{l^\sharp \to x \mid l \to r \in R, l^\sharp \in \mathsf{NF}_\mu(\mathcal{R}), x \in \mathcal{V}ar^\mu(r) \setminus \mathcal{V}ar^\mu(l)\} \end{array}$$

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* $f \in \mathcal{D}$.

Example 5 *Consider Example 1. The set* $\mathsf{iDP}(\mathcal{R},\mu)$ *consists of the following pairs:*

$$\begin{array}{rcll}
\mathtt{MINUS}(\mathtt{s}(x),\mathtt{s}(y)) & \to & \mathtt{MINUS}(x,y) & (1)\\
\mathtt{QUOT}(\mathtt{s}(x),s(y)) & \to & \mathtt{MINUS}(x,y) & (2)\\
\mathtt{QUOT}(\mathtt{s}(x),s(y)) & \to & \mathtt{QUOT}(\mathtt{minus}(x,y),\mathtt{s}(y)) & (3)\\
\mathtt{SEL}(\mathtt{s}(y),\mathtt{cons}(x,xs)) & \to & \mathtt{SEL}(y,xs) & (4)\\
\mathtt{SEL}(\mathtt{s}(y),\mathtt{cons}(x,xs)) & \to & xs & (5)\\
\mathtt{ZWQUOT}(\mathtt{cons}(x,xs),\mathtt{cons}(y,ys)) & \to & \mathtt{QUOT}(x,y) & (6)
\end{array}$$

The ICSDPs $u \to v \in \mathsf{iDP}_{\mathcal{X}}(\mathcal{R},\mu)$ in Definition 5, consisting of collapsing rules only, are called the *collapsing* ICSDPs. A rule $l \to r$ of a TRS $\mathcal{R}$ is μ-conservative if $\mathcal{V}ar^{\mu}(r) \subseteq \mathcal{V}ar^{\mu}(l)$, i.e., it does not contain migrating variables; $\mathcal{R}$ is μ-conservative if all its rules are (see [Luc96, Luc06]).

Clearly, If $\mathcal{R}$ is μ-conservative, then $\mathsf{iDP}(\mathcal{R},\mu) = \mathsf{iDP}_{\mathcal{F}}(\mathcal{R},\mu)$.

Therefore, in order to deal with μ-conservative TRSs $\mathcal{R}$ we only need to consider the 'classical' dependency pairs in $\mathsf{iDP}_{\mathcal{F}}(\mathcal{R},\mu)$. If the TRS $\mathcal{R}$ contains non-μ-conservative rules, then we also need to consider dependency pairs with variables in the right-hand side.

To deal with the information corresponding to hidden terms and hiding contexts when trying to characterize innermost μ-termination with ICSDPs, we use an *unhiding TRS* $\mathsf{unh}(\mathcal{R},\mu)$. This unhiding TRS captures the situation described in Theorem 1 when managing migrating variables. According to this, we have to remove the (instance of the) hiding context $C_i[]$ to extract the delayed call t_i and then connect this delayed call, which is an instance $\theta(t'_i)$ of a hidden term t'_i with the next pair in the innermost μ-chain. We perform these two actions by using two kind of rewrite rules:

- If $\theta(Ci[t'_i]) = \theta(f(t_1,\ldots,t_{i-1},C'_i[t'_i],t_{i+1},\ldots,t_k))$ then, since $C_i[]$ is a hiding context, f hides position i and $C'_i[]$ is a hiding context as well. Then, we can extract $\theta(C'_i[t'_i])$ from $\theta(C_i[t'_i])$ by using the following projection rule: $f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_k) \to x_i$
- Once t_i has been reached, we know that it is an instance $t_i = \theta(t'_i)$ of a nonvariable hidden term $t'_i \in \mathcal{NHT}(\mathcal{R},\mu)$ and we have to connect t_i with the next innermost context-sensitive dependency pair. Since the root of the innermost context-sensitive dependency pair is a marked symbol, we can do it by using a rule that just changes the root symbol by its marked version in the following way: $t'_i \to t'^{\sharp}_i$

Definition 6 (Unhiding TRS [GL10]) *Let* $\mathcal{R}$ *be a TRS and* $\mu \in M_{\mathcal{R}}$*. We define* $\mathsf{unh}(\mathcal{R},\mu)$ *as the TRS consisting of the following rules:*

1. $f(x_1,\ldots,x_i,\ldots,x_k) \to x_i$ *for all function symbols* f *of arity* k*, distinct variables* $x_1,\ldots,x_k$*, and* $1 \leq i \leq k$ *such that* f hides position i *in* $l \to r \in R$*, and*

2. $t \to t^\sharp$ *for every* $t \in \mathcal{NHT}(\mathcal{R}, \mu)$.

Example 6 *The unhiding TRS* $\mathsf{unh}(\mathcal{R}, \mu)$ *for* $\mathcal{R}$ *and* μ *in Example 1 is:*

$$\begin{array}{rcl}
\mathtt{from}(x) & \to & \mathtt{FROM}(x) \\
\mathtt{zWquot}(x, y) & \to & \mathtt{ZWQUOT}(x, y) \\
\mathtt{zWquot}(x, y) & \to & x \\
\mathtt{zWquot}(x, y) & \to & y
\end{array}$$

Definitions 5 and 6 lead to a suitable notion of *chain* which captures minimal infinite μ-rewrite sequences according to the description in Theorem 1. In the following, given a TRS $\mathcal{S}$, we let $\mathcal{S}_{\rhd_\mu}$ be the rules from $\mathcal{S}$ of the form $s \to t \in \mathcal{S}$ and $s \rhd_\mu t$ (Definition 6-1); and $\mathcal{S}_\sharp = \mathcal{S} \setminus \mathcal{S}_{\rhd_\mu}$ (Definition 6-2).

As in the DP-framework [GTS04, GTSF06], where the procedence of *pairs* does not matter, we rather think of another TRS $\mathcal{P}$ which is used together with $\mathcal{R}$ to build the chains. Once this more abstract notion of chain is introduced, it can be particularized to be used with ICSDPs, by just taking $\mathcal{P} = \mathsf{iDP}(\mathcal{R}, \mu)$.

Definition 7 ((Minimal) Innermost μ-Chain) *Let* $\mathcal{R}$*,* $\mathcal{P}$ *and* $\mathcal{S}$ *be TRSs and* $\mu \in M_{\mathcal{R}\cup\mathcal{P}\cup\mathcal{S}}$*. An innermost* $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$*-chain is a finite or infinite sequence of pairs* $u_i \to v_i \in \mathcal{P}$*, together with a substitution* σ *satisfying that, for all* $i \geq 1$*,* $\sigma(u_i) \in \mathsf{NF}_\mu(\mathcal{R})$ *and :*

1. *if* $v_i \notin \mathcal{V}ar(u_i) \setminus \mathcal{V}ar^\mu(u_i)$*, then* $\sigma(v_i) = t_i \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} \sigma(u_{i+1})$*, and*

2. *if* $v_i \in \mathcal{V}ar(u_i) \setminus \mathcal{V}ar^\mu(u_i)$*, then* $\sigma(v_i) \overset{\Lambda}{\hookrightarrow}^*_{\mathcal{S}_{\rhd_\mu},\mu} \circ \overset{\Lambda}{\hookrightarrow}_{\mathcal{S}_\sharp,\mu} t_i \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} \sigma(u_{i+1})$*.*

An innermost $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$*-chain is called* minimal *if for all* $i \geq 1$*,* t_i *is innermost* $(\mathcal{R}, \mu)$*-terminating.*

Note that the condition $v_i \in \mathcal{V}ar(u_i) \setminus \mathcal{V}ar^\mu(u_i)$ in Definition 8 implies that v_i is a variable. Furthermore, since each $u_i \to v_i \in \mathcal{P}$ is a rewrite rule (i.e., $\mathcal{V}ar(v_i) \subseteq \mathcal{V}ar(u_i)$), v_i is a *migrating variable* in the rule $u_i \to v_i$.

In the following, the pairs in a CS-TRS $(\mathcal{P}, \mu)$, where $\mathcal{P} = (\mathcal{G}, P)$, are partitioned according to its role in Definition 8 as follows:

$$P_\mathcal{X} = \{u \to v \in P \mid v \in \mathcal{V}ar(u) \setminus \mathcal{V}ar^\mu(u)\} \text{ and } P_\mathcal{G} = P - P_\mathcal{X}$$

Despite this fact, we refer to $\mathcal{P}_\mathcal{X}$ as the set of *collapsing* pairs in $\mathcal{P}$ because its intended role in Definition 8 is capturing the computational behavior of collapsing ICSDPs in $\mathsf{iDP}_\mathcal{X}(\mathcal{R}, \mu)$.

The following result establishes the soundness of the innermost context-sensitive dependency pair approach. As usual, in order to fit the requirement of variable-disjointness among two arbitrary pairs in a chain of pairs, we assume that appropriately renamed ICSDPs are available when necessary.

Theorem 2 (Soundness) *Let* $\mathcal{R}$ *be a TRS and* $\mu \in M_\mathcal{R}$*. If there is no infinite minimal innermost* $(\mathsf{iDP}(\mathcal{R}, \mu), \mathcal{R}, \mathsf{unh}(\mathcal{R}, \mu), \mu^\sharp, \mathbf{i})$*-chain, then* $\mathcal{R}$ *is innermost* μ*-terminating.*

Proof.

By contradiction. If $\mathcal{R}$ is not innermost μ-terminating, then by Proposition 2 (2) there is $t \in \mathcal{T}_{\infty,\mu,\mathsf{i}}$. By Theorem 1, there are rules $l_i \to r_i \in \mathcal{R}$, matching substitutions σ_i, such that $\sigma_i(l_i)$ is argument μ-normalized and terms $t_i \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$, for $i \geq 1$ such that

$$t = t_0 \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \sigma_1(l_1) \overset{\Lambda}{\hookrightarrow}_{\mathsf{i}} \sigma_1(r_1) \trianglerighteq_\mu t_1 \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \sigma_2(l_2) \overset{\Lambda}{\hookrightarrow}_{\mathsf{i}} \sigma_2(r_2) \trianglerighteq_\mu t_2 \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \cdots$$

where either (D1) $t_i = \sigma_i(s_i)$ for some s_i such that $r_i \trianglerighteq_\mu s_i$ or (D2) $\sigma_i(x_i) = C_i[t_i]$ for some $x_i \in \mathcal{V}ar^\mu(r_i) \setminus \mathcal{V}ar^\mu(\ell_i)$ and $C_i[t_i] = \theta_i(C'_i[t'_i])$ for some $t'_i \in \mathcal{NHT}$ and hiding context $C'_i[\Box]$. Furthermore, since $t_{i-1} \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \sigma_i(l_i)$ and $t_{i-1} \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$ (in particular, $t_0 = t \in \mathcal{T}_{\infty,\mu,\mathsf{i}} \subseteq \mathcal{M}_{\infty,\mu,\mathsf{i}}$), by Proposition 2 (3), $\sigma_i(l_i) \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$ for all $i \geq 1$. Note that, since $t_i \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$, we have that $t_i^\sharp$ is innermost μ-terminating (with respect to $\mathcal{R}$), because all μ-replacing subterms of t_i (hence of $t_i^\sharp$ as well) are innermost μ-terminating and $root(t^\sharp)$ is not a defined symbol of $\mathcal{R}$.

First, note that $\mathsf{iDP}(\mathcal{R},\mu)$ is a TRS $\mathcal{P}$ over the signature $\mathcal{G} = \mathcal{F} \cup \mathcal{D}^\sharp$ and $\mu^\sharp \in M_{\mathcal{F}\cup\mathcal{G}}$ as required by Definition 8. Furthermore, $\mathcal{P}_\mathcal{G} = \mathsf{iDP}_\mathcal{F}(\mathcal{R},\mu)$ and $\mathcal{P}_\mathcal{X} = \mathsf{iDP}_\mathcal{X}(\mathcal{R},\mu)$. We can define an infinite minimal innermost $(\mathsf{iDP}(\mathcal{R},\mu), \mathcal{R}, \mathsf{unh}(\mathcal{R},\mu), \mu^\sharp, \mathbf{i})$-chain using ICSDPs $u_i \to v_i$ for $i \geq 1$, where $u_i = l_i^\sharp$, and

1. $v_i = s_i^\sharp$ if (D1) holds. Since $t_i \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$, we have that $root(s_i) \in \mathcal{D}$ and, because $t_i = \sigma_i(s_i)$ and $\sigma_i(s_i) \hookrightarrow^*_{\mathsf{i}} \sigma_{i+1}(\ell_{i+1})$, by Corollary 2 $\textsc{Ren}^\mu(s_i)$ is μ-narrowable. Furthermore, if we assume that s_i is a μ-replacing subterm of l_i (i.e., $l_i \triangleright_\mu s_i$), then $\sigma_i(l_i) \triangleright_\mu \sigma_i(s_i)$ which (since $\sigma_i(s_i) = t_i \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$) contradicts that $\sigma_i(l_i) \in \mathcal{M}_{\infty,\mu,\mathsf{i}}$. Thus, $l_i \not\triangleright_\mu s_i$. Moreover, since $\sigma_i(l_i)$ is argument μ-normalized, it implies that $\sigma_i(l_i^\sharp)$ also, which means that $\sigma_i(l_i^\sharp) \in \mathsf{NF}_\mu(\mathcal{R})$ (since $root(l_i^\sharp)$ is not a defined symbol of $\mathcal{R}$) and trivially also is $l_i^\sharp$. Hence, $u_i \to v_i \in \mathsf{iDP}_\mathcal{F}(\mathcal{R},\mu)$. Furthermore, by minimality, $t_i^\sharp = \sigma_i(v_i)$ is innermost μ-terminating. Finally, since $t_i = \sigma_i(s_i) \overset{>\Lambda}{\hookrightarrow}{}^{*}_{\mathsf{i}} \sigma_{i+1}(l_{i+1})$ and $\mu^\sharp$ extends μ to $\mathcal{F} \cup \mathcal{D}^\sharp$ by $\mu^\sharp(f^\sharp) = \mu(f)$ for all $f \in \mathcal{D}$, we also have that $\sigma_i(v_i) = \sigma_i(s_i^\sharp) \overset{!}{\hookrightarrow}_{\mathcal{R},\mu^\sharp,\mathsf{i}} \sigma_{i+1}(u_{i+1})$.
2. $v_i = x_i$ if (D2) holds. Again, since $\sigma_i(l_i)$ is argument μ-normalized, it implies that $\sigma_i(l_i^\sharp)$ also, which means that $\sigma_i(l_i^\sharp) \in \mathsf{NF}_\mu(\mathcal{R})$ (since $root(l_i^\sharp)$ is not a defined symbol of $\mathcal{R}$) and trivially also is $l_i^\sharp$. Clearly, $u_i \to v_i \in \mathsf{iDP}_\mathcal{X}(\mathcal{R},\mu)$. As discussed above, $t_i^\sharp$ is innermost μ-terminating by minimality. Since $\sigma_i(x_i) = C_i[t_i]$, we have that $\sigma_i(v_i) = C_i[t_i]$. By the hiding property, we know that $C_i[\,]$ is an instance of hiding context $C'_i[\,]$, then we have that $\theta_i(C'_i)[t_i] \overset{\Lambda}{\hookrightarrow}{}^{*}_{\mathcal{S}_{\triangleright_\mu},\mu} t_i$. And we also know that t_i is an instance $\theta_i(t'_i)$ of a hidden term $t'_i \in \mathcal{NHT}(\mathcal{R},\mu)$. Thus $t'_i \to t'^\sharp_i \in \mathcal{S}_\sharp$ and

we have $\theta(t'_i) \xhookrightarrow{\Lambda}_{\mathcal{S}_\sharp,\mu} \theta(t'^\sharp_i)$. Finally, since $t_i \xhookrightarrow{>\Lambda}^*_{\mathsf{i}} \sigma_{i+1}(l_{i+1})$, again we have that $t^\sharp_i \xhookrightarrow{!}_{\mathcal{R},\mu^\sharp,\mathsf{i}} \sigma_{i+1}(u_{i+1})$.

Regarding σ, w.l.o.g. we can assume that $\mathcal{V}ar(l_i) \cap \mathcal{V}ar(l_j) = \varnothing$ for all $i \neq j$, and therefore $\mathcal{V}ar(u_i) \cap \mathcal{V}ar(u_j) = \varnothing$ as well. Then, σ is given by $\sigma(x) = \sigma_i(x)$ whenever $x \in \mathcal{V}ar(u_i)$ for $i \geq 1$. From the discussion in points (1) and (2) above, we conclude that the ICSDPs $u_i \to v_i$ for $i \geq 1$ together with σ define an infinite minimal innermost $(\mathsf{iDP}(\mathcal{R},\mu),\mathcal{R},\mathsf{unh}(\mathcal{R},\mu),\mu^\sharp,\mathbf{i})$-chain which contradicts our initial assumption. □

Now we prove that the use of ICSDPs is not only correct but also complete for proving innermost termination of *CSR*.

Theorem 3 (Completeness) *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. If $\mathcal{R}$ is innermost μ-terminating, then there is no infinite minimal innermost $(\mathsf{iDP}(\mathcal{R},\mu),\mathcal{R},\mathsf{unh}(\mathcal{R},\mu),\mu^\sharp,\mathbf{i})$-chain.*

Proof.

By contradiction. If there is an infinite minimal innermost $(\mathsf{iDP}(\mathcal{R},\mu),\mathcal{R},\mathsf{unh}(\mathcal{R},\mu),\mu^\sharp,\mathbf{i})$-chain, then there is a substitution σ and dependency pairs $u_i \to v_i \in \mathsf{iDP}(\mathcal{R},\mu)$ such that $\sigma(u_i) \in \mathsf{NF}_\mu(\mathcal{R})$ and

1. $\sigma(v_i) \xhookrightarrow{!}_{\mathcal{R},\mu^\sharp,\mathsf{i}} \sigma(u_{i+1})$, if $u_i \to v_i \in \mathsf{iDP}_{\mathcal{F}}(\mathcal{R},\mu)$, and
2. if $u_i \to v_i = u_i \to x_i \in \mathsf{iDP}_{\mathcal{X}}(\mathcal{R},\mu)$, then $\sigma(v_i) \xhookrightarrow{\Lambda}^*_{\mathcal{S}_{\rhd_\mu},\mu} \circ \xhookrightarrow{\Lambda}_{\mathcal{S}_\sharp,\mu} t_i \xhookrightarrow{!}_{\mathcal{R},\mu,\mathsf{i}} \sigma(u_{i+1})$.

for $i \geq 1$. Now, consider the first dependency pair $u_1 \to v_1$ in the sequence:

1. If $u_1 \to v_1 \in \mathsf{iDP}_{\mathcal{F}}(\mathcal{R},\mu)$, then $v_1^\natural$ is a μ-replacing subterm of the right-hand-side r_1 of a rule $l_1 \to r_1$ in $\mathcal{R}$. Therefore, $r_1 = C_1[v_1^\natural]_{p_1}$ for some $p_1 \in \mathcal{P}os^\mu(r_1)$ and, since $\sigma(u_1) \in \mathsf{NF}_\mu(\mathcal{R})$, we can perform the innermost μ-rewriting step $s_1 = \sigma(u_1^\natural) \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} \sigma(r_1) = \sigma(C_1)[\sigma(v_1^\natural)]_{p_1} = t_1$, where $\sigma(v_1^\natural)^\sharp = \sigma(v_1) \xhookrightarrow{!}_{\mathcal{R},\mu^\sharp,\mathsf{i}} \sigma(u_2)$ and $\sigma(u_2)$ also initiates an infinite minimal innermost $(\mathsf{iDP}(\mathcal{R},\mu),\mathcal{R},\mathsf{unh}(\mathcal{R},\mu),\mu^\sharp,\mathbf{i})$-chain. Note that $p_1 \in \mathcal{P}os^\mu(t_1)$.
2. If $u_1 \to x_1 \in \mathsf{iDP}_{\mathcal{X}}(\mathcal{R},\mu)$, then there is a rule $l_1 \to r_1$ in $\mathcal{R}$ such that $u_1 = l_1^\sharp$, and $x_1 \in \mathcal{V}ar^\mu(r_1) \setminus \mathcal{V}ar^\mu(l_1)$, i.e., $r_1 = C_1[x_1]_{q_1}$ for some $q_1 \in \mathcal{P}os^\mu(r_1)$. Furthermore, if $\sigma(v_1) = \sigma(x_1) = C_1[t_1] \xhookrightarrow{\Lambda}^*_{\mathcal{S}_{\rhd_\mu},\mu} t_1$ this means that $C_1[\Box]$ is an instance of a hiding context $C'_1[\Box]$, $C_1[\Box] = \theta_1(C'_1)[\Box]$. Furthermore, t_1 is μ-replacing in $C_1[t_1]$. If $t_1 \xhookrightarrow{\Lambda}_{\mathcal{S}_\sharp,\mu} t_1^\sharp$ means that $t_1 = \theta_1(t'_1)$ for some $t'_1 \in \mathcal{DHT}$, then since $\sigma(u_1) = \sigma(l_1)^\sharp \in \mathsf{NF}_\mu(\mathcal{R})$, we can perform the innermost μ-rewriting step $s_1 = \sigma(l_1) \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} \sigma(r_1) = \sigma(C_1)[C'_1[t_1]_{p'_1}]_{q_1} = s_1$ where $t_1^\sharp \xhookrightarrow{!}_{\mathcal{R},\mu^\sharp,\mathsf{i}} \sigma(u_2)$ (hence $t_1 \xhookrightarrow{!}_{\mathsf{i}} u_2^\natural$) and $\sigma(u_2)$

initiates an infinite minimal innermost $(\mathsf{iDP}(\mathcal{R},\mu),\mathcal{R},\mathsf{unh}(\mathcal{R},\mu),\mu^\sharp,\mathbf{i})$-chain. Note that $p_1 = q_1.p'_1 \in \mathcal{P}os^\mu(s_1)$ where p'_1 is the position of the hole in $C_1[\Box]_{p'_1}$.

Since $\mu^\sharp(f^\sharp) = \mu(f)$, and $p_1 \in \mathcal{P}os^\mu(s_1)$, we have that $t_1 \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} s_2[\sigma(u_2)]_{p_1} = s_2$ and $p_1 \in \mathcal{P}os^\mu(s_2)$. Therefore, we can build in that way an infinite innermost μ-rewrite sequence

$$s_1 \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} t_1 \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} s_2 \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} \cdots$$

which contradicts the innermost μ-termination of $\mathcal{R}$.

$\Box$

As a corollary of Theorems 2 and 3, we have.

Corollary 3 (Characterization of innermost μ-termination) *Let $\mathcal{R}$ be a TRS and $\mu \in M_\mathcal{R}$. Then, $\mathcal{R}$ is innermost μ-terminating if and only if there is no infinite minimal innermost $(\mathsf{iDP}(\mathcal{R},\mu),\mathcal{R},\mathsf{unh}(\mathcal{R},\mu),\mu^\sharp,\mathbf{i})$-chain.*

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

$$\begin{array}{rcl} \mathtt{b} & \to & \mathtt{c}(\mathtt{b}) \\ \mathtt{f}(\mathtt{c}(x),x) & \to & \mathtt{f}(x,x) \end{array}$$

together with $\mu(\mathtt{f}) = \{1,2\}$ and $\mu(\mathtt{c}) = \varnothing$. There is only one ICSDP:

$$\mathtt{F}(\mathtt{c}(x),x) \quad \to \quad \mathtt{F}(x,x)$$

Since $\mu^\sharp(\mathtt{F}) = \{1,2\}$, if a substitution σ satisfies $\sigma(\mathtt{F}(\mathtt{c}(x),x)) \in \mathsf{NF}_\mu(\mathcal{R})$, then $\sigma(x) = s$ is in μ-normal form. Assume that the dependency pair is part of an innermost μ-chain. Since there is no way to μ-rewrite $F(s,s)$, there must be $F(s,s) = F(c(t),t)$ for some term t, which means that $s = t$ and $c(t) = s$, i.e., $t = c(t)$ which is not possible. Thus, there is no infinite innermost chain of ICSDPs for $\mathcal{R}$, which is proved innermost terminating by Theorem 2.

Of course, ad-hoc reasonings like in Example 7 do not lead to automation. In following sections we discuss how to prove termination of innermost *CSR* by giving *constraints* on terms that can be solved by using standard methods.

5 Mechanizing Proofs of Innermost μ-termination

Regarding termination proofs, the central notion in the Dependency Pair Framework [GTS04, GTSF06, Thi07] is that of *DP-termination problem*: given a TRS $\mathcal{R}$ and a set of pairs $\mathcal{P}$, the goal is checking the absence (or presence) of infinite (minimal) chains. Termination of a TRS $\mathcal{R}$ is addressed as a DP-termination problem where $\mathcal{P} = \mathsf{DP}(\mathcal{R})$. The most important notion regarding mechanization of the proofs is that of *processor*. A (correct) processor basically transforms DP-termination problems into (hopefully) *simpler* ones, in such a way that the existence of an infinite chain in the original DP-termination problem implies the existence of an infinite chain in the transformed one. Here 'simpler' usually means that fewer pairs are involved. Often, processors are not only correct but also *complete*, i.e., there is an infinite minimal chain in the original DP-termination problem if and only if there is an infinite minimal chain in the transformed problem. This is essential if we are interested in *disproving* termination.

In [AEF+08, AGL10, GL10], we have developed a CSDP framework for *CSR*. In this chapter, we extend the CSDP framework developed in [GL10] to innermost *CSR*. First, we recall the definition of chain for standard *CSR*.

Definition 8 ((Minimal) μ-Chain of Pairs [GL10]) *Let $\mathcal{R}$, $\mathcal{P}$ and $\mathcal{S}$ be TRSs and $\mu \in M_{\mathcal{R}\cup\mathcal{P}\cup\mathcal{S}}$. A $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{t})$-chain is a finite or infinite sequence of pairs $u_i \to v_i \in \mathcal{P}$, together with a substitution σ satisfying that, for all $i \geq 1$,*

1. *if $v_i \notin \mathcal{V}ar(u_i) \setminus \mathcal{V}ar^\mu(u_i)$, then $\sigma(v_i) = t_i \hookrightarrow^*_{\mathcal{R},\mu} \sigma(u_{i+1})$, and*
2. *if $v_i \in \mathcal{V}ar(u_i) \setminus \mathcal{V}ar^\mu(u_i)$, then $\sigma(v_i) \stackrel{\Lambda}{\hookrightarrow}^*_{\mathcal{S}_{\rhd_\mu},\mu} \circ \stackrel{\Lambda}{\hookrightarrow}_{\mathcal{S}_\sharp,\mu} t_i \hookrightarrow^*_{\mathcal{R},\mu} \sigma(u_{i+1})$.*

A $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{t})$-chain is called minimal *if for all $i \geq 1$, t_i is $(\mathcal{R}, \mu)$-terminating.*

Definition 9 (CS Problem) *A CS problem τ is a tuple $\tau = (\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, e)$, where $\mathcal{R}$, $\mathcal{P}$ and $\mathcal{S}$ are TRSs, and $\mu \in M_{\mathcal{R}\cup\mathcal{P}\cup\mathcal{S}}$ and $e \in \{\mathbf{t}, \mathbf{i}\}$ is a flag that stands for* **t**ermination *or* **i**nnermost termination *of* CSR. *The* CS problem $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, e)$ *is* finite *if there is no infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, e)$-chain. The* CS problem $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, e)$ *is* infinite *if $\mathcal{R}$ is non-μ-terminating (for $e = \mathbf{t}$) or innermost non-μ-terminating (for $e = \mathbf{i}$) or there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, e)$-chain.*

Definition 10 (CS Processor) *A CS processor* Proc *is a mapping from CS problems into sets of CS problems. Alternatively, it can also return* "no"*. A CS processor* Proc *is*

- sound *if for all CS problems τ, τ is finite whenever* $\mathsf{Proc}(\tau) \neq \mathsf{no}$ *and $\forall \tau' \in \mathsf{Proc}(\tau)$, τ' is finite.*
- complete *if for all CS problems τ, τ is infinite whenever* $\mathsf{Proc}(\tau) = \mathsf{no}$ *or $\exists \tau' \in \mathsf{Proc}(\tau)$ such that τ' is infinite.*

Now we have the following result which extends the framework in [GL10] to innermost *CSR*.

Theorem 4 (CSDP Framework) *Let $\mathcal{R}$ be a TRS and $\mu \in M_\mathcal{R}$. We construct a tree whose nodes are labeled with CS problems or* "yes" *or* "no"*, and whose root is labeled with $(\mathcal{P}, \mathcal{R}, \mathsf{unh}(\mathcal{R}, \mu), \mu^\sharp, e)$, where $\mathcal{P} = \mathsf{DP}(\mathcal{R}, \mu)$ if $e = \mathbf{t}$ and $\mathcal{P} = \mathsf{iDP}(\mathcal{R}, \mu)$ if $e = \mathbf{i}$. For every inner node labeled with τ, there is a sound processor* Proc *satisfying one of the following conditions:*

1. $\mathsf{Proc}(\tau) = \mathsf{no}$ *and the node has just one child, labeled with* “no”*.*

2. $\mathsf{Proc}(\tau) = \varnothing$ *and the node has just one child, labeled with* “yes”*.*

3. $\mathsf{Proc}(\tau) \neq \mathsf{no}$*,* $\mathsf{Proc}(\tau) \neq \varnothing$*, and the children of the node are labeled with the CS problems in* $\mathsf{Proc}(\tau)$*.*

If all leaves of the tree are labeled with “yes”*, then* $\mathcal{R}$ *is innermost* μ*-terminating. Otherwise, if there is a leaf labeled with* “no” *and if all processors used on the path from the root to this leaf are complete, then* $\mathcal{R}$ *is not innermost* μ*-terminating.*

In following sections we describe a number of sound and (most of them) complete CS-processors for proving termination of innermost *CSR*. First, we formalize some basic processors.

6 CS Basic Processors for Innermost Termination of *CSR*

In standard rewriting, Gramlich, showed that termination and innermost termination coincide for locally confluent overlay TRSs $\mathcal{R}$[Gra95, Theorem 3.23]. Thus, his result allows us to prove termination of such TRSs $\mathcal{R}$by proving innermost termination of $\mathcal{R}$. Although local confluence is undecidable, every nonoverlapping rewrite system is also a locally confluent overlay system, therefore, this approximation is commonly adopted. However, for context-sensitive rewriting this is not enough. This fact was noticed by Lucas in a personal communication showing the following example:

Example 8 *Consider the following TRS* $\mathcal{R}$*:*

$$\begin{array}{rcl} \mathtt{f}(x,x) & \to & \mathtt{b} \\ \mathtt{f}(x,g(x)) & \to & \mathtt{f}(x,x) \\ \mathtt{c} & \to & \mathtt{g}(c) \end{array}$$

together with $\mu(\mathtt{f}) = \{1,2\}$ *and* $\mu(\mathtt{g}) = \varnothing$*. This system is nonoverlapping and innermost* μ*-terminating, but not* μ*-terminating since* $\mathtt{f}(\mathtt{c},\mathtt{c}) \hookrightarrow_\mu \underline{\mathtt{f}(\mathtt{c},\mathtt{g}(\mathtt{c}))} \hookrightarrow_\mu \mathtt{f}(\mathtt{c},\mathtt{c}) \hookrightarrow_\mu \cdots$

Later, in [GL02b, GM02b] it is proved that the equivalence between termination of innermost *CSR* and termination of *CSR* holds in some interesting cases. Thanks to this, the following result was formulated:

Theorem 5 *[GM02b] Let* $\mathcal{R} = (\Sigma, R)$ *be an orthogonal TRS and* $\mu \in M_\Sigma$*.* $\mathcal{R}$ *is* μ*-terminating if and only if it is innermost* μ*-terminating.*

A similar result can be found in [GL02b]. First, we need the following definition:

Definition 11 *[Luc98, Definition 5] Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu \in M_{\mathcal{R}}$. $\mathcal{R}$ has* left-homogeneous replacing variables *(LHRV for short) if, for every μ-replacing variable x in the left-hand side l of a rule $l \to r \in R$, all occurrences of x are replacing in both, l and r.*

Theorem 6 *[GL02b, Theorem 7] Let $\mathcal{R} = (\Sigma, R)$ be a TRS and $\mu \in M_{\mathcal{R}}$ be such that $\mathcal{R}$ is a locally confluent overlay system satisfying LHRV. If $\mathcal{R}$ is innermost μ-terminating, then it is also μ-terminating.*

So, whenever it is possible, we switch to innermost μ-termination since proofs are often easier due to the fact that when considering an innermost rewriting step, we know that every possible subterm of our redex is in normal form with respect to our rewriting relation. For instance, this is shown when estimating the graph.

On the other hand, we have developed a huge amount of processors for proving termination of *CSR* [AGL10, GL10] and it is also interesting to use them in proofs of innermost termination of *CSR*.

Theorem 7 (Commuting Processors) *Let $\tau = (\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$ be a CS problem such that*

1. *$(\mathcal{R} \cup \mathcal{P} \cup \mathcal{S})$ is nonoverlapping and satisfies LHRV, or*
2. *$(\mathcal{R} \cup \mathcal{P} \cup \mathcal{S})$ is orthogonal*

Then, the processors $\mathsf{Proc}_{\mathbf{t}\to\mathbf{i}}$ and $\mathsf{Proc}_{\mathbf{i}\to\mathbf{t}}$ given by

$$\mathsf{Proc}_{\mathbf{t}\to\mathbf{i}}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{t}) = \begin{cases} \{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})\} & \textit{if (1) or (2)} \\ \{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{t})\} & \textit{otherwise} \end{cases}$$

$$\mathsf{Proc}_{\mathbf{i}\to\mathbf{t}}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i}) = \begin{cases} \{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{t})\} & \textit{if (1) or (2)} \\ \{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})\} & \textit{otherwise} \end{cases}$$

are sound and $\mathsf{Proc}_{\mathbf{t}\to\mathbf{i}}$ also complete.

Proof. Regarding soundness of $\mathsf{Proc}_{\mathbf{t}\to\mathbf{i}}$, we proceed by contradiction. Assume that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{t})$-chain A, but there is no infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chain. Due to the finiteness of $\mathcal{P}$ and $\mathcal{S}$, we can assume that there are subsets $\mathcal{Q} \subseteq \mathcal{P}$ and $\mathcal{T} \subseteq \mathcal{S}$ such that A has a tail B

$$\sigma(u_1) \left\{ \begin{array}{c} \hookrightarrow_{\mathcal{Q}_{\mathcal{G}},\mu} \\ \hookrightarrow_{\mathcal{Q}_{\mathcal{X}},\mu} \circ \xrightarrow{\Lambda}{}^{*}_{\mathcal{T}_{\rhd_\mu},\mu} \circ \stackrel{\Lambda}{\hookrightarrow}_{\mathcal{T}_\sharp,\mu} \end{array} \right\} t_1 \hookrightarrow^{*}_{\mathcal{R},\mu} \sigma(u_2) \left\{ \begin{array}{c} \hookrightarrow_{\mathcal{Q}_{\mathcal{G}},\mu} \\ \hookrightarrow_{\mathcal{Q}_{\mathcal{X}},\mu} \circ \xrightarrow{\Lambda}{}^{*}_{\mathcal{T}_{\rhd_\mu},\mu} \circ \stackrel{\Lambda}{\hookrightarrow}_{\mathcal{T}_\sharp,\mu} \end{array} \right\} t_2 \hookrightarrow^{*}_{\mathcal{R},\mu} \cdots$$

for some substitution σ, where all pairs in $\mathcal{Q}$ and all rules in $\mathcal{T}$ are infinitely often used (note that, if $\mathcal{T} \neq \varnothing$, then $\mathcal{T}_\sharp \neq \varnothing$ and $\mathcal{Q}_{\mathcal{X}} \neq \varnothing$), and for all $i \geq 1$, (a) if $u_i \to v_i \in \mathcal{Q}_{\mathcal{G}}$, then $t_i = \sigma(v_i)$ and (b) if $u_i \to v_i = u_i \to x_i \in \mathcal{Q}_{\mathcal{X}}$, then $\sigma(u_i) \hookrightarrow_{\mathcal{Q}_{\mathcal{X}}} \circ \xrightarrow{\Lambda}{}^{*}_{\mathcal{T}_{\rhd_\mu},\mu} \circ \stackrel{\Lambda}{\hookrightarrow}_{\mathcal{T}_\sharp,\mu} t_i$. Moreover, all t_i are $(\mathcal{R}, \mu)$-terminating. W.l.o.g. we can assume that $\sigma(u_1)$ is $(\mathcal{R}, \mu)$-terminating.

Proof of item (1) follows [GL02b] result for *CSR* and the details can be found in the original paper for equivalence between innermost termination and termination of TRSs [Gra95]. Proof of item (2) follows the results in [GM02b] and more precisely, the ones in [Emm08] that adapt them to the CSDP framework of [AEF+08]. For details about some statements in the following, we recall the reader to the corresponding lemmas in these papers to simplify the proof. Both shares the main idea of [Gra95] about using a transformation Φ which (*uniquely* since in both cases the system is nonoverlaping) μ-normalizes all maximal subterms of a given term with respect to $\mathcal{R}$ (therefore, top parts of the pairs are untouched). Formally,

$$\Phi_\mu(t) = C[t_1{}^{\mathsf{i}}{\downarrow}_{(\mathcal{R},\mu)}, \ldots, t_n{}^{\mathsf{i}}{\downarrow}_{(\mathcal{R},\mu)}]$$

where $t = C[t_1, \ldots, t_n]$ and $t_1, \ldots, t_n$ are innermost μ-terminating subterms at active positions. Clearly $t \hookrightarrow^*_{\mathcal{R},\mu,\mathsf{i}} \Phi_\mu(t)$. The main difference dealing with context-sensitive rewriting arises in the synchronization within variable parts since e.g. one occurrence of a variable x can be active while another can be inactive. This is solved requiring linearity of left-hand sides (en the case of condition (2)) or *LHRV* (in the case of condition (1)).

Let $u'_1 = \Phi_\mu(\sigma(u_1))$, therefore, u'_1 is an innermost $(\mathcal{R},\mu)$-terminating instance of u_1 and there exists a substitution σ' s.t.

$$\sigma(u_1) \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} u'_1 = \sigma'(u_1) \left\{ \begin{array}{c} \hookrightarrow_{\mathcal{Q}_\mathcal{G},\mu} \\ \hookrightarrow_{\mathcal{Q}_\mathcal{X},\mu} \circ \overset{\Lambda}{\hookrightarrow}^*_{\mathcal{T}_{\rhd_\mu},\mu} \circ \overset{\Lambda}{\hookrightarrow}_{\mathcal{T}_\sharp,\mu} \end{array} \right\} \sigma'(t_1)$$

and $\sigma'(t_1) = \Phi_\mu(\sigma(t_1))$ is innermost $(\mathcal{R},\mu)$-terminating. Paying attention in the part $t_1 \hookrightarrow^*_{\mathcal{R},\mu} \sigma(u_2)$, since by minimality, t_i are $(\mathcal{R},\mu)$-terminating, all contracted redexes in the sequence also will be. Therefore, we can reach the point where $\sigma'(t_1) \hookrightarrow^*_{\mathcal{R},\mu,\mathsf{i}} u'_2$ such that $u'_2 = \Phi_\mu(\sigma(u_2))$ and u'_2 is innermost $(\mathcal{R},\mu)$-terminating.

Since for all pair $u_i \to v_i$ asume variable disjoint, the new substitution can be extended to $\sigma'(u_2) = u'_2$. Reasoning in this way, the original infinite minimal $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{t})$-chain can be seen as:

$$u'_1 \hookrightarrow^*_{\mathcal{R},\mu,\mathsf{i}} \sigma'(u_1) \left\{ \begin{array}{c} \hookrightarrow_{\mathcal{Q}_\mathcal{G},\mu} \\ \hookrightarrow_{\mathcal{Q}_\mathcal{X},\mu} \circ \overset{\Lambda}{\hookrightarrow}^*_{\mathcal{T}_{\rhd_\mu},\mu} \circ \overset{\Lambda}{\hookrightarrow}_{\mathcal{T}_\sharp,\mu} \end{array} \right\} \sigma'(t_1) \hookrightarrow^*_{\mathcal{R},\mu,\mathsf{i}} u'_2 \hookrightarrow^*_{\mathcal{R},\mu,\mathsf{i}} \sigma'(u_2) \cdots$$

where $\sigma'(t_i)$ is innermost $(\mathcal{R},\mu)$-terminating and $\sigma'(u_i) \in \mathsf{NF}_\mu(\mathcal{R})$ for all $i \geq 1$. Therefore we get an infinite minimal $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$-chain, leading to a contradiction.

Regarding completeness of $\mathsf{Proc}_{\mathbf{t}\to\mathbf{i}}$, since if $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$ is infinite, that means that $\mathcal{R}$ is not innermost μ-terminating and therefore $\mathcal{R}$ is not μ-terminating or there is an infinite minimal $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$-chain and, since condition (1) or (2) hold then the equivalence between innermost μ-termination and μ-termination comes from Theorems 5 and 6 respectively and t_i is $(\mathcal{R},\mu)$-terminating and therefore there is also an infinite minimal $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{t})$-chain. Therefore $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{t})$ is infinite.

We prove soundness of $\mathsf{Proc}_{\mathbf{i}\to\mathbf{t}}$ by contradiction. Assume that there is an infinite minimal $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$-chain but there is no infinite minimal $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{t})$. Since condition (1) or (2) hold, reasoning as above every minimal $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$-chain is also a minimal $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{t})$-chain, therefore there is an infinite minimal $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{t})$-chain, leading to a contradiction.

□

Soundness of $\mathsf{Proc}_{\mathbf{i}\to\mathbf{t}}$ needs to impose the requirements about equivalence between innermost μ-termination and μ-termination since we are dealing with minimal chains. Obviously, it is always possible to prove innermost μ-termination of a TRS by proving μ-termination without taking into account any additional condition but this cannot be done when managing minimality.

The following proposition establishes some important 'basic' cases of (absence of) infinite context-sensitive chains of pairs which are used later and with slight differences were presented in [AGL10]. Note that in the innermost case they also hold.

Proposition 5 *Let $\mathcal{R}=(\mathcal{F},R)$ and $\mathcal{P}=(\mathcal{G},P)$ and $\mathcal{S}=(\mathcal{H},S)$ be TRSs and $\mu\in M_{\mathcal{R}\cup\mathcal{P}\cup\mathcal{S}}$.*

1. *If $P=\varnothing$, then there is no $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$-chain.*
2. *If $R=\varnothing$, then there is no infinite $(\mathcal{P}_\mathcal{X},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$-chain.*
3. *Let $u\to v\in\mathcal{P}_\mathcal{G}$ be such that $v'=\theta(u)$ for some substitution θ such that $\theta(u)\in\mathsf{NF}_\mu(\mathcal{R})$ and renamed version v' of v. Then, there is an infinite innermost $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$-chain.*

Proof.

1. Trivial.
2. By contradiction. If there is an infinite $(\mathcal{P}_\mathcal{X},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$-chain, then, since there is no rule in $\mathcal{R}$, there is a substitution σ such that

   $$\sigma(u_1)\hookrightarrow_{\mathcal{P},\mu}\sigma(x_1)\overset{\Lambda}{\hookrightarrow}{}^{*}_{\mathcal{S}_{\rhd_\mu},\mu}s_1\overset{\Lambda}{\hookrightarrow}_{\mathcal{S}_\sharp,\mu}t_1=\sigma(u_2)\hookrightarrow_{\mathcal{P},\mu}\sigma(x_2)\overset{\Lambda}{\hookrightarrow}{}^{*}_{\mathcal{S}_{\rhd_\mu},\mu}s_2\overset{\Lambda}{\hookrightarrow}_{\mathcal{S}_\sharp,\mu}\cdots$$

   For $i\geq 1$, since $x_i\in\mathcal{V}ar(u_i)$ and u_i is not a variable, we have $u_i\rhd x_i$, hence $\sigma(u_i)\rhd\sigma(x_i)$ (by stability of $\rhd$), and also $\sigma(u_i)\rhd s_i$. Since s_i and $\sigma(u_{i+1})$ only differ in the root symbol, we can actually say that $s_i\rhd s_{i+1}$ for all $i\geq 1$. Thus, we obtain an infinite sequence $s_1\rhd s_2\rhd\cdots$ which contradicts the well-foundedness of $\rhd$.

3. Since we always deal with *renamed* versions $u_i\to v_i$ of the pair $u\to v\in\mathcal{P}$, for each $x\in\mathcal{V}ar(u)$, we write x_i to denote the 'name' of the variable x in $u_i\to v_i$. According to our hypothesis, we can assume the existence of substitutions θ_{i+1} such that $v_i=\theta_{i+1}(u_{i+1})$. Note that, for all $x\in\mathcal{V}ar(u)$ and $i\geq 1$, $\mathcal{V}ar(\theta_{i+1}(u_{i+1}))\subseteq\mathcal{V}ar(v_i)\subseteq\mathcal{V}ar(u_i)$ and $\theta(u)\in\mathsf{NF}_\mu(\mathcal{R})$ is needed to deal only with innermost μ-chains.

We can define an infinite innermost $(\{u \to v\}, \varnothing, \varnothing, \mu, \mathbf{i})$-chain (hence an innermost $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chain) by using the renamed versions $u_i \to v_i$ of $u \to v$ for $i \geq 1$ together with σ given (inductively) as follows: for all $x \in \mathcal{V}ar(u)$, $\sigma(x_1) = x_1$ and $\sigma(x_i) = \sigma(\theta_i(x_i))$ for all $i > 1$. Note that $\sigma(v_i) = \sigma(\theta_{i+1}(u_{i+1})) = \sigma(u_{i+1})$ for all $i \geq 1$.

□

According to Proposition 5, for some specific CS problems it is easy to say whether they are finite or not.

Theorem 8 (Basic Innermost CS Processors) *Let $\mathcal{R} = (\mathcal{F}, R)$, $\mathcal{P} = (\mathcal{G}, P)$ and $\mathcal{S} = (\mathcal{H}, S)$ be TRSs and $\mu \in M_{\mathcal{R}\cup\mathcal{P}\cup\mathcal{S}}$.*

Then, the processors Proc_{Fin} and Proc_{Inf} given by

$$\mathsf{Proc}_{Fin}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i}) = \begin{cases} \varnothing & \text{if } P = \varnothing \vee (R = \varnothing \wedge \mathcal{P} = \mathcal{P}_{\mathcal{X}})\text{; and} \\ \{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})\} & \text{otherwise} \end{cases}$$

$$\mathsf{Proc}_{Inf}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i}) = \begin{cases} \mathsf{no} & \text{if } v = \theta(u) \text{ and } \theta(u) \in \mathsf{NF}_{\mu}(\mathcal{R}) \\ & \text{for some } u \to v \in \mathcal{P}_{\mathcal{G}} \text{ and substitution } \theta\text{; and} \\ \{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})\} & \text{otherwise} \end{cases}$$

are sound and complete.

The CS problems in Theorem 8 provide the necessary *base* cases for our proofs of innermost termination of *CSR*.

In the following sections we are going to show some powerful techniques adapted from standard rewriting to deal with proofs of innermost termination of *CSR*.

7 Innermost Context-Sensitive Dependency Graph

The analysis of infinite (minimal) chains of pairs is essential in the (CS)DP framework [GTSF06, AGL10, GL10]. Following [GL10], for innermost *CSR* we have the following.

Definition 12 (Innermost Context-Sensitive Graph of Pairs) *Let $\mathcal{R}$, $\mathcal{P}$ and $\mathcal{S}$ be TRSs and $\mu \in M_{\mathcal{R}\cup\mathcal{P}\cup\mathcal{S}}$. The* innermost context-sensitive (ICS) graph *$\mathsf{IG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu)$ has $\mathcal{P}$ as the set of nodes. Given $u \to v, u' \to v' \in \mathcal{P}$, there is an arc from $u \to v$ to $u' \to v'$ if $u \to v, u' \to v'$ is a minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chain for some substitution σ.*

In termination proofs, we are concerned with the analysis of *strongly connected components* (SCCs). A strongly connected component in a graph is a *maximal cycle*, i.e., a cycle which is not contained in any other cycle. The following result justifies the use of SCCs for proving the absence of infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chains.

Theorem 9 (SCC processor [GL10]) *Let $\mathcal{R}$, $\mathcal{P}$ and $\mathcal{S}$ be TRSs and $\mu \in M_{\mathcal{R}\cup\mathcal{P}\cup\mathcal{S}}$. Then, the processor* Proc_{SCC} *given by*

$$\mathsf{Proc}_{SCC}(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i}) = \{(\mathcal{Q},\mathcal{R},\mathcal{S}_{\mathcal{Q}},\mu,\mathbf{i}) \mid \mathcal{Q} \textit{ contains the pairs of an SCC in } \mathsf{IG}(\mathcal{P},\mathcal{R},\mathcal{S},\mu)\}$$

(where $\mathcal{S}_{\mathcal{Q}}$ are the rules from $\mathcal{S}$ involving a possible $(\mathcal{Q},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$-chain) is sound and complete.

As a consequence of this theorem, we can *separately* work with the strongly connected components of $\mathsf{IG}(\mathcal{P},\mathcal{R},\mathcal{S},\mu)$, disregarding other parts of the graph.

Now we can use these notions to introduce the innermost context-sensitive dependency graph, i.e., the graph whose nodes instead of being arbitrary pairs are the ICSDPs ($\mathcal{P} = \mathsf{iDP}(\mathcal{R},\mu)$).

Definition 13 (Innermost Context-Sensitive Dependency Graph (ICS-DG)) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. The* Innermost Context-Sensitive Dependency Graph *associated to $\mathcal{R}$ and μ is* $\mathsf{IDG}(\mathcal{R},\mu) = \mathsf{IG}(\mathsf{iDP}(\mathcal{R},\mu),\mathcal{R},\mathsf{unh}(\mathcal{R},\mu),\mu^\sharp)$.

7.1 Estimating the ICS Graph

In general, the innermost context-sensitive graph of a CS problem is *not* computable: it involves reachability of $\theta'(u')$ from $\theta(v)$ (for $u \to v \in \mathcal{P}_{\mathcal{G}}$) or $\theta(t)$ (for t such that $s \to t \in \mathcal{S}_\sharp$) using innermost *CSR*; as in the unrestricted case, the reachability problem for innermost *CSR* is undecidable. So, we need to use some approximation of it.

In [AGL10], we have adapted to the context-sensitive setting the more recent approximation for standard rewriting [GTS05]. Given a TRS $\mathcal{R}$ and a replacement map μ, we let $\textsc{TCap}^\mu_\mathcal{R}$ be as follows:

$$\begin{array}{rcl}
\textsc{TCap}^\mu_\mathcal{R}(x) & = & y \;\text{ if } x \text{ is a variable, and} \\
\textsc{TCap}^\mu_\mathcal{R}(f(t_1,\ldots,t_k)) & = & \left\{\begin{array}{ll} f([t_1]^f_1,\ldots,[t_k]^f_k) & \text{if } f([t_1]^f_1,\ldots,[t_k]^f_k) \text{ does not unify} \\ & \text{with } l \text{ for any } l \to r \text{ in } \mathcal{R} \\ y & \text{otherwise} \end{array}\right.
\end{array}$$

where y is intended to be a new, fresh variable that has not yet been used and given a term s, $[s]^f_i = \textsc{TCap}^\mu_\mathcal{R}(s)$ if $i \in \mu(f)$ and $[s]^f_i = s$ if $i \notin \mu(f)$. We assume that l shares no variable with $f([t_1]^f_1,\ldots,[t_k]^f_k)$ when the unification is attempted. Function $\textsc{TCap}^\mu_\mathcal{R}$ is intended to provide a suitable approximation of the aforementioned $(\mathcal{R},\mu)$-reachability problems by means of *unification*.

Proposition 6 ([AGL10]) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. Let $t, u \in \mathcal{T}(\mathcal{F},\mathcal{X})$ be such that $\mathcal{V}ar(t) \cap \mathcal{V}ar(u) = \varnothing$. If $\theta(t) \hookrightarrow^* \theta(u)$ for some substitution θ, then $\textsc{TCap}^\mu_\mathcal{R}(t)$ and u unify.*

In contrast to standard (μ-)rewriting, in the innermost setting it is not necessary to rename multiple occurrences of variables since all variables are always instantiated to (μ-)normal forms and cannot be reduced. However, in innermost *CSR* we have to replace by fresh variables those ones that are μ-replacing in the right hand side of the pair, v, but not in the left-hand side, u, since they are not μ-normalized. Moreover we need to substitute every subterm with a defined root symbol by fresh variables only if the term is not equal to a μ-replacing subterm of u or it unifies with the left-hand side of some rule in $\mathcal{R}$.

We define a new version of the function, $\mathsf{i}\mathrm{TC}\textsc{ap}^{\mu}_{\mathcal{R},u}$, which is able to approximate the ICS graph by taking into account these particularities of innermost *CSR*.

Definition 14 *Given a TRS $\mathcal{R}$, a replacement map μ and a term u, we let $\mathsf{i}\mathrm{TC}\textsc{ap}^{\mu}_{\mathcal{R},u}$ be as follows:*

$$\mathrm{iTC}\textsc{ap}^{\mu}_{\mathcal{R},u}(x) = \begin{cases} y & \text{if } x \in \mathcal{X} \text{ and } x \notin \mathcal{V}ar^{\mu}(u) \\ x & \text{otherwise} \end{cases}$$

$$\mathrm{iTC}\textsc{ap}^{\mu}_{\mathcal{R},u}(f(t_1,\ldots,t_k)) = \begin{cases} f([t_1]^f_1,\ldots,[t_k]^f_k) & \text{if } f([t_1]^f_1,\ldots,[t_k]^f_k) \text{ does not unify with } l \text{ for any} \\ & l \to r \text{ in } \mathcal{R} \text{ or it is equal to a } \mu\text{-replacing subterm of } u \\ y & \text{otherwise} \end{cases}$$

where y is intended to be a new, fresh variable that has not yet been used and given a term s, $[s]^f_i = \mathsf{i}\mathrm{TC}\textsc{ap}^{\mu}_{\mathcal{R},u}(s)$ if $i \in \mu(f)$ and $[s]^f_i = s$ if $i \notin \mu(f)$. We assume that l shares no variable with $f([t_1]^f_1,\ldots,[t_k]^f_k)$ when the unification is attempted.

Since when connecting in a chain collapsing pairs we deal with rules in $\mathcal{S}_\sharp$ instead of pairs in $\mathcal{P}_\mathcal{G}$, we cannot look at the left hand side of the pairs. Therefore, for dealing with pairs in $\mathcal{P}_\mathcal{X}$, we have to approximate their arcs in the same way that for *CSR*since we do not store information about left-hand sides of the pairs from which the hidden terms are obtained. So, we have the following:

Definition 15 (Estimated Innermost Context-Sensitive Graph of Pairs) *Let $\mathcal{R} = (\mathcal{F}, R)$, $\mathcal{S} = (\mathcal{H}, S)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F}\cup\mathcal{G}\cup\mathcal{H}}$. The* estimated *ICS-graph associated to $\mathcal{R}$, $\mathcal{P}$ and $\mathcal{S}$ (denoted $\mathsf{EIG}(\mathcal{P},\mathcal{R},\mathcal{S},\mu)$) has $\mathcal{P}$ as the set of nodes and arcs which connect them as follows:*

1. *There is an arc from $u \to v \in \mathcal{P}_\mathcal{G}$ to $u' \to v' \in \mathcal{P}$ if $\mathsf{i}\mathrm{TC}\textsc{ap}^{\mu}_{\mathcal{R},u}(v)$ and u' unify by some mgu σ such that $\sigma(u), \sigma(u') \in \mathsf{NF}_\mu(\mathcal{R})$.*
2. *There is an arc from $u \to v \in \mathcal{P}_\mathcal{X}$ to $u' \to v' \in \mathcal{P}$ if there is $s \to t \in \mathcal{S}_\sharp$ such that $\mathrm{TC}\textsc{ap}^{\mu}_{\mathcal{R}}(t)$ and u' unify by some mgu σ such that $\sigma(u') \in \mathsf{NF}_\mu(\mathcal{R})$.*

Definition 16 (Correctness of the Estimated ICS-Graph of Pairs) *Let $\mathcal{R} = (\mathcal{F}, R)$, $\mathcal{S} = (\mathcal{H}, S)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F}\cup\mathcal{G}\cup\mathcal{H}}$. The* estimated *ICS-graph associated to $\mathcal{R}$, $\mathcal{P}$ and $\mathcal{S}$ (denoted $\mathsf{EIG}(\mathcal{P},\mathcal{R},\mathcal{S},\mu)$) has $\mathcal{P}$ as the set of nodes and arcs which connect them as follows:*

1. *If there is an arc from $u \to v \in \mathcal{P}_\mathcal{G}$ to $u' \to v' \in \mathcal{P}$ and substitutions θ and θ' such that $\theta(v) \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} \theta'(u')$, $\theta(u), \theta'(u') \in \mathsf{NF}_\mu(\mathcal{R})$ then $\mathsf{i}\mathrm{TC}\textsc{ap}^{\mu}_{\mathcal{R},u}(v)$ and u' unify by some mgu σ such that $\sigma(u), \sigma(u') \in \mathsf{NF}_\mu(\mathcal{R})$.*

2. *If there is an arc from $u \to v \in \mathcal{P}_{\mathcal{X}}$ to $u' \to v' \in \mathcal{P}$ and there is $s \in \mathcal{NHT}(\mathcal{R},\mu)$ such that $\theta(s^\sharp) = \theta(t) = t'$ and substitutions θ and θ' such that $\theta(v) \stackrel{\Lambda}{\hookrightarrow}^{*}_{\mathcal{S}_{\rhd_\mu},\mu} \circ \stackrel{\Lambda}{\hookrightarrow}_{\mathcal{S}_\sharp,\mu} t' \stackrel{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} \theta'(u')$ then there is $s \to t \in \mathcal{S}_\sharp$ such that $\textsc{TCap}^\mu_\mathcal{R}(t)$ and u' unify by some mgu σ such that $\sigma(u') \in \mathsf{NF}_\mu(\mathcal{R})$.*

According to Definition 13, we would have the corresponding one for the *estimated* ICS-DG: $\mathsf{EIDG}(\mathcal{R},\mu) = \mathsf{EIG}(\mathsf{iDP}(\mathcal{R},\mu),\mathcal{R},\mathsf{unh}(\mathcal{R},\mu),\mu^\sharp)$.

Example 9 *Consider the following TRS $\mathcal{R}$ [Zan97, Example 4]:*

$$\begin{array}{rcl}
\mathtt{f}(x) & \to & \mathtt{cons}(x,\mathtt{f}(\mathtt{g}(x))) \\
\mathtt{g}(\mathtt{0}) & \to & \mathtt{s}(\mathtt{0}) \\
\mathtt{g}(\mathtt{s}(x)) & \to & \mathtt{s}(\mathtt{s}(\mathtt{g}(x))) \\
\mathtt{sel}(\mathtt{0},\mathtt{cons}(x,y)) & \to & x \\
\mathtt{sel}(\mathtt{s}(x),\mathtt{cons}(y,z)) & \to & \mathtt{sel}(x,z)
\end{array}$$

with $\mu(\mathtt{0}) = \varnothing$, $\mu(\mathtt{f}) = \mu(\mathtt{g}) = \mu(\mathtt{s}) = \mu(\mathtt{cons}) = \{1\}$, and $\mu(\mathtt{sel}) = \{1,2\}$. Then, $\mathsf{iDP}(\mathcal{R},\mu)$ consists of the following pairs:

$$\mathtt{G}(\mathtt{s}(x)) \to \mathtt{G}(x) \qquad (7)$$

$$\mathtt{SEL}(\mathtt{s}(x),\mathtt{cons}(y,z)) \to \mathtt{SEL}(x,z) \qquad (8)$$

$$\mathtt{SEL}(\mathtt{s}(x),\mathtt{cons}(y,z)) \to z \qquad (9)$$

and the unhiding rules are: $\mathsf{unh}_{\rhd_\mu}(\mathcal{R},\mu) = \{\mathtt{f}(x) \to x\}$ and $\mathsf{unh}_\sharp(\mathcal{R},\mu) = \{\mathtt{f}(\mathtt{g}(x)) \to \mathtt{F}(g(x)), \mathtt{g}(x) \to \mathtt{G}(x)\}$.

Regarding pairs (7) and (8) in $\mathsf{iDP}_\mathcal{F}(\mathcal{R},\mu)$, there is an arc from (7) to itself and another one from (8) to itself. Regarding the only collapsing pair (9), we have $\textsc{TCap}^\mu_\mathcal{R}(\mathtt{F}(\mathtt{g}(x))) = \mathtt{F}(y)$ and $\textsc{TCap}^\mu_\mathcal{R}(\mathtt{G}(x)) = \mathtt{G}(y)$. Since $\mathtt{F}(y)$ does not unify with the left-hand side of any pair, and $\mathtt{G}(y)$ unifies with the left-hand side $\mathtt{G}(\mathtt{s}(x))$ of (7) and $\mathtt{G}(\mathtt{s}(x))$ is in μ-normal form, there is an arc from (9) to (7), see Figure 1. Thus, there are two cycles: $\{(7)\}$ and $\{(8)\}$.

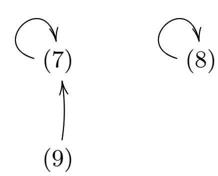

Figure 1: Innermost CS-Dependency graph for Example 9

The following example shows that using $\textsc{iTCap}^\mu_{\mathcal{R},u}$ provides a better approximation of the ICS-DG than using $\textsc{TCap}^\mu_\mathcal{R}$ for noncollapsing pairs.

Example 10 *Consider the following TRS $\mathcal{R}$:*

$$\begin{array}{rcl} \mathtt{f}(\mathtt{a},\mathtt{b},x) & \to & \mathtt{f}(x,x,x) \\ \mathtt{c} & \to & \mathtt{a} \\ \mathtt{c} & \to & \mathtt{b} \end{array}$$

together with $\mu(\mathtt{f}) = \{1,2\}$. There are two ICS-dependency pairs:

$$\begin{array}{rcl} \mathtt{F}(\mathtt{a},\mathtt{b},x) & \to & \mathtt{F}(x,x,x) \\ \mathtt{F}(\mathtt{a},\mathtt{b},x) & \to & x \end{array}$$

$\mathcal{R}$ is not innermost μ-terminating:

$$\mathtt{F}(\underline{\mathtt{c}},\mathtt{c},\mathtt{c}) \hookrightarrow_{\mathcal{R},\mu^\sharp,\mathsf{i}} \mathtt{F}(\mathtt{a},\underline{\mathtt{c}},\mathtt{c}) \hookrightarrow_{\mathcal{R},\mu^\sharp,\mathsf{i}} \underline{\mathtt{F}(\mathtt{a},\mathtt{b},\mathtt{c})} \hookrightarrow_{\mathsf{iDP}(\mathcal{R},\mu,\mathsf{i}),\mu^\sharp} \mathtt{F}(\underline{\mathtt{c}},\mathtt{c},\mathtt{c}) \hookrightarrow_{\mathcal{R},\mu^\sharp,\mathsf{i}} \cdots$$

In order to build the ICS-DG, since there are not hidden terms and therefore $\mathcal{S}_\sharp$ is empty, we only have to check if $\mathsf{i}\mathrm{TCAP}^{\mu}_{\mathcal{R},u}(\mathtt{F}(x,x,x)) = \mathsf{i}\mathrm{TCAP}^{\mu}_{\mathcal{R},\mathtt{F(a,b,x)}}(\mathtt{F}(x,x,x)) = \mathtt{F}(x''',x'',x)$ unifies with $\mathtt{F}(\mathtt{a},\mathtt{b},y)$ so, we get a cycle and the same would be obtained with $\mathrm{TCAP}^{\mu}_{\mathcal{R}}(\mathtt{F}(x,x,x))$. However, if we use $\mu(\mathtt{f}) = \{1,3\}$, the system now is innermost μ-terminating (the collapsing pair now disappears) but if we use the $\mathrm{TCAP}^{\mu}_{\mathcal{R}}$ we get $\mathrm{TCAP}^{\mu}_{\mathcal{R}}(\mathtt{F}(x,x,x)) = \mathtt{F}(x''',x'',x)$ again unifies with $\mathtt{F}(\mathtt{a},\mathtt{b},x)$ and we obtain a spurious cycle. By using $\mathsf{i}\mathrm{TCAP}^{\mu}_{\mathcal{R},u}$, we obtain $\mathsf{i}\mathrm{TCAP}^{\mu}_{\mathcal{R},\mathtt{F(a,b,x)}}(\mathtt{F}(x,x,x)) = \mathtt{F}(x,x,x)$ (since there are not migrating variables now) which does not unify with $\mathtt{F}(\mathtt{a},\mathtt{b},y)$. Now, innermost μ-termination can be easily proved since there are no cycles in the ICS-DG.

After showing that $\mathsf{i}\mathrm{TCAP}^{\mu}_{\mathcal{R},u}$ provides a better approximation of the ICS-DG for noncollapsing pairs, we are going to show that for the collapsing pairs this is not true since we can lead into and underestimation of the graph and conclude a false result.

Example 11 *Consider the following TRS $\mathcal{R}$ which is a variant of Example 10:*

$$\begin{array}{rcl} \mathtt{f}(\mathtt{a},\mathtt{b},x) & \to & \mathtt{g}(\mathtt{f}(x,x,x)) \\ \mathtt{g}(x) & \to & x \\ \mathtt{c} & \to & \mathtt{a} \\ \mathtt{c} & \to & \mathtt{b} \end{array}$$

together with $\mu(\mathtt{f}) = \{1,2\}$ and $\mu(\mathtt{g}) = \varnothing$. There are two ICS-dependency pairs:

$$\mathtt{F}(\mathtt{a},\mathtt{b},x) \to \mathtt{G}(\mathtt{f}(x,x,x)) \quad (10)$$

$$\mathtt{G}(x) \to x \quad (11)$$

$\mathcal{R}$ is not innermost μ-terminating:

$$\mathtt{F}(\underline{\mathtt{c}},\mathtt{c},\mathtt{c}) \hookrightarrow_{\mathcal{R},\mu^\sharp,\mathsf{i}} \mathtt{F}(\mathtt{a},\underline{\mathtt{c}},\mathtt{c}) \hookrightarrow_{\mathcal{R},\mu^\sharp,\mathsf{i}} \underline{\mathtt{F}(\mathtt{a},\mathtt{b},\mathtt{c})} \hookrightarrow_{\mathsf{iDP}(\mathcal{R},\mu),\mu^\sharp,\mathsf{i}} \underline{\mathtt{G}(\mathtt{f}(\mathtt{a},\mathtt{b},\mathtt{c}))} \hookrightarrow_{\mathsf{iDP}(\mathcal{R},\mu),\mu^\sharp,\mathsf{i}} \mathtt{F}(\underline{\mathtt{c}},\mathtt{c},\mathtt{c}) \cdots$$

We have $\mathcal{S}_\sharp = \{\mathtt{f}(x,x,x) \to \mathtt{F}(x,x,x)\}$. Regarding the pair (10) $\in \mathsf{iDP}_{\mathcal{F}}(\mathcal{R},\mu)$, there is an obvious arc from (10) to (11). Regarding the only collapsing pair (11), since we do not have any information in $\mathcal{S}_\sharp$ about migrating variables, we have *to use* $\mathrm{TC}\textsc{ap}^\mu_\mathcal{R}$. *In this way, we have that* $\mathrm{TC}\textsc{ap}^\mu_\mathcal{R}(\mathtt{F}(x,x,x)) = \mathtt{F}(x'',x',x)$ *unifies with* $\mathtt{F}(\mathtt{a},\mathtt{b},y)$ *and we obtain an arc from* (11) *to* (10)*, thus obtaining the existing cycle* $\{(11)\text{-}(10)\}$*. Otherwise, we will not rename any variable and we would not have obtained the arc.*

Example 12 *(Continuing Example 7) Since* $\mathrm{iTC}\textsc{ap}^\mu_{\mathcal{R},\mathtt{F}(c(x),x)}(\mathtt{F}(x,x)) = \mathtt{F}(x,x)$ *and* $\mathtt{F}(c(y),y)$ *do not unify we conclude (and this can easily be implemented) that the ICS-dependency graph for the CS-TRS* $(\mathcal{R},\mu)$ *in Example 7 contains no cycles.*

Since for approximating the innermost context-sensitive graph of a set of pairs, we use function $\mathrm{TC}\textsc{ap}^\mu_\mathcal{R}$ for connecting pairs in $\mathcal{P}_\mathcal{X}$ as done in the context-senstive case, we can also use the following processor instead, which allows a better approximation of the SCCs. This is because if the SCC has no collapsing pairs, the set $\mathcal{S}$ has no sense and if it has, some pairs from $\mathcal{S}_\sharp$ can be removed: those that are not involved in the unification process. Therefore, we will always compute the SCCs by applying the following processor:

Theorem 10 (SCC Processor using $\mathrm{TC}\textsc{ap}^\mu_\mathcal{R}$ **[GL10])** *Let* $\tau = (\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$ *be a CS problem. The CS processor* Proc_{SCC} *given by*

$$\mathsf{Proc}_{SCC}(\tau) = \{(\mathcal{Q},\mathcal{R},\mathcal{S}_\mathcal{Q},\mu) \mid \mathcal{Q} \textit{ contains the pairs of an SCC in } \mathsf{EIG}(\mathcal{P},\mathcal{R},\mathcal{S},\mu)\}$$

where

- $\mathcal{S}_\mathcal{Q} = \varnothing$ *if* $\mathcal{Q}_\mathcal{X} = \varnothing$.
- $\mathcal{S}_\mathcal{Q} = \mathcal{S}_{\rhd_\mu} \cup \{s \to t \mid s \to t \in \mathcal{S}_\sharp, \mathrm{TC}\textsc{ap}^\mu_\mathcal{R}(t)$ *and* u' *unify for some* $u' \to v' \in \mathcal{Q}$ *by some mgu* σ *such that* $\sigma(u') \in \mathsf{NF}_\mu(\mathcal{R})\}$ *if* $\mathcal{Q}_\mathcal{X} \neq \varnothing$.

is sound and complete.

Example 13 *Consider again Example 1. The set* $\mathsf{iDP}(\mathcal{R},\mu)$ *is in Example 5 and the unhiding TRS* $\mathsf{unh}(\mathcal{R},\mu)$ *consists of the rules in Example 6. We can define the following CS problem:*

$$\tau_0 = (\mathsf{iDP}(\mathcal{R},\mu),\mathcal{R},\mathsf{unh}(\mathcal{R},\mu),\mu^\sharp,\mathbf{i})$$

The $\mathsf{EIDG}(\mathcal{R},\mu) = \mathsf{EIG}(\mathsf{iDP}(\mathcal{R},\mu),\mathcal{R},\mathsf{unh}(\mathcal{R},\mu),\mu^\sharp)$ *of the CS problem* τ_0 *is shown in Figure 2. If now we apply the improved SCC processor we get the followings CS subproblems:*

$$\mathsf{Proc}_{SCC}(\tau_0) = \{(\{1\},\mathcal{R},\varnothing,\mu^\sharp,\mathbf{i}),(\{3\},\mathcal{R},\varnothing,\mu^\sharp,\mathbf{i}),(\{4\},\mathcal{R},\varnothing,\mu^\sharp,\mathbf{i})\}$$

8 Usable Rules

An interesting feature in the treatment of innermost termination problems using the dependency pair approach is that, since the variables in the right-hand side of the dependency pairs are in normal form, the rules which can be used to

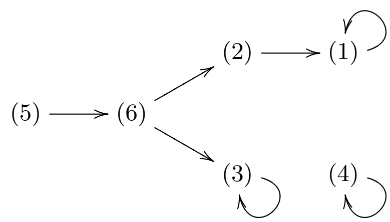

Figure 2: Estimated Innermost CS-Dependency Graph for Example 1

connect contiguous dependency pairs are usually a proper subset of the rules in the TRS. This leads to the notion of *usable rules* [AG00, Definition 32] which simplifies the proofs of innermost termination of rewriting. We adapt this notion to the context-sensitive setting.

Definition 17 (Basic usable CS-rules) *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. For any symbol f let $Rules(\mathcal{R}, f)$ be the set of rules of $\mathcal{R}$ defining f and such that the left-hand side l has no proper μ-replacing $\mathcal{R}$-redex. For any term t, the set of* basic *usable rules $\mathbf{U}_0(\mathcal{R}, \mu, t)$ is as follows:*

$$\begin{array}{lcl} \mathbf{U}_0(\mathcal{R},\mu,x) & = & \varnothing \\ \mathbf{U}_0(\mathcal{R},\mu,f(t_1,\ldots,t_n)) & = & Rules(\mathcal{R},f) \cup \bigcup_{i\in\mu(f)} \mathbf{U}_0(\mathcal{R}',\mu,t_i) \cup \bigcup_{l\to r\in Rules(\mathcal{R},f)} \mathbf{U}_0(\mathcal{R}',\mu,r) \end{array}$$

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

Consider now a TRS $\mathcal{P}$. Then, $\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}) = \bigcup_{l\to r\in\mathcal{P}} \mathbf{U}_0(\mathcal{R}, \mu, r)$. Obviously, $\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}) \subseteq \mathcal{R}$ for all TRSs $\mathcal{P}$ and $\mathcal{R}$.

Interestingly, although our definition is a straightforward extension of the classical one (which just takes into account that μ-rewritings are possible only on μ-replacing subterms), some subtleties arise due to the presence of *non-conservative* rules.

Basic usable rules $\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ in Definition 17 can be used instead of $\mathcal{R}$ when dealing with innermost $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathsf{i})$-chains associated to μ-conservative TRSs $\mathcal{P}$ provided that $\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ is also μ-conservative. This is proved in Theorem 11 below. First, we need some auxiliary results.

Proposition 7 *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let $t, s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and σ be a substitution such that $s = \sigma(t)$ and $\forall x \in \mathcal{V}ar^{\mu}(t)$, $\sigma(x) \in \mathsf{NF}_{\mu}(\mathcal{R})$. If $s \hookrightarrow_{\mathsf{i}} s'$ by applying a rule $l \to r \in \mathcal{R}$, then there is a substitution σ' such that $s' = \sigma'(t')$ for $t' = t[r]_p$ and $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(t)$.*

Proof. Let $p \in \mathcal{P}os^{\mu}(s)$ be the position of an innermost redex $s|_p = \theta(l)$ for some substitution θ. Since $s = \sigma(t)$ and for all replacing variables in t, we have $\sigma(x) \in \mathsf{NF}_{\mu}(\mathcal{R})$, it follows that p is a non-variable (replacing) position of t. Therefore, $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(t)$. Since $s = \sigma(t)$, we have that $s' = \sigma(t)[\theta(r)]_p$ and since

$p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(t)$, by defining $\sigma'(x) = \sigma(x)$ for all $x \in Var(t)$ and $\sigma(x) = \theta(x)$ for all $x \in \mathcal{V}ar(r)$ (as usual, we assume $\mathcal{V}ar(t) \cap \mathcal{V}ar(r) = \varnothing$), we have $s' = \sigma'(t[r]_p)$. $\Box$

The following proposition states that an innermost μ-rewrite step by applying a conservative rule makes the set of μ-replacing variables of the contractum will be instantiated to μ-normal forms.

Proposition 8 *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let $t, s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and σ be a substitution such that $s = \sigma(t)$ and $\forall x \in \mathcal{V}ar^{\mu}(t)$, $\sigma(x) \in \mathsf{NF}_{\mu}(\mathcal{R})$. If $s \hookrightarrow_{\mathsf{i}} s'$ by applying a conservative rule $l \to r \in \mathcal{R}$, then there is a substitution σ' such that $s' = \sigma'(t')$ for $t' = t[r]_p$, $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(t)$ and $\forall x \in \mathcal{V}ar^{\mu}(t')$, $\sigma'(x) \in \mathsf{NF}_{\mu}(\mathcal{R})$.*

Proof. By Proposition 7, we know that σ', as in Proposition 7, satisfies $s' = \sigma'(t')$ for θ as in Proposition 7 and some $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(t)$. Since $s|_p$ is an innermost μ-replacing redex, we have that $\forall y \in \mathcal{V}ar^{\mu}(l)$, $\theta(y) \in \mathsf{NF}_{\mu}(\mathcal{R})$. Since the rule $l \to r$ is conservative, $\mathcal{V}ar^{\mu}(r) \subseteq \mathcal{V}ar^{\mu}(l)$, hence $\forall z \in \mathcal{V}ar^{\mu}(r)$, $\sigma'(z) \in \mathsf{NF}_{\mu}(\mathcal{R})$. Since $\mathcal{V}ar^{\mu}(t[r]_p) \subseteq \mathcal{V}ar^{\mu}(t) \cup \mathcal{V}ar^{\mu}(r)$, we have that $\forall x \in \mathcal{V}ar^{\mu}(t')$, $\sigma'(x) \in \mathsf{NF}_{\mu}(\mathcal{R})$. $\Box$

Now, we prove that in an innermost μ-rewrite sequence starting from a term instantiated with a μ-normalized substitution, the only rules that can be applied are the usable rules (if they are μ-conservative).

Proposition 9 *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let $t, s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and σ be a substitution such that $s = \sigma(t)$ and $\forall x \in \mathcal{V}ar^{\mu}(t)$, $\sigma(x) \in \mathsf{NF}_{\mu}(\mathcal{R})$. If $\mathbf{U}_0(\mathcal{R}, \mu, t)$ is conservative and $s = s_1 \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} s_2 \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} \cdots \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} s_n \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} s_{n+1} = u$ for some $n \geq 0$ then $s_i \hookrightarrow_{\mathbf{U}_0(\mathcal{R},\mu,t),\mu,\mathsf{i}} s_{i+1}$ for all i, $1 \leq i \leq n$.*

Proof. By induction on n. If $n = 0$, then $s = \sigma(t) = u$, it is trivial. Otherwise, if $s_1 \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} s_2 \hookrightarrow^{*}_{\mathcal{R},\mu,\mathsf{i}} u$, we first prove that the result also holds in $s_1 \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} s_2$. By Proposition 7, $s_1 = \sigma(t)$, and $s_2 = \sigma'(t')$ for $t' = t[r]_p$ is such that $s_1|_p = \theta(l)$ and $s_2|_p = \theta(r)$ for some $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(t)$. Thus, $root(l) = root(t|_p)$ and by Definition 17, we can conclude that $l \to r \in \mathbf{U}_0(\mathcal{R}, \mu, t)$. By hypothesis, $\mathbf{U}_0(\mathcal{R}, \mu, t)$ is conservative. Thus, $l \to r$ is conservative and by Proposition 8, $s_2 = \sigma'(t')$ and $\forall x \in \mathcal{V}ar^{\mu}(t')$, $\sigma'(x) \in \mathsf{NF}_{\mu}(\mathcal{R})$. Since $t' = t[r]_p$ and $root(t|_p) = root(l)$, we have that $\mathbf{U}_0(\mathcal{R}, \mu, t') \subseteq \mathbf{U}_0(\mathcal{R}, \mu, t)$ and (since $\mathbf{U}_0(\mathcal{R}, \mu, t)$ is conservative) $\mathbf{U}_0(\mathcal{R}, \mu, t')$ is conservative as well. By the induction hypothesis we know that $s_i \hookrightarrow_{\mathbf{U}_0(\mathcal{R},\mu,t'),\mu,\mathsf{i}} s_{i+1}$ for all i, $2 \leq i \leq n$. Thus we have $s_i \hookrightarrow_{\mathbf{U}_0(\mathcal{R},\mu,t),\mu,\mathsf{i}} s_{i+1}$ for all i, $1 \leq i \leq n$ as desired. $\Box$

The following theorem formalizes a processor to remove pairs from $\mathcal{P}$ by using the previous result and μ-reduction pairs.

Theorem 11 *Let $\tau = (\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$ be a CS problem. Let $(\gtrsim, \sqsupset)$ be a μ-reduction pair such that*

1. $\mathcal{P}$ *and* $\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P})$ *are conservative,*

2. $\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}) \subseteq \gtrsim$ *and* $\mathcal{P} \subseteq \gtrsim \cup \sqsupset$,

Let $\mathcal{P}_{\sqsupset} = \{u \to v \in \mathcal{P} \mid u \sqsupset v\}$. *Then, the processor* Proc_{UR} *given by*

$$\mathsf{Proc}_{UR}(\tau) = \begin{cases} \{(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}), \varnothing, \mu, \mathbf{i})\} & \textit{if (1) and (2) hold} \\ \{(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i})\} & \textit{otherwise} \end{cases}$$

is sound.

Proof.

We proceed by contradiction. Assume that there is an infinite minimal innermost $(\mathcal{P},\mathcal{R},\mathcal{S},\mu,\mathbf{i})$-chain A, but that there is no infinite minimal innermost $(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}), \varnothing, \mu, \mathbf{i})$-chain. Due to the finiteness of $\mathcal{P}$, and since $\mathcal{P}$ is conservative, we have $\mathcal{P}_{\mathcal{X}} = \varnothing$. Thus, we can assume that there is $\mathcal{Q} \subseteq \mathcal{P}$ such that A has a tail B

$$\sigma(u_1) \hookrightarrow_{\mathcal{Q}_{\mathcal{G}},\mu} t_1 \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} \sigma(u_2) \hookrightarrow_{\mathcal{Q}_{\mathcal{G}},\mu} t_2 \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} \sigma(u_3) \hookrightarrow_{\mathcal{Q}_{\mathcal{G}},\mu} \cdots$$

for some substitution σ, where all pairs in $\mathcal{Q}$ are infinitely often used, and, for all $i \geq 1$, since all $u_i \to v_i \in \mathcal{P}$ are conservative $u_i \to v_i \in \mathcal{Q}_{\mathcal{G}}$ (i.e. $\mathcal{P}_{\mathcal{X}} = \mathcal{Q}_{\mathcal{X}} = \varnothing$), then $t_i = \sigma(v_i)$ and $\sigma(u_i) \in \mathsf{NF}_{\mu}(\mathcal{R})$, this implies that $\forall x \in \mathcal{V}ar^{\mu}(u_i)$, $\sigma(x) \in \mathsf{NF}_{\mu}(\mathcal{R})$ and by Proposition 9 the sequence can be seen as:

$$\sigma(u_1) \hookrightarrow_{\mathcal{Q}_{\mathcal{G}},\mu} t_1 \overset{!}{\hookrightarrow}_{\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu,\mathsf{i}} \sigma(u_2) \hookrightarrow_{\mathcal{Q}_{\mathcal{G}},\mu} t_2 \overset{!}{\hookrightarrow}_{\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu,\mathsf{i}} \sigma(u_3) \hookrightarrow_{\mathcal{Q}_{\mathcal{G}},\mu} \cdots$$

Furthermore, by minimality, $t_i = \sigma(v_i)$ is innermost $(\mathcal{R},\mu)$-terminating for all $i \geq 1$. Since $u_i \ (\gtrsim \cup \sqsupset)\ v_i$ for all $u_i \to v_i \in \mathcal{Q} \subseteq \mathcal{P}$, by stability of $\gtrsim$ and $\sqsupset$, we have $\sigma(u_i)\ (\gtrsim \cup \sqsupset)\ \sigma(v_i)$ for all $i \geq 1$. No pair $u \to v \in \mathcal{Q}$ satisfies that $u \sqsupset v$. Otherwise, we get a contradiction by considering that since all pairs in $\mathcal{P}$ are conservative, we have that $u_i \to v_i \in \mathcal{Q}_{\mathcal{G}}$. Then, $t_i = \sigma(v_i) \overset{!}{\hookrightarrow}_{\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu,\mathsf{i}} \sigma(u_{i+1})$ and $t_i \gtrsim \sigma(u_{i+1})$. Since we have $\sigma(u_i)\ (\gtrsim \cup \sqsupset)\ \sigma(v_i)$, by using transitivity of $\gtrsim$ and compatibility between $\gtrsim$ and $\sqsupset$, we conclude that $\sigma(u_i)\ (\gtrsim \cup \sqsupset)\ \sigma(u_{i+1})$. Since $u \to v$ occurs infinitely often in B, there is an infinite set $\mathcal{I} \subseteq \mathbb{N}$ such that $\sigma(u_i) \sqsupset \sigma(u_{i+1})$ for all $i \in \mathcal{I}$. And we have $\sigma(u_i)\ (\gtrsim \cup \sqsupset)\ \sigma(u_{i+1})$ for all other $u_i \to v_i \in \mathcal{Q}$. Thus, by using the compatibility conditions of the μ-reduction pair, we obtain an infinite decreasing $\sqsupset$-sequence which contradicts well-foundedness of $\sqsupset$. Therefore, $\mathcal{Q} \subseteq (\mathcal{P} \setminus \mathcal{P}_{\sqsupset})$. Since $\mathsf{NF}_{\mu}(\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P})) \supseteq \mathsf{NF}_{\mu}(\mathcal{R})$, we have that $\sigma(u_i) \in \mathsf{NF}_{\mu}(\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}))$. By Proposition 9, innermost $(\mathcal{R},\mu)$-termination of $\sigma(v_i)$ implies innermost $(\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu)$-termination of $\sigma(v_i)$ for all $i \geq 1$. Hence, B is an infinite minimal innermost $(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}), \varnothing, \mu, \mathbf{i})$-chain, thus leading to a contradiction. □

Note that the processor is only sound because we refine the result to be applied only to the set of usable rules instead of over the whole set of rules as in standard rewriting [GTSF06] or even for context-sensitive in [AEF+08]. In this way, (i.e. by taking all the rules in $\mathcal{R}$), it would be also complete, that is:

$$\mathsf{Proc}_{UR}(\tau) = \begin{cases} \{(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathcal{R}, \varnothing, \mu, \mathbf{i})\} & \text{if (1) and (2) hold} \\ \{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})\} & \text{otherwise} \end{cases}$$

is sound and complete, but since complete processors are useful for disproving termination, we pay more attention on be more precise with the soundness.

Note also, that in the case of usable rules for context-sensitive rewriting (non innermost) [GLU08], this improvement is not possible to be taken into consideration, since it would be unsound.

Unfortunately, dealing with nonconservative pairs, considering the basic usable CS-rules does *not* ensure a correct approach.

Example 14 *Consider again the TRS $\mathcal{R}$:*

$$\begin{array}{rcl} \mathtt{b} & \to & \mathtt{c}(\mathtt{b}) \\ \mathtt{f}(\mathtt{c}(x), x) & \to & \mathtt{f}(x, x) \end{array}$$

together with $\mu(\mathtt{f}) = \{1\}$ and $\mu(\mathtt{c}) = \varnothing$. There are two *non-conservative ICS-DPs (note that $\mu^\sharp(\mathtt{F}) = \mu(\mathtt{f}) = \{1\}$):*

$$\begin{array}{rcl} \mathtt{F}(\mathtt{c}(x), x) & \to & \mathtt{F}(x, x) \\ \mathtt{F}(\mathtt{c}(x), x) & \to & x \end{array}$$

and only one cycle in the ICS-DG:

$$\{\mathtt{F}(\mathtt{c}(x), x) \quad \to \quad \mathtt{F}(x, x)\}$$

Note that $\mathbf{U}_0(\mathcal{R}, \mu, \mathtt{F}(x, x)) = \varnothing$. Since this ICSDP is strictly compatible with, e.g., an LPO, we would conclude the innermost μ-termination of $\mathcal{R}$. However, this system is not *innermost μ-terminating:*

$$\mathtt{f}(\underline{\mathtt{b}}, \mathtt{b}) \hookrightarrow_{\mathsf{i}} \underline{\mathtt{f}(\mathtt{c}(\mathtt{b}), \mathtt{b})} \hookrightarrow_{\mathsf{i}} \mathtt{f}(\underline{\mathtt{b}}, \mathtt{b}) \hookrightarrow_{\mathsf{i}} \cdots$$

The problem is that we have to take into account the special status of variables in the right-hand side of a nonconservative ICSDP. Instances of such variables are *not* guaranteed to be μ-normal forms. Furthermore, conservativeness of $\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ cannot be dropped either since we could infer an incorrect result as shown by the following example.

Example 15 *Consider the TRS $\mathcal{R}$:*

$$\begin{array}{rcl} \mathtt{b} & \to & \mathtt{c}(\mathtt{b}) \\ \mathtt{f}(\mathtt{c}(x), x) & \to & \mathtt{f}(\mathtt{g}(x), x) \\ \mathtt{g}(x) & \to & x \end{array}$$

together with $\mu(\mathtt{f}) = \{1\}$ and $\mu(\mathtt{g}) = \mu(\mathtt{c}) = \varnothing$. There is only one conservative cycle:

$$\{\mathtt{F}(\mathtt{c}(x), x) \to \mathtt{F}(\mathtt{g}(x), x)\}$$

having only one usable (but non-conservative!) rule $\mathtt{g}(x) \to x$ *This is compatible with the* μ*-reduction pair induced by the following polynomial interpretation:*

$$[\mathtt{f}](x,y) = 0 \qquad [\mathtt{c}](x) = x+1 \qquad [\mathtt{g}](x) = x \qquad [\mathtt{F}](x,y) = x$$

However the system is not innermost μ*-terminating:*

$$\underline{\mathtt{f(c(b),b)}} \hookrightarrow_{\mathsf{i}} \mathtt{f(}\underline{\mathtt{g(b)}}\mathtt{,b)} \hookrightarrow_{\mathsf{i}} \mathtt{f(}\underline{\mathtt{b}}\mathtt{,b)} \hookrightarrow_{\mathsf{i}} \underline{\mathtt{f(c(b),b)}} \hookrightarrow_{\mathsf{i}} \cdots$$

Nevertheless, Theorem 11 is useful to improve the proofs of termination of innermost *CSR* as the following example shows.

Example 16 *Consider again the TRS* $\mathcal{R}$ *in Example 1. As we have seen in Example 13 the initial CS problem can be decomposed in the following three:*

$$\tau_1 = (\{1\}, \mathcal{R}, \varnothing, \mu^\sharp, \mathbf{i}) \tag{12}$$
$$\tau_2 = (\{3\}, \mathcal{R}, \varnothing, \mu^\sharp, \mathbf{i}) \tag{13}$$
$$\tau_3 = (\{4\}, \mathcal{R}, \varnothing, \mu^\sharp, \mathbf{i}) \tag{14}$$

Problems τ_1 *and* τ_3 *can be solved by using the subterm processor (see [GL10]). However, without the notion of usable rules,* τ_2 *is difficult to solve. The pair (3) is* μ*-conservative and the obtained usable rules are also* μ*-conservative:*

$$\mathtt{minus}(x, 0) \to x$$

and

$$\mathtt{minus}(\mathtt{s}(x), \mathtt{s}(x)) \to \mathtt{minus}(x, y)$$

According to Theorem 11, we can apply the usable rules processor $\mathsf{Proc}_{UR}(\tau_2)$ *and get the following problem:*

$$\tau_4 = (\varnothing, \{\mathtt{minus}(x,0) \to x, \mathtt{minus}(\mathtt{s}(x), \mathtt{s}(x)) \to \mathtt{minus}(x,y)\}, \varnothing, \mu^\sharp, \mathbf{i})$$

by using a polynomial interpretation:

$$\begin{array}{rcl@{\qquad}rcl} [\mathtt{minus}](x,y) & = & x & [\mathtt{0}] & = & 0 \\ [\mathtt{s}](x) & = & x+1 & [\mathtt{QUOT}](x,y) & = & x \end{array}$$

Then, by applying $\mathsf{Proc}_{Fin}(\tau_4)$*, since the set of pairs is empty, we can conclude the innermost* μ*-termination of Example 1. Furthermore, since Example 1 is orthogonal, we have also concluded its* μ*-termination.*

9 Usable Arguments for *CSR*

Since in innermost reductions, matching substitutions are always normalized, in an innermost sequence $t_1 \xrightarrow{p_1}_i t_2 \xrightarrow{p_2}_i \cdots \xrightarrow{p_n}_i t_{n+1}$ 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 $l_k \to r_k$ 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 $l \to r$.

In [Fer05] 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 17 *Consider the following TRS $\mathcal{R}$:*

$$\begin{array}{rcl} \mathtt{f}(\mathtt{s}(x), \mathtt{s}(x)) & \to & \mathtt{f}(x, \mathtt{g}(x)) \\ \mathtt{g}(\mathtt{s}(x)) & \to & \mathtt{g}(x) \end{array}$$

No innermost sequence starting at root position takes into account the first argument of $\mathtt{f}$ *nor the argument of* $\mathtt{g}$. *The reason is that since an innermost redex is an argument normalized redex, that means that all variables (e.g. x) of the applied rule are normalized and cannot be reduced. Only the second argument* $\mathtt{g}(x)$ *of* $\mathtt{f}$ *in the right-hand side of the first rule could be innermost reduced after applying it.*

Considering those usable arguments could be helpful in proofs of innermost termination since they impose weaker monotonicity requirements. For instance, when using polynomial orderings, we can use even negative or rational coefficients for interpreting the symbols that do not need to be monotonic.

As Fernández noticed in [Fer05], the set of usable arguments can be seen as a replacement map which specifies the arguments to be reduced. In her approach, proving 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.

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 have adapted Fernández's ideas to *CSR* in [AL09]. 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 10 *A CS-TRS $(\mathcal{R}, \mu)$ is innermost μ-terminating iff $\mathcal{R}'$ is innermost μ-terminating, where $\mathcal{R}' \subseteq \mathcal{R}$ contains all rules $l \to r \in \mathcal{R}$ such that l is argument μ-normalized.*

Proof. Trivial since the only rules that can be applied in innermost μ-reductions are those whose the left-hand sides are argument μ-normalized as we have shown in the Definition 5 of ICSDPs. $\square$

In the following, we assume that all CS-TRS $(\mathcal{R}, \mu)$ are argument μ-normalized, i.e., for all rule $l \to r$ in $\mathcal{R}$, l is argument μ-normalized. Proposition 10 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: the usable CS-arguments for a function symbol $f \in \mathcal{F}$ are those arguments with a μ-replacing subterm rooted by a defined symbol in some right-hand side of a pair or usable rule.

Definition 18 (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 $u \to v \in \mathcal{P}$, u is argument μ-normalized. The* basic usable CS-arguments *for a function symbol $f \in \mathcal{F}$ are defined as $\mathbf{UA}_\mu(f, \mathcal{R}, \mathcal{P}) = \{i \in \mu(f) \mid \exists u \to v \in \mathcal{P} \cup \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$, $\exists p, p' \in \mathcal{P}os^\mu(v)$ s.t. $root(v|_{p'}) = f$, $root(v|_p) \in \mathcal{D}$, $p'.i \leqslant p$, $u \not\trianglerighteq_\mu v|_p\}$.*

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: $\mu'(f) \subseteq \mu(f)$ for all $f \in \mathcal{F}$.

The following proposition is the context-sensitive version of [Fer05, Lemma 5].

Proposition 11 *Let $(\mathcal{R}, \mu)$ be a CS-TRS and $\mathcal{P}$ be a set of pairs of terms s.t. for all $u \to v \in \mathcal{P}$, u is argument μ-normalized and $\mathcal{P} \cup \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ is μ-conservative. Let $l \to r \in \mathcal{P} \cup \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ be such that $\sigma(r) \overset{>\Lambda}{\hookrightarrow}{}^*_{\mathsf{i}\,\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P})} t$ 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 $p'.k \leqslant p$, we have that $k \in \mathbf{UA}_\mu(root(t|_{p'}), \mathcal{R}, \mathcal{P})$.*

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 $x \in \mathcal{V}ar^\mu(l), \sigma(x) \in \mathsf{NF}_\mu(\mathcal{R})$. Since the rule $l \to r$ is conservative (that is $\mathcal{V}ar^\mu(r) \subseteq \mathcal{V}ar^\mu(l)$), we have that for all $x \in \mathcal{V}ar^\mu(r), \sigma(x) \in \mathsf{NF}_\mu(\mathcal{R})$. It follows that p is a nonvariable (μ-replacing) position of r, i.e. $p \in \mathcal{P}os^\mu_\mathcal{F}(r)$. Thus, $\mathrm{root}(r|_p) \in \mathcal{D}$ and the result follows by Definition 18.

If $n > 0$, then there is a term s such that $\sigma(r) \overset{>\Lambda}{\hookrightarrow}{}^*_{\mathsf{i}} s$ and $s \overset{>\Lambda}{\hookrightarrow}_{\mathsf{i}} t$ 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 $k \in \mathbf{UA}_\mu(root(t|_{p'}), \mathcal{R}, \mathcal{P})$, for all $q < p'.k \leqslant p$. If $s \overset{>\Lambda}{\hookrightarrow}_{\mathsf{i}} t$, then $s|_q = \sigma'(l')$ and $t|_q = \sigma'(r')$, for some rule $l' \to r' \in \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ and substitution σ' s.t. $\sigma'(l')$ is argument μ-normalized. This implies that every innermost redex of $t|_q$ occurs at a position $p'' \in \mathcal{P}os^\mu(r')$ s.t. $root(r'|_{p''}) \in \mathcal{D}$ (since the rule $l' \to r'$ is conservative we have that for all $x \in \mathcal{V}ar^\mu(r'), \sigma(x) \in \mathsf{NF}_\mu(\mathcal{R})$) and $l' \not\trianglerighteq_\mu r'|_{p''}$(otherwise, $\sigma'(l')$ would not be an innermost redex of

s. By definition, when $p'' > \Lambda$, $p'.k \leqslant p''$, $k \in \mathbf{UA}_\mu(root(t|_{q.p'}), \mathcal{R}, \mathcal{P})$ which is equivalent to what we needed to prove ($k \in \mathbf{UA}_\mu(root(t|_{p'}), \mathcal{R}, \mathcal{P})$, for all $q < p'.k \leqslant p$). □

Corollary 11 in [Fer05] suggests that innermost μ-termination could be proved by using a μ'-reduction ordering for μ' given by $\mu'(f) = \mathbf{UA}_\mu(f, \mathcal{R}, \mathcal{P})$ for all $f \in \mathcal{F}$. This is true for μ'-conservative CS-TRSs, as the following theorem shows.

Theorem 12 *A μ-conservative CS-TRS $(\mathcal{R}, \mu)$ is innermost μ-terminating if there is a μ'-reduction ordering $\succ$ s.t. $\mathcal{R} \subseteq \succ$, where for all symbol $f \in \mathcal{F}$, $\mu'(f) = \mathbf{UA}_\mu(f, \mathcal{R}, R)$.*

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: $s_1 \hookrightarrow_{\mathsf{i}} s_2 \hookrightarrow_{\mathsf{i}} s_3 \hookrightarrow_{\mathsf{i}} \cdots$ (without loss of generality). By Proposition 11 (where we let $\mathcal{P} = \mathcal{R}$), every step $s_j \stackrel{>\Lambda}{\hookrightarrow}_{\mathsf{i}} s_{j+1}$ at position p satisfies that $p'.k \leqslant p$, $k \in \mathbf{UA}_\mu(root(s_j|_{p'}), \mathcal{R}, \mathcal{P})$. Since $\mathcal{R} \subseteq \succ$ and $\succ$ is stable and μ'-monotonic, $s_j \succ s_{j+1}$ holds. Therefore, there is an infinite $\succ$-decreasing sequence of terms $s_1 \succ s_2 \succ \cdots \succ s_n \succ \cdots$ which contradicts the well-foundedness of $\succ$.

□

Since μ-reduction orderings characterize termination of *CSR* we have the following corollary.

Corollary 4 *Let $\mathcal{R}$ be a μ-conservative TRS for $\mu \in M_{\mathcal{R}}$. Let μ' be given by $\mu'(f) = \mathbf{UA}_\mu(f, \mathcal{R}, R)$ for every $f \in \mathcal{F}$. If $\mathcal{R}$ is innermost μ'-terminating, then $\mathcal{R}$ is innermost μ-terminating.*

Example 18 *Consider the TRS $\mathcal{R}$:*

$$\begin{array}{rcl} \mathtt{f}(\mathtt{a}, \mathtt{b}, x) & \to & \mathtt{f}(x, x, x) \\ \mathtt{c} & \to & \mathtt{a} \\ \mathtt{c} & \to & \mathtt{b} \end{array}$$

together with $\mu(\mathtt{f}) = \{1, 3\}$. By using $\mu'(f) = \mathbf{UA}_\mu(f, \mathcal{R}, \mathcal{P})$ for every $f \in \mathcal{F}$ we obtain $\mu'(\mathtt{f}) = \varnothing$. The pair $\mathtt{f}(\mathtt{a}, \mathtt{b}, x) \to \mathtt{f}(x, x, x)$ cannot form a cycle now, thus easily concluding the μ'-termination of $\mathcal{R}$ and, by Corollary 4, 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 12 and Corollary 4.

Example 19 *Consider again the TRS $\mathcal{R}$ in Example 18 but now together with $\mu(\mathtt{f}) = \{1, 2\}$. If we try to apply Corollary 4 to prove innermost μ-termination* *of $\mathcal{R}$, we obtain $\mu'(\mathtt{f}) = \varnothing$ and (as discussed in Example 18) the CS-dependency graph has no cycle thus concluding the innermost μ-termination of $\mathcal{R}$. However,$\mathcal{R}$ is* not *innermost μ-terminating:*

$$\underline{\mathtt{f(a,b,c)}} \hookrightarrow_{\mathsf{i}} \mathtt{f(\underline{c},c,c)} \hookrightarrow_{\mathsf{i}} \mathtt{f(a,\underline{c},c)} \hookrightarrow_{\mathsf{i}} \underline{\mathtt{f(a,b,c)}} \hookrightarrow_{\mathsf{i}} \cdots$$

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

9.1 Relaxing Monotonicity with CS-DPs

Fernández's criterion was also adapted to deal with proofs of termination of rewriting using dependency pairs, what allows us using reduction pairs instead of reduction orderings in proofs of termination.

In previous sections, we have shown how to prove innermost termination of *CSR* by using ICSDPs. Now, we can adapt the use of CS-usable arguments to be applied in proofs of innermost μ-termination with ICSDPs. We do that by providing a new processor for dealing with innermost μ-termination problems.

Theorem 13 *Let $\tau = (\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$ be a CS problem. Let $\mu_{\mathcal{A}}(f) = \mathbf{UA}_\mu(f, \mathcal{R}, \mathcal{P})$ for all $f \in \mathcal{F} \cup \mathcal{G} \cup \mathcal{H}$ and $(\gtrsim, \sqsupset)$ be a $\mu_{\mathcal{A}}$-reduction pair such that*

1. *$\mathcal{P}$ and $\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P})$ are μ-conservative,*
2. *$\mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}) \subseteq \gtrsim$ and $\mathcal{P} \subseteq \gtrsim \cup \sqsupset$,*

Let $\mathcal{P}_\sqsupset = \{u \to v \in \mathcal{P} \mid u \sqsupset v\}$. Then, the processor Proc_{Fer} given by

$$\mathsf{Proc}_{Fer}(\tau) = \begin{cases} \{(\mathcal{P} \setminus \mathcal{P}_\sqsupset, \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}), \varnothing, \mu_{\mathcal{A}}, \mathbf{i})\} & \textit{if (1) and (2) hold} \\ \{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})\} & \textit{otherwise} \end{cases}$$

is sound.

Proof.

We have to prove that every infinite minimal innermost $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chain introduces an infinite minimal innermost $(\mathcal{P} \setminus \mathcal{P}_\sqsupset, \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}), \mathcal{S}, \mu_{\mathcal{A}}, \mathbf{i})$-chain. We proceed by contradiction. Assume that there is an infinite minimal innermost $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chain A, but that there is no infinite minimal innermost $(\mathcal{P} \setminus \mathcal{P}_\sqsupset, \mathbf{U}_0(\mathcal{R}, \mu, \mathcal{P}), \mathcal{S}, \mu_{\mathcal{A}}, \mathbf{i})$-chain. Due to the finiteness of $\mathcal{P}$, and since $\mathcal{P}$ is conservative, we have $\mathcal{P}_{\mathcal{X}} = \varnothing$. Thus, we can assume that there is $\mathcal{Q} \subseteq \mathcal{P}$ such that A has a tail B

$$\sigma(u_1) \hookrightarrow_{\mathcal{Q},\mu} t_1 \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} \sigma(u_2) \hookrightarrow_{\mathcal{Q},\mu} t_2 \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathsf{i}} \sigma(u_3) \hookrightarrow_{\mathcal{Q},\mu} \cdots$$

for some substitution σ, where all pairs in $\mathcal{Q}$ are infinitely often used, and, for all $i \geq 1$, since all $u_i \to v_i \in \mathcal{P}$ are conservative $u_i \to v_i \in \mathcal{Q}_{\mathcal{G}}$ (i.e. $\mathcal{P}_{\mathcal{X}} = \mathcal{Q}_{\mathcal{X}} = \varnothing$), then $t_i = \sigma(v_i)$ such that for all $i > 0$, $\sigma(u_i)$ is argument μ-normalized and $\sigma(v_i)$ is innermost $(\mathcal{R}, \mu)$-terminating. By Proposition 9 and

11, every innermost step in the sequence $t_i \overset{!}{\hookrightarrow}_{\mathcal{R},\mu,\mathrm{i}} \sigma(u_{i+1})$ is performed at a $\mu_{\mathcal{A}}$-replacing position by means of a conservative rule in $\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P})$:

$$\sigma(u_1) \hookrightarrow_{\mathcal{Q}_{\mathcal{G}},\mu_{\mathcal{A}}} t_1 \overset{!}{\hookrightarrow}_{\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu_{\mathcal{A}},\mathrm{i}} \sigma(u_2) \hookrightarrow_{\mathcal{Q}_{\mathcal{G}},\mu_{\mathcal{A}}} t_2 \overset{!}{\hookrightarrow}_{\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu_{\mathcal{A}},\mathrm{i}} \sigma(u_3) \hookrightarrow \cdots$$

Since $u_i \ (\gtrsim \cup \sqsupset) \ v_i$ for all $u_i \to v_i \in \mathcal{Q} \subseteq \mathcal{P}$, by stability of $\gtrsim$ and $\sqsupset$, we have $\sigma(u_i) \ (\gtrsim \cup \sqsupset) \ \sigma(v_i)$ for all $i \geq 1$.

No pair $u \to v \in \mathcal{Q}$ satisfies that $u \sqsupset v$. Otherwise, we get a contradiction by considering that since all pairs $\in \mathcal{P}$ are conservative $u_i \to v_i \in \mathcal{Q}_{\mathcal{G}}$, then $t_i = \sigma(v_i) \overset{!}{\hookrightarrow}_{\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu_{\mathcal{A}},\mathrm{i}} \sigma(u_{i+1})$ and $t_i \gtrsim \sigma(u_{i+1})$. Since we have $\sigma(u_i) \ (\gtrsim \cup \sqsupset) \ \sigma(v_i) = \sigma(v_i) = t_i$, by using transitivity of $\gtrsim$ and compatibility between $\gtrsim$ and $\sqsupset$, we conclude that $\sigma(u_i) \ (\gtrsim \cup \sqsupset) \ \sigma(u_{i+1})$. Since $u \to v$ occurs infinitely often in B, there is an infinite set $\mathcal{I} \subseteq \mathbb{N}$ such that $\sigma(u_i) \sqsupset \sigma(u_{i+1})$ for all $i \in \mathcal{I}$. And we have $\sigma(u_i) \ (\gtrsim \cup \sqsupset) \ \sigma(u_{i+1})$ for all other $u_i \to v_i \in \mathcal{Q}$. Thus, by using the compatibility conditions of the μ-reduction pair, we obtain an infinite decreasing $\sqsupset$-sequence which contradicts well-foundedness of $\sqsupset$. Therefore, $\mathcal{Q} \subseteq (\mathcal{P} \setminus \mathcal{P}_{\sqsupset})$. Since $\mu_{\mathcal{A}} \subseteq \mu$ and $\mathsf{NF}_{\mu_{\mathcal{A}}}(\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P})) \supseteq \mathsf{NF}_{\mu}(\mathcal{R})$, we have that $\sigma(u_i) \in \mathsf{NF}_{\mu_{\mathcal{A}}}(\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}))$. By Proposition 9, innermost $(\mathcal{R},\mu)$-termination of $\sigma(v_i)$ implies innermost $(\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu)$- termination of $\sigma(v_i)$ for all $i \geq 1$ and by Proposition 11, innermost $(\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu)$-termination of $\sigma(v_i)$ implies innermost $(\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu_{\mathcal{A}})$-termination, so we get that innermost $(\mathcal{R},\mu)$-termination of $\sigma(v_i)$ implies innermost $(\mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu_{\mathcal{A}})$-termination. Hence, B is an infinite minimal innermost $(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathbf{U}_0(\mathcal{R},\mu,\mathcal{P}),\mu_{\mathcal{A}})$-chain, thus leading to a contradiction.

□

Corollary 4 can be generalized to (certain) non-μ-conservative CS-TRSs thanks to Theorem 13. Now, for a given CS-TRS $(\mathcal{R},\mu)$ that satisfies the conditions of Theorem 13, we can prove its innermost μ-termination by relaxing μ-monotonicity requirements for each cycle.

10 Narrowing Transformation

Although, function TCap provides a good approximation of the graph, it can lead to *overestimate* the arcs that connect two dependency pairs. As already observed by Arts and Giesl for the standard and innermost case [AG00], in our setting the overestimation comes when a (noncollapsing) pair $u_i \to v_i$ is followed in a chain by a second one $u_{i+1} \to v_{i+1}$ and v_i and u_{i+1} are not directly unifiable, i.e., at least one innermost μ-rewriting step is needed to innermost μ-reduce $\sigma(v_i)$ to $\sigma(u_{i+1})$. Then, the innermost μ-reduction from $\sigma(v_i)$ to $\sigma(u_{i+1})$ requires at least one step, i.e., we always have $\sigma(v_i) \hookrightarrow_{\mathcal{R},\mu^\sharp} \sigma(v_i') \hookrightarrow^*_{\mathcal{R},\mu^\sharp} \sigma(u_{i+1})$. Furthermore, we could discover that v_i has *no μ-narrowings*. In this case, we know that no innermost chain starts from $\sigma(v_i)$. A restriction that have to be taken into account when μ-narrowing a noncollapsing pair $u \to v$ is that the

μ-replacing variables in v have to be μ-replacing in u as well (this corresponds with the notion of conservativeness), but furthermore, they cannot be both μ-replacing and non-μ-replacing at the same time. This corresponds to the following definition.

Definition 19 (Strongly Conservative [GLU08]**)** *Let $\mathcal{R}$ be a TRS and $\mu \in M_{\mathcal{R}}$. A rule $l \to r$ is strongly μ-conservative if it is μ-conservative and $\mathcal{V}ar^{\mu}(l) \cap \mathcal{V}ar^{\not\mu}(l) = \mathcal{V}ar^{\mu}(r) \cap \mathcal{V}ar^{\not\mu}(r) = \varnothing$.*

In [AGL10], we define the μ-narrowing processor. Of course, μ-narrowing can also be used in proofs of innermost termination of *CSR*. In the standard setting, when using narrowing for proving innermost termination we do not require that the right-hand side of the dependency pair to be narrowed is linear since the involved substitution σ is normalized. However, in the context-sensitive setting, if the pair to be μ-narrowed is *not strongly μ-conservative*, we can not ensure that the variables on the right-hand side are μ-normalized so *we also have to demand linearity*. When dealing with innermost narrowing in context-sensitive rewriting we can drop the linearity condition if the pair to be μ-narrowed is strongly conservative since all μ-replacing variables in the right-hand side of a pair are instantiated to μ-normal form and μ-reductions can not take place on them.

Theorem 14 (Innermost Narrowing processor) *Let $\tau = (\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$ be a CS problem. Let $u \to v \in \mathcal{P}$ be such that*

1. *for all $u' \to v' \in \mathcal{P}$ (with possibly renamed variables), v and u' do not unify or they unify by some mgu θ such that one of the terms $\theta(u)$ or $\theta(u')$ is not a μ-normal form.*

Let $\mathcal{Q} = (\mathcal{P} - \{u \to v\}) \cup \{u' \to v' \mid u' \to v'$ is a μ-narrowing of $u \to v\}$. Then, the processor Proc_{Inarr} given by

$$\mathsf{Proc}_{Inarr}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i}) = \begin{cases} \{(\mathcal{Q}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})\} & \text{if (1) holds} \\ \{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})\} & \text{otherwise} \end{cases}$$

is

1. *sound whenever $u \to v$ is strongly conservative, and*

2. *complete in all cases.*

PROOF.

We have to prove that there is an infinite minimal innermost $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chain iff there is an infinite minimal innermost $(\mathcal{Q}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chain. We prove that for every minimal innermost $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chain "$\ldots, u_1 \to v_1, u \to v, u_2 \to v_2, \ldots$", there is an innermost μ-narrowing v' of v with the mgu θ such that "$\ldots, u_1 \to v_1, \theta(u) \to v', u_2 \to v_2, \ldots$" is also a minimal innermost $(\mathcal{Q}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chain.

If "$\ldots, u_1 \to v_1, u \to v, u_2 \to v_2, \ldots$" is a minimal innermost $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu, \mathbf{i})$-chain, then there is an substitution σ such that for all pairs $s \to t$ in the chain,

1. if $s \to t \in \mathcal{P}_{\mathcal{G}}$, then $\sigma(t)$ is μ-terminating and it μ-reduces innermost to the instantiated left-hand side $\sigma(s')$ of the next pair $s' \to t'$ in the chain

2. if $s \to t = s \to x \in \mathcal{P}_{\mathcal{X}}$, then $\sigma(x) \overset{\Lambda}{\hookrightarrow}^{*}_{\mathcal{S}_{\rhd_\mu},\mu} \circ \overset{\Lambda}{\hookrightarrow}_{\mathcal{S}_\sharp,\mu} t$ and t, which is innermost μ-terminating, μ-reduces innermost to the instantiated left-hand side $\sigma(s')$ of the next pair $s' \to t'$ in the chain.

3. all instantiated left-hand sides are μ-normal forms w.r.t. $(\mathcal{R}, \mu)$.

Assume that σ is a substitution satisfying the above requirements and such that the length of the sequence $\sigma(v) \hookrightarrow^{*}_{\mathcal{R},\mu,\mathsf{i}} \sigma(u_2)$ is *minimal*.

Note that $\sigma(v) \neq \sigma(u_2)$. Otherwise σ would unify v and u_2, where both, u and v_2 are μ-normal forms, hence, there is a term q such that $\sigma(v) \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} q \hookrightarrow^{*}_{\mathcal{R},\mu,\mathsf{i}} \sigma(u_2)$.

The reduction $\sigma(v) \hookrightarrow_{\mathcal{R},\mu,\mathsf{i}} q$ cannot take place within a binding of σ because $u \to v$ is strongly conservative. Hence, $\sigma(u)$ would not be a μ-normal form which violates the last condition for σ. In the innermost case, we do not have to demand linearity since μ-replacing variables in v come from being replacing in u (strongly conservative) and they are instantiated to μ-normal forms and no one can be reduced in v. The remainder of the proof is completely analogous to the noninnermost case.

□

Example 20 *Consider the following example:*

$$\begin{array}{rcl} \mathtt{f}(\mathtt{s}(x)) & \to & \mathtt{f}(\mathtt{p}(\mathtt{s}(x))) \\ \mathtt{p}(\mathtt{s}(x)) & \to & x \end{array}$$

together with $\mu(\mathtt{f}) = \mu(\mathtt{s}) = \{1\}$ *and* $\mu(\mathtt{p}) = \varnothing$.

The only ICSDP that could generate a cycle is $\mathtt{F}(\mathtt{s}(x)) \to \mathtt{F}(\mathtt{p}(\mathtt{s}(x)))$. *However since the right-hand side* $\mathtt{F}(\mathtt{p}(\mathtt{s}(x)))$ *does not unify with any (renamed) the left-hand side (including itself) and the pair is strongly conservative, we can apply* μ*-narrowing. Therefore, the pair can be* μ*-narrowed at position 1 (notice that* $\mu(f) = \mu(\mathtt{F}) = \{1\}$*) by using the rule* $\mathtt{p}(\mathtt{s}(x)) \to x$*. Then, the pair is transformed into the pair* $\mathtt{F}(\mathtt{s}(y)) \to \mathtt{F}(y)$ *that can be easy disregarded by using the subterm criterion*[1]*.*

11 Experiments

We have implemented the techniques described in the previous sections as part of the tool MU-TERM [AGL+10]. In order to evaluate the techniques which are reported in this paper we have made some benchmarks. We have considered the examples in the Termination Problem Data Base[2] (TPDB).

---

[1] Instead of using in the proof a polynomial interpretation with rationals, like MU-TERM or matrix interpretations like AProVE.

[2] `http://www.termination-portal.org/wiki/TPDB`

	ICSDPs	Transformations
YES score	95/109	60/109
YES average time	0.7 sec.	1.5 sec.

Table 1: Comparative in proofs of termination of innermost *CSR*

	C	GM	iGM
YES score	33	57	42

Table 2: Comparing transformations for proving termination of innermost *CSR*

11.1 Proving Termination of Innermost *CSR*: Direct Techniques vs. Transformations

Although there is no special TPDB category for innermost termination of *CSR* (yet) we have used the examples used in the *CSR*category in order to test our techniques for proving termination of innermost *CSR*. The TPDB v7.0.2 contains 109 examples of CS-TRSs. In order to evaluate our direct techniques in comparison with the transformational approach of [GM02b, GM04, Luc01a], where termination of innermost *CSR* for a CS-TRS ($\mathcal{R}$, μ) is proved by proving innermost termination of a transformed TRS $\mathcal{R}^{\mu}_{\Theta}$, where Θ specifies a particular transformation (see [GM02a, GM02b] for a survey on this topic), we have transformed the set of examples by using the transformations that are correct for proving innermost termination of *CSR*: Giesl and Middeldorp's correct transformations for proving termination of innermost *CSR*, see [GM02b], although we use the 'authors-based' notation introduced in [Luc06]: GM and C for transformations 1 and 2 for proving termination of *CSR* introduced in [GM04], and iGM for the specific transformation for proving termination of innermost *CSR* introduced in [GM02b]. Then we have proved innermost termination of the set of examples with AProVE [GST06], which is able to prove innermost termination of standard rewriting. The results are summarized in Table 1 and 11.1. Further details can be found here:

`http://www.dsic.upv.es/~balarcon/iCSR/benchmarks.html`

These are the first known benchmarks comparing not only transformational techniques vs. direct (DP-based) techniques, but also the existing correct transformations for proving innermost termination of *CSR* among them. They show that, quite surprisingly, the iGM transformation (which is in principle the more suitable one for proving innermost termination of *CSR*) obtains worse results than GM (in the average).

In [AL07], we obtained 70 over 90 successful proofs against 44 for transformations (it was used the TPDB v3.2). Obviously, the use of ICSDPs were imposed without doubts for proving innermost termination of *CSR*. Moreover, now, with the recent developments of mu-term embracing the DP-framework, mu-term would solve 77 over those 90 of the previous version (and 18 over the new 19 included in the last one). Therefore, from the results in Table 1, it is clear that using transformations for proving termination of innermost *CSR* makes no sense after introducing the ICSDP framework.

11.2 Proving Innermost Termination of *CSR*: Relaxing Monotonicity Requirements

For our experiments about proving termination of innermost *CSR* by means of a new replacement map which imposes less monotonicity requirements we have used the set of examples mentioned in Section 11.1.

We have implemented the use of Theorem 13 to deal with nonconservative systems (but conservative cycles). MU-TERM tries to solve each μ-conservative cycle (with associated μ-conservative usable rules) by using CS-usable arguments as the new replacement map. This implementation of MU-TERM succeed over the same 95 examples, the same number of examples that we had already solved using ICSDPs. The time average rates has no exhibit substantial differences. Further details can be found here:

`http://www.dsic.upv.es/~balarcon/iCSR_UA/benchmarks.html`

Although no improvement over the practical use of ICSDPs explained in previous subsection is shown, we expect that in the future, when we implement nonmonotonic orderings in our termination tool MU-TERM we take advantage of this technique.

Moreover, we have implemented the use of Corollary 11 in [Fer05] to prove innermost termination of TRSs by proving μ-termination of the CS-TRS obtained after using the usable arguments as replacement map (this was one of the main results in Fernández's paper). The relevance of this result in practice had not been tested yet as no implementation of Fernández's results was available (to our knowledge). In order to evaluate it, we have considered the examples used in the *innermost category*. There are 358 examples. Using usable arguments (we call this tool MU-TERM UA), MU-TERM succeeds in 158 examples. However, we have also implemented the use of (standard) dependency pairs for proving innermost termination (according to [AG00, Theorem 37]) together with the narrowing refinement (we call this tool MU-TERM iDPs) and we are able to prove 199 examples, including all examples solved with Fernández's criterion.

Therefore, it seems that using her result to prove innermost termination of rewriting is not as good idea (at least with the considered set of examples) since we loose some examples and the average time is worse. The results are summarized in Table 3. Further details can be found in:

`http://www.dsic.upv.es/~balarcon/UA/benchmarks.html`

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.

	mu-term UA	mu-term iDPs
YES score	158	199
YES average time	4.87 sec.	3.31 sec.

Table 3: Benchmarks for innermost termination of rewriting

11.3 Transforming CS-dependency Pairs

We have also implemented innermost μ-narrowing in mu-term. Due to the possibility of performing an unbounded number of narrowing steps, the μ-narrowing transformation could be infinite (this also happens in the standard approach). In order to implement the transformation, we have chosen to use *one-step μ-narrowing* only if the innermost context-sensitive dependency graph obtained has less cycles and arcs than the original one. One of the best advantages of using μ-narrowing lies in the possibility of dismissing some CS-DPs if the right-hand sides do not unify with any left-hand side of another (possible renamed) CS-DP and they have no μ-narrowings.

11.4 Termination Competition

Thanks to the new developments reported in this paper and in [AGL10, GL10], mu-term 5.07 has proven to be the most powerful tool for proving termination of *CSR* in the *context-sensitive* subcategory of the 2007, 2009 and 2010 editions of the International Competition of Termination Tools[3].As we have commented, under some conditions, termination of *CSR* and termination of innermost *CSR* coincide [GM02b, GL02b]. For this reason, one of the most important aspect of innermost *CSR* is its use for proving termination of *CSR* as part of the CSDP framework. We switch from termination of *CSR* to termination of innermost *CSR*whenever termination is equivalent, for which we can apply the existing processors more successfully. Actually, we proceed like that in $30-50\%$ of the *CSR* termination problems which are proved by mu-term 5.0.

12 Conclusions

The results of this paper are revised and extended versions of the results published in [AL07, AL09], having into account all improvements made in the CSDP framework in [AGL10, GL10].

[3]See `http://www.lri.fr/~marche/termination-competition/2007/`, where only AProVE and mu-term participated, and `http://termcomp.uibk.ac.at/termcomp/` where there were three more tools in the competition: AProVE, Jambox [End], and VMTL [SG09]. AProVE and mu-term solved the same number of examples but mu-term was much faster. The same situation has happened in 2010 (but without Jambox's participation).

12.1 Theoretical Contributions

We have investigated the structure of infinite innermost context-sensitive rewrite sequences starting from (strongly) minimal innermost non-μ-terminating terms (Theorem 1). This knowledge has been used to provide an appropriate definition of innermost context-sensitive dependency pair (Definition 5), and the related notion of innermost chain (Definition 8). We have proved that it can be used to characterize innermost μ-termination (Theorems 2 and 3). We have provided a suitable adaptation of Giesl et al.'s *dependency pair framework* to innermost *CSR* by defining appropriate notions of *CS problem* (Definition 9) and *CS processor* (Definition 10). In this setting we have described a number of sound and (most of them) complete CS-processors which can be used in any practical implementation of the ICSDP framework. In particular, we have introduced the notion of (estimated) *innermost context-sensitive (dependency) graph* (Definitions 12 and 15) by using functions to approximate it (Definition 14) and the associated CS processor showing how to automatically prove innermost μ-termination by means of the ICS dependency graph (Theorem 10). We have formulated the notion of basic usable rules showing how to use them in proofs of innermost termination of *CSR* (Definition 17, Theorem 11). Narrowing context-sensitive dependency pairs has also been investigated. It can also be helpful to simplify or restructure the dependency graph and eventually simplify the proof of (innermost) termination (Theorem 14). We have also shown how to relax monotonicity requirements for proving innermost termination of context-sensitive rewriting. We have adapted Fernández's approach [Fer05] to be used for proving innermost termination of context-sensitive rewriting (Theorems 12 and 13).

12.2 Applications and Practical Impact

We have implemented these ideas as part of the termination tool MU-TERM [AGIL07, Luc04]. The implementation and practical use of the developed techniques yield a novel and powerful framework which improves the current state-of-the-art of methods for proving termination of *CSR*. Actually, ICSDPs were an essential ingredient for MU-TERM in winning the context-sensitive subcategory of the 2007, 2009 and 2010 competitions of termination tools.

Up to our contributions, no direct method has been proposed to prove termination of innermost *CSR*. So this is the first proposal of a direct method for proving termination of innermost *CSR*. We have extended the DP framework [GTS04, GTSF06] to prove innermost termination of TRSs to innermost *CSR* (thus also extending [AGL07]). Our benchmarks show that the use of ICSDPs dramatically improves the performance of existing (transformational) methods for proving termination of innermost *CSR*.

12.3 Future Work

As remarked in the introduction, we aim at applying all previous developments to deal with termination of Maude programs. Since its computational mechanism can be thought of as kind of "context-sensitive call by value", we believe that our research is a essential contribution to the development of tools for proving termination of Maude programs.

References

[AAC+08] E. Albert, P. Arenas, M. Codish, S. Genaim, G. Puebla, and D. Zanardini. Termination Analysis of Java Bytecode. FMOODS 2–18. 2008.

[Ala08] B. Alarcón. Innermost Termination of Context-Sensitive Rewriting. Master Thesis, DSIC, Universidad Politécnica de Valencia. December 2008.

[AEF+08] B. Alarcón, F.Emmes, C. Fuhs, J. Giesl, R. Gutiérrez, S. Lucas, P. Schneider-Kamp, and R. Thiemann. Improving context-sensitive dependency pairs. In I. Cervesato, H. Veith and A. Voronkov, editors, *Proc. of 15th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR'08*, LNAI 5330:636–651, Springer-Verlag, 2008.

[AG00] T. Arts and J. Giesl. Termination of Term Rewriting Using Dependency Pairs. *Theoretical Computer Science*, 236(1–2):133–178, 2000.

[AGIL07] B. Alarcón, R. Gutiérrez, J. Iborra, and S. Lucas. Proving Termination of Context-Sensitive Rewriting with MU-TERM. *Electronic Notes in Theoretical Computer Science*, 188:105–115, 2007.

[AGL06] B. Alarcón, R. Gutiérrez, and S. Lucas. Context-Sensitive Dependency Pairs. In S. Arun-Kumar and N. Garg, editors, *Proc. of XXVI Conference on Foundations of Software Technology and Theoretical Computer Science, FST&TCS'06*, LNCS 4337:297–308, Springer-Verlag, 2006.

[AGL07] B. Alarcón, R. Gutiérrez, and S. Lucas. Improving the Context-Sensitive Dependency Graph. *Electronic Notes in Theoretical Computer Science*, 188:91–103, 2007.

[AGL10] B. Alarcón, R. Gutiérrez, and S. Lucas. Context-sensitive dependency pairs. Information and Computation 208(8), 922–968, 2010.

[AGL+10] B. Alarcón, R. Gutiérrez, S. Lucas and R. Navarro-Marset. Proving Termination Properties with MU-TERM. *Proc. of Thirteenth*

International Conference on Algebraic Methodology And Software Technology, AMAST2010, to appear 2010.

[AL07] B. Alarcón 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, 2007.

[AL09] B. Alarcón, S. Lucas.
Using Context-Sensitive Rewriting for Proving Innermost Termination of Rewriting. Electronic Notes in Theoretical Computer Science, 248:3–17, 2009.

[AM93] G. Aguzzi and U. Modigliani. Proving termination of logic programs by transforming them into equivalent term rewriting systems. *Proc. of the 13th Conference on Foundations of Software Technology and Theoretical Computer Science,FST&TCS93*, LNCS 761: 114–124, Springer-Verlag, 1993.

[BM06] R. Bruni and J. Meseguer. Semantic foundations for generalized rewrite theories. *Theoretical Computer Science* 351(1):386–414, 2006.

[BMS05] A. R. Bradley, Z. Manna, and H. B. Sipma. Termination of polynomial programs. *Proc. of the 6th International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI 05*, LNCS, 3385:113–129, Springer-Verlag, 2005.

[BN98] F. Baader and T. Nipkow. Term Rewriting and All That. Cambridge University Press, 1998.

[Bor03] C. Borralleras. Ordering-based methods for proving termination automatically. PhD Thesis, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, May 2003.

[CDE+07] 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, Springer-Verlag, 2007.

[CLS05] M. Codish, V. Lagoon, and P. Stuckey. Testing for termination with monotonicity constraints. *Proc. of the 21th International Conference on Logic Programming, ICLP 05*, LNCS 3668: 326–340, Springer-Verlag, 2005.

[CPR06] B. Cook, A. Podelski, and A. Rybalchenko. Termination proofs for systems code. *Proc. of the ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 06*, pages 415–426. ACM Press, 2006.

[CS02] M. Colon and H. B. Sipma. Practical methods for proving program termination. *Proc. of the 14th International Conference on Computer Aided Verification, CAV 02*, LNCS 2034: 442–454, Springer-Verlag, 2002.

[CT99] M. Codish and C. Taboch. A semantic basis for the termination analysis of logic programs. *The Journal of Logic Programming*, 41(1):103–123, 1999.

[DD94] D. De Schreye and S. Decorte. Termination of logic programs: The neverending story. *Journal of Logic Programming*, 19/20:199–260, 1994.

[Der04] N. Dershowitz. Termination by Abstraction. In B. Demoen and V. Lifschitz, editors, *Proc. of 20th International Conference on Logic Programming, ICLP'04*, LNCS 3132:1–18, Springer-Verlag, 2004.

[DL01] N. Dershowitz, N. Lindenstrauss, Y. Sagiv, and A. Serebrenik. A general framework for automatic termination analysis of logic programs. *Appl. Algebra Eng. Commun. Comput.*, 12(1/2):117–156, 2001.

[DLM+04] F. Durán, S. Lucas, C. Marché, J. Meseguer, and X. Urbain. Proving Termination of Membership Equational Programs. In *Proc. of 2004 ACM SIGPLAN Symposium on Partial Evaluation and Program Manipulation, PEPM'04*, pages 147–158, Verona, Italy. ACM Press, 2004.

[DLM+08] F. Durán, S. Lucas, C. Marché, J. Meseguer, and X. Urbain. Proving Operational Termination of Membership Equational Programs. *Higher-Order and Symbolic Computation*, 21(1-2):59–88, 2008.

[DS02] D. De Schreye and A. Serebrenik. Acceptability with general orderings. *Computational Logic: Logic Programming and Beyond*, LNCS 2407: 187–210, Springer-Verlag, 2002.

[Emm08] F. Emmes. Automated Termination Analysis of Context-Sensitive Term Rewrite Systems. Diplomarbeit im Studiengang Informatik, RWTH Aachen, 2008.

[End] Endrullis, J.: Jambox, Automated Termination Proofs for String and Term Rewriting. Available at `http://joerg.endrullis.de/jambox.html`

[Fer05] M. L. Fernández. Relaxing monotonicity for innermost termination. Information Processing Letters 93(3):117–123, 2005.

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

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

[GAO02] J. Giesl, T. Arts, and E. Ohlebusch. Modular Termination Proofs for Rewriting Using Dependency Pairs. *Journal of Symbolic Computation*, 34(1):21–58, 2002.

[Gie95] J. Giesl. Termination analysis for functional programs using term orderings. *Proc. of the 2nd International Static Analysis Symposium, SAS 95*, LNCS 983:154–171, Springer-Verlag, 1995.

[GL02a] B. Gramlich and S. Lucas. Simple Termination of Context-Sensitive Rewriting. In B. Fischer and E. Visser, editors, *Proc. of III ACM SIGPLAN Workshop on Rule-Based Programming, RULE'02*, pages 29–41. ACM Press, 2002.

[GL02b] 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*, pages 50–61. ACM Press, 2002.

[GL10] R. Gutiérrez, S. Lucas. Proving Termination in the Context-Sensitive Dependency Pair Framework. *Proc. of the 8th International Workshop on Rewriting Logic and its Applications, WRLA'10*. LNCS, 6381:19–35, Springer-Verlag, 2010.

[GLU08] R. Gutiérrez, S. Lucas, and X. Urbain. Usable Rules for Context-Sensitive Rewrite Systems. In A. Voronkov, editor, *Proc. of the 19th International Conference on Rewriting Techniques and Applications, RTA'08*, LNCS, 5117:126–141, Springer-Verlag, 2008.

[GM02a] J. Giesl and A. Middeldorp. Innermost termination of context-sensitive rewriting. Aachener Informatik-Berichte (AIBs), RWTH Aachen, 2002. http://citeseer.ist.psu.edu/giesl02innermost.html

[GM02b] J. Giesl 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, 2003.

[GM04] J. Giesl and A. Middeldorp. Transformation techniques for context-sensitive rewrite systems. *Journal of Functional Programming*, 14(4):379–427, 2004.

[Gra95] B. Gramlich. Abstract Relations between Restricted Termination and Confluence Properties of Rewrite Systems, *Fundamenta Informaticae*, 24:3–23, 1995.

[Gra96] B. Gramlich. On Proving Termination by Innermost Termination. *Proc. 7th Int. Conf. on Rewriting Techniques and Applications Proc. of RTA'96*, LNCS 1103:93–107, Springer-Verlag, 1996.

[GSST06] J. Giesl, S. Swiderski, P. Schneider-Kamp, and R. Thiemann. Automated termination analysis for Haskell: From term rewriting to programming languages. *Proc. of the 17th International Conference on Rewriting Techniques and Applications, RTA 06*, LNCS 4098: 297–312, Springer-Verlag 2006.

[GST06] J. Giesl, P. Schneider-Kamp, and R. Thiemann. AProVE 1.2: Automatic Termination Proofs in the Dependency Pair Framework. In U. Furbach and N. Shankar, editors, *Proc. of Third International Joint Conference on Automated Reasoning, IJCAR'06*, LNAI 4130:281–286, Springer-Verlag, 2006. Available at `http://www-i2.informatik.rwth-aachen.de/AProVE`.

[GTS04] J. Giesl, R. Thiemann, and P. Schneider-Kamp. The Dependency Pair Framework: Combining Techniques for Automated Termination Proofs. In F. Baader and A. Voronkov, editors, *Proc. of XI International Conference on Logic for Programming Artificial Intelligence and Reasoning, LPAR'04*, LNAI 3452:301–331, Springer-Verlag, 2004.

[GTS05] J. Giesl, R. Thiemann, and P. Schneider-Kamp. Proving and Disproving Termination of Higher-Order Functions. In B. Gramlich, editor, *Proc. of 5th International Workshop on Frontiers of Combining Systems, FroCoS'05*, LNAI 3717:216–231, Springer-Verlag, 2005.

[GTSF06] J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Mechanizing and Improving Dependency Pairs. *Journal of Automatic Reasoning*, 37(3):155–203, 2006.

[GWM+00] J.A. Goguen, T. Winkler, J. Meseguer, K. Futatsugi, and J.-P. Jouannaud. Introducing OBJ. In J. Goguen and G. Malcolm, editors, *Software Engineering with OBJ: algebraic specification in action*, Kluwer, 2000.

[HM04] N. Hirokawa and A. Middeldorp. Dependency Pairs Revisited. In V. van Oostrom, editor, *Proc. of XV International Conference on Rewriting Techniques and Applications, RTA'04*, LNCS 3091: 249–268, Springer-Verlag, 2004.

[HM05] N. Hirokawa and A. Middeldorp. Automating the Dependency Pair Method. *Information and Computation*, 199:172–199, 2005.

[HM07] N. Hirokawa and A. Middeldorp. Tyrolean termination tool: Techniques and features. *Information and Computation*, 205:474-511, 2007.

[HPW92] P. Hudak, S.J. Peyton-Jones, and P. Wadler. Report on the Functional Programming Language Haskell: a non–strict, purely functional language. *Sigplan Notices*, 27(5):1–164, 1992.

[KKS98] M. R. K. Krishna Rao, D. Kapur, and R. Shyamasundar. Transformational methodology for proving termination of logic programs. *Journal of Logic Programming*, 34(1):1–42, 1998.

[LJB01] C. S. Lee, N. D. Jones, and A. M. Ben-Amram. The size-change principle for program termination. *ACM Symposium on Principles of Programming Languages, POPL 01*, pages 81–92, ACM Press, 2001.

[LMS03] V. Lagoon, F. Mesnard, and P. J. Stuckey. Termination analysis with types is more accurate. *Proc. of the 19th International Conference on Logic Programming, ICLP 03*, LNCS 2916: 254–268, Springer-Verlag, 2003.

[Luc96] S. Lucas. Termination of context-sensitive rewriting by rewriting. In F. Meyer auf der Heide and B. Monien, editors, *Proc. of 23rd. International Colloquium on Automata, Languages and Programming, ICALP'96*, LNCS 1099:122-133, Springer-Verlag, 1996.

[Luc98] S. Lucas. Context-Sensitive Computations in Functional and Functional Logic Programs. *Journal of Functional and Logic Programming*, 1998(1):1–61, 1998.

[Luc01a] S. Lucas. 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, 2001.

[Luc01b] S. Lucas. Termination of on-demand rewriting and termination of OBJ programs. In *Proc. of 3rd International Conference on Principles and Practice of Declarative Programming, Proc. of PPDP'01*, pages 82–93, ACM Press, 2001.

[Luc02] S. Lucas. Context-Sensitive Rewriting Strategies. *Information and Computation*, 178(1):293–343, 2002.

[Luc04] S. Lucas. MU-TERM: A Tool for Proving Termination of Context-Sensitive Rewriting. In V. van Oostrom, editor, *Proc. of XV International Conference on Rewriting Techniques and Applications, RTA'04*, LNCS 3091:200–209, Springer-Verlag, 2004. Available at `http://zenon.dsic.upv.es/muterm`.

[Luc06] S. Lucas. Proving Termination of Context-Sensitive Rewriting by Transformation. *Information and Computation*, 204(12):1782–1846, 2006.

[LM08] S. Lucas and J. Meseguer. Order-Sorted Dependency Pairs. In S. Antoy, editor, *Proc. of the 10th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming, PPDP'08*, pages 108–119, ACM Press, 2008.

[LM09] S. Lucas and J. Meseguer. Operational Termination of Membership Equational Programs: the Order-Sorted Way. In G. Rosu, editor, *Proc. of the 7th International Workshop on Rewriting Logic and its Applications, WRLA'08*, *Electronic Notes in Theoretical Computer Science*, 238(3):207–225, 2009.

[Mar94] M. Marchiori. Logic programs as term rewriting systems. *Proc. of the 4th International Conference on Algebraic and Logic Programming, ALP94*, LNCS 850:223–241, Springer-Verlag, 1994.

[Mar96] M. Marchiori. Proving existential termination of normal logic programs. *Proc. of the 5th International Conference on Algebraic Methodology and Software Technology, AMAST 96*, LNCS 1101: 375–390, Springer-Verlag, 1996.

[McC60] J. McCarthy. Recursive Functions of Symbolic Expressions and their Computations by Machine, Part I. *Communications of the ACM*, 3(4):184–195, 1960.

[Mid01] A. Middeldorp. Approximating dependency graphs using tree automata techniques. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proc. of the International Joint Conference on Automated Reasoning, IJCAR'01*, LNAI 2083:593–610, Springer-Verlag, 2001.

[Mid02] A. Middeldorp, Approximations for strategies and termination *Electronic Notes in Computer Science*, volume 70(6), 2002.

[NSEP92] E.G.J.M.H. Nöcker, J.E.W. Smetsers, M.C.J.D. van Eekelen, and M.J. Plasmeijer. Concurrent Clean. In E.H.L. Aarts, J. Leeuwen, and M. Rem, editors, *Proc. of Parallel Architectures and Languages Europe, PARLE'91*, LNCS 506:202–219, Springer-Verlag, 1992.

[OBEG10] C. Otto, M. Brockschmidt, C. von Essen and J. Giesl Automated Termination Analysis of Java Bytecode by Term Rewriting. *Proc of 21st International Conference on Rewriting Techniques and Applications, RTA 2010*, pages 259–276, 2010.

[Ohl02] E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer-Verlag, 2002.

[SG09] Schernhammer, F., Gramlich, B.: VMTL - A Modular Termination Laboratory. In R. Treinen, editor, *Proc. of the 20th International Conference on Rewriting Techniques and Applications, RTA'09*. LNCS, 5595:285–294. Springer-Verlag, 2009.

[SGN09] P. Schneider-Kamp, J. Giesl and M. T. Nguyen. The Dependency Triple Framework for Termination of Logic Programs. *Proc. of the 19th International Symposium on Logic-Based Program Synthesis and Transformation, LOPSTR 09*, LNCS 6037: 37–51, Springer-Verlag, 2009.

[SGST06] P. Schneider-Kamp, J. Giesl, A. Serebrenik, and R. Thiemann. Automated termination analysis for logic programs by term rewriting. *Proc. of the 16th International Symposium on Logic-Based Program Synthesis and Transformation, LOPSTR 06*, LNCS 4407: 177–193, Springer-Verlag, 2006.

[Sma04] J.-G. Smaus. Termination of logic programs using various dynamic selection rules. *Proc. of the 20th International Conference on Logic Programming, ICLP 04*, LNCS 3132: 43–57, Springer-Verlag, 2004.

[TeR03] TeReSe. *Term Rewriting Systems*. Cambridge University Press, 2003.

[Thi07] R. Thiemann. The DP Framework for Proving Termination of Term Rewriting. PhD Thesis. Available as Technical Report AIB-2007-17, RWTH Aachen University, Germany, 2007.

[Tiw04] A. Tiwari. Termination of linear programs. *Proc. of the 16th International Conference on Computer Aided Verification, CAV 04*, LNAI, 3097:70–82, Springer-Verlag, 2004.

[Xi02] H. Xi. Dependent types for program termination verification. *Higher-Order and Symbolic Computation*, 15(1):91–131, 2002.

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