

UNIVERSIDAD POLITÉCNICA DE VALENCIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

MÁSTER EN INGENIERÍA DEL SOFTWARE, MÉTODOS FORMALES Y SISTEMAS
DE INFORMACIÓN

TESIS DE MÁSTER

Context-Sensitive Dependency Pairs Framework

CANDIDATO:
Raúl Gutiérrez

DIRECTOR:
Salvador Lucas

Curso Académico 2007/2008

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera, s/n
46022 Valencia
España

Contents

1. Introduction	1
1.1. Termination of context-sensitive rewriting	3
1.2. Dependency pairs for context-sensitive rewriting	4
1.3. Plan of the Thesis	6
2. Preliminaries	9
2.1. Abstract Reduction Systems	9
2.2. Signatures, Terms, and Positions	9
2.3. Substitutions	10
2.4. Renamings and unifiers	10
2.5. Rewrite Systems and Term Rewriting	11
2.6. Narrowing	11
2.7. Context-Sensitive Rewriting	12
3. Structure of Infinite μ-Rewrite Sequences	15
3.1. Minimal non- μ -terminating terms	15
3.1.1. Minimal non- μ -terminating terms	16
3.2. Hidden terms and hiding contexts in minimal μ -rewrite sequences . .	19
3.3. Infinite μ -rewrite sequences starting from minimal terms	24
3.3.1. Infinite μ -rewrite sequences starting from strongly minimal terms	25
4. Context-Sensitive Dependency Pairs	29
4.1. Chains of CSDPs	32
4.2. Characterizing termination of CSR using chains of CSDPs	33
5. Mechanizing proofs of μ-termination using CSDPs	37
5.1. Mechanizing termination proofs with the dependency pair framework	38
5.2. CS-termination problems and processors	39

6. CS-Termination Processors	41
6.1. Context-Sensitive Dependency Graph	41
6.1.1. Definition of the context-sensitive dependency graph	42
6.1.2. Estimating the CS-dependency graph	43
6.2. Treating collapsing pairs	47
6.3. Use of μ -reduction pairs	50
6.3.1. Argument filterings for CSR	51
6.3.2. Removing pairs using μ -reduction orderings	52
6.3.3. Usable Rules for CSR	57
6.4. Subterm criterion	78
7. Experiments	83
7.1. CSDPs vs. other techniques for proving termination of CSR	84
7.2. Contribution of the different CS-processors	84
7.3. CSDPs at the 2007 International Termination Competition	87
8. Conclusions	89
Bibliography	91

Abstract

The idea of an incremental application of different termination techniques as *processors* for solving termination problems has shown to be a powerful and efficient way to prove termination of rewriting. Nowadays, the *dependency pair framework* (which develop this idea) is the most successful approach for proving termination of rewriting. The DP-framework relies on the notion of dependency pair to decompose a termination problem into a set of *dependency pair termination problems* (DP-termination problems). These problems can be treated independently by applying different *dependency pair processors* (DP-processors). If we prove (disprove) the termination of all (some) of the DP-problems, we can ensure that the system is terminating (nonterminating).

Context-sensitive rewriting (CSR [Luc98, Luc02]) is a restriction of rewriting that forbids reductions on some subexpressions and that has been proved useful to model and analyze programming language features at different levels. Advances in dependency pair techniques related with *context-sensitive rewriting* (CSR) are growing fast in the last years. The definition of a *context-sensitive dependency pair* (CSDP) framework for CSR becomes necessary to benefit from these new techniques and approaches in the realm of CSR termination techniques.

In this thesis, we show how to develop a dependency pair framework for proving termination of CSR.

1

Introduction

Term rewriting is a branch of theoretical computer science which combines elements of logic, universal algebra, automated theorem proving and functional programming. Its foundation is equational logic. The difference between the term rewriting and the equational logic is that equations are used only in one way, i.e. the left-hand side can be replaced by the right-hand side, but not vice versa. This constitutes a Turing-complete computational model which is very close to functional programming [BN98].

Term rewriting techniques are applicable in various fields of computer science: in software engineering (e.g., equationally specified abstract data types), in programming languages (e.g., functional-logic programming), in computer algebra (e.g., symbolic computations, Gröbner bases), in program verification (e.g., automatically proving termination of programs), in automated theorem proving (e.g., equational unification), in algebra (e.g., Boolean algebra, group theory and ring theory) and in recursion theory (what is and is not computable with certain sets of rewrite rules). In other words, term rewriting has applications in practical computer science, theoretical computer science, and mathematics. Roughly speaking, term rewriting techniques can successfully be applied in areas that demand efficient methods for reasoning with equations [Ohl02].

One of the major problems one encounters in the theory of term rewriting is the characterization of classes of rewrite systems that have a desirable property like *confluence* or *termination*. A *terminating* rewrite system does not permit infinite computations, that is, every computation starting from a term must end in a normal form [BN98]. In many cases, the termination behavior depends on the *rewriting strategy*. A *rewriting strategy* (roughly speaking, a rule for appropriately choosing rewriting steps to be issued in a computation) is a restriction of the rewriting relation [TeR03]. 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. For this reason, the designers of programming languages have 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 [NSEP91], Haskell [HPJW92], 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, Luc01, LM08]. 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, \dots, f_n, \dots together with an *arity* function $\text{ar} : \mathcal{F} \rightarrow \mathbb{N}$ which establishes the number of ‘arguments’ associated to each symbol. A *replacement map* is a mapping $\mu : \mathcal{F} \rightarrow \wp(\mathbb{N})$ satisfying $\mu(f) \subseteq \{1, \dots, k\}$, 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 $f(t_1, \dots, t_k)$ if $i \in \mu(f)$.

Example 1

The following nonterminating TRS \mathcal{R} can be used to compute the list of prime numbers by using the well-known Erathostenes sieve¹ [GM99]:

$$\begin{aligned} \text{primes} &\rightarrow \text{sieve}(\text{from}(\text{s}(\text{s}(0)))) \\ \text{from}(x) &\rightarrow \text{cons}(x, \text{from}(\text{s}(x))) \\ \text{head}(\text{cons}(x, y)) &\rightarrow x \\ \text{if}(\text{true}, x, y) &\rightarrow x \end{aligned}$$

¹Without appropriate rules for defining symbol `div`, the TRS has no complete computational meaning. However, we took it here as given in [GM99] for the purpose of comparing different techniques for proving termination of CSR by transformation.

$$\begin{aligned}
\text{if}(\text{false}, x, y) &\rightarrow y \\
\text{tail}(\text{cons}(x, y)) &\rightarrow y \\
\text{sieve}(\text{cons}(x, y)) &\rightarrow \text{cons}(x, \text{filt}(x, \text{sieve}(y))) \\
\text{filt}(\text{s}(\text{s}(x)), \text{cons}(y, z)) &\rightarrow \text{if}(\text{div}(\text{s}(\text{s}(x)), y), \text{filt}(\text{s}(\text{s}(x)), z), \text{cons}(y, \text{filt}(\text{s}(\text{s}(x)), z)))
\end{aligned}$$

Consider the replacement map μ for the signature \mathcal{F} given by:

$$\mu(\text{cons}) = \mu(\text{if}) = \{1\} \text{ and } \mu(f) = \{1, \dots, \text{ar}(f)\} \text{ for all } f \in \mathcal{F} - \{\text{cons}, \text{if}\}.$$

This replacement map exemplifies two of the most typical applications of context-sensitive rewriting as a computational mechanism:

1. The declaration $\mu(\text{if}) = \{1\}$ allows us to forbid reductions on the two *alternatives* s and t of *if-then-else* expressions $\text{if}(b, s, t)$ whereas it is still possible to perform reductions on the *boolean* part b , as required to implement the usual semantics of the operator.
2. The declaration $\mu(\text{cons}) = \{1\}$ 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 tail to continue the evaluation when needed.

1.1. Termination of context-sensitive rewriting

Termination is one of the most interesting practical problems in computation and software engineering. Ensuring termination is often a prerequisite for essential program properties like soundness. Messages reporting (a neverending) “processing”, “waiting for an answer”, or even “abnormal termination” (which are often raised during the execution of software applications) usually correspond to nonterminating computations arising from bugs in the program. Thus, being able to *automatically prove* termination of programs is a key issue in modern software development.

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. For instance, as we prove below, all context-sensitive computations for the TRS \mathcal{R} in Example 1 are terminating when the replacement map μ in the example is considered.

Recently, proving termination of CSR has been recognized as an interesting problem with several applications in the fields of term rewriting and programming languages [DLM⁺04, DLM⁺08, GM04, Luc02, Luc06]. Several methods have been developed for proving termination of CSR under a replacement map μ for a given TRS \mathcal{R} (i.e., for proving the μ -termination of \mathcal{R}). A number of transformations which permit to treat termination of CSR as a standard termination problem have been described (see [GM04, Luc06] for recent surveys). Polynomial orderings and the context-sensitive version of the recursive path ordering have also been investigated [BLR02, GL02, Luc04b, Luc05].

1.2. Dependency pairs for context-sensitive rewriting

The *dependency pairs method* [AG00, GAO02, GTSK04, GTSKF06, HM04, HM05] is one of the most powerful techniques for proving termination of rewriting. Roughly speaking, given a TRS \mathcal{R} , the dependency pairs associated to \mathcal{R} conform a new TRS $DP(\mathcal{R})$ which (together with \mathcal{R}) determines the so-called *dependency chains* whose finiteness or infiniteness characterize termination or nontermination of \mathcal{R} .

Given a rewrite rule $l \rightarrow r$, we get dependency pairs $l^\sharp \rightarrow s^\sharp$ for all subterms s of r which are rooted by a defined symbol²; the notation t^\sharp for a given term t means that the root symbol f of t is *marked* thus becoming f^\sharp (often just capitalized: F , as done in our examples). Intuitively, a dependency pair captures a transition between function calls in our system. If we ensure that all possible function call paths are finite, then our system is terminating; otherwise, the system is nonterminating.

Example 2

Consider the TRS \mathcal{R} in Example 1. According to [AG00], the set $DP(\mathcal{R})$ of dependency pairs in \mathcal{R} consists of the following pairs:

$$\text{PRIMES} \rightarrow \text{SIEVE}(\text{from}(\text{s}(\text{s}(0)))) \quad (1.1)$$

$$\text{PRIMES} \rightarrow \text{FROM}(\text{s}(\text{s}(0))) \quad (1.2)$$

$$\text{FROM}(x) \rightarrow \text{FROM}(\text{s}(x)) \quad (1.3)$$

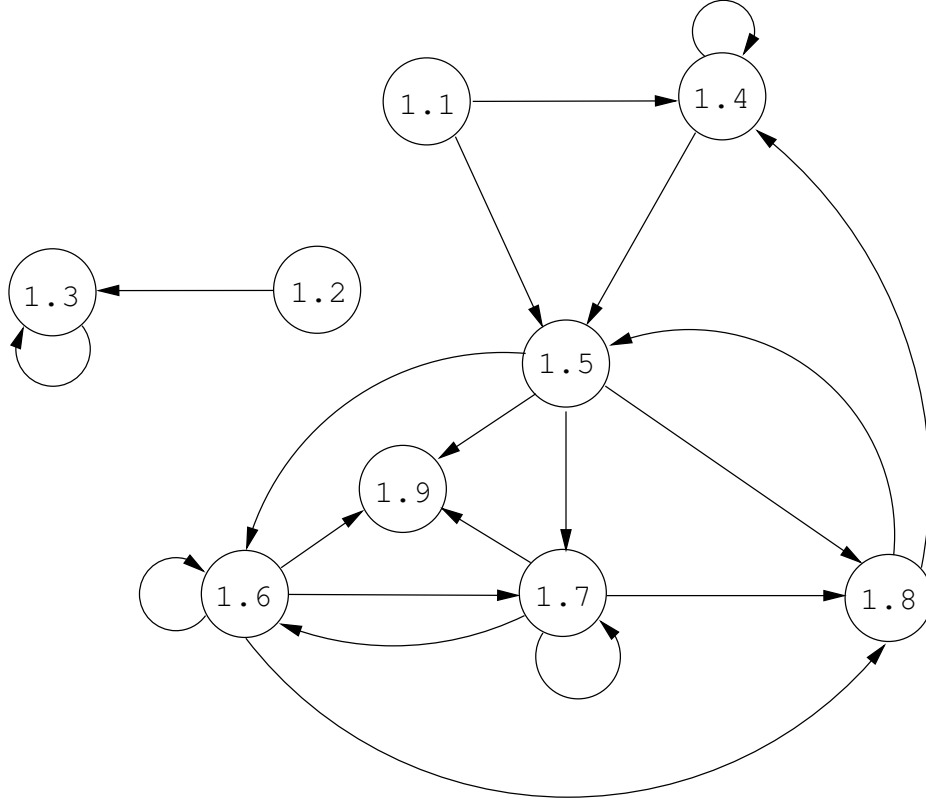
$$\text{SIEVE}(\text{cons}(x, y)) \rightarrow \text{SIEVE}(y) \quad (1.4)$$

$$\text{SIEVE}(\text{cons}(x, y)) \rightarrow \text{FILT}(x, \text{sieve}(y)) \quad (1.5)$$

$$\text{FILT}(\text{s}(\text{s}(x)), \text{cons}(y, z)) \rightarrow \text{FILT}(\text{s}(\text{s}(x)), z) \quad (1.6)$$

$$\text{FILT}(\text{s}(\text{s}(x)), \text{cons}(y, z)) \rightarrow \text{FILT}(x, \text{sieve}(y)) \quad (1.7)$$

²A symbol f is said to be *defined* in a TRS \mathcal{R} if \mathcal{R} contains a rule $f(l_1, \dots, l_k) \rightarrow r$.

Figure 1.1: Dependency Graph for the TRS \mathcal{R} in Example 1

$$\text{FILT}(s(s(x)), \text{cons}(y, z)) \rightarrow \text{SIEVE}(y) \quad (1.8)$$

$$\text{FILT}(s(s(x)), \text{cons}(y, z)) \rightarrow \text{IF}(\text{div}(s(s(x)), y), \text{filt}(s(s(x)), z), \text{cons}(y, \text{filt}(s(s(x)), z))) \quad (1.9)$$

A chain of dependency pairs is a sequence $u_i \rightarrow v_i$ of dependency pairs together with a substitution σ such that $\sigma(v_i)$ rewrites to $\sigma(u_{i+1})$ for all $i \geq 1$. The dependency pairs can be presented as a *dependency graph*, where the absence of infinite chains can be analyzed by considering the *cycles* \mathfrak{C} in the graph. For instance, the dependency graph which corresponds to the TRS \mathcal{R} in Example 1 is depicted in Figure 1.1. The cycle consisting of the node (1.3) together with the arc going from this node to itself witnesses the *nontermination* of \mathcal{R} (viewed as an ordinary rewrite system, without any restriction on its rewriting relation).

In general, these intuitions are valid for CSR: the subterms s of the right-hand sides r of the rules $l \rightarrow r$ which are considered to build the *context-sensitive dependency pairs* $l^\sharp \rightarrow s^\sharp$ must be μ -replacing terms now.

However, this is not sufficient to obtain a correct approach. The following example shows the need of a new kind of dependency pairs.

Example 3

Consider the following TRS \mathcal{R} :

$$a \rightarrow c(\underline{f(a)}) \qquad f(\underline{c(x)}) \rightarrow x \qquad (1.10)$$

together with $\mu(c) = \emptyset$ and $\mu(f) = \{1\}$ where subterms at frozen positions have been underlined in the left- and right-hand sides of the rewrite rule. No μ -replacing subterm s in the right-hand sides of the rules is rooted by a defined symbol. Thus, there is no ‘regular’ dependency pair. If no other dependency pair is considered, we could wrongly conclude that \mathcal{R} is μ -terminating. Roughly speaking: without dependency pairs no chain of function calls is possible and the system is trivially μ -terminating, which is not true (redexes to be reduced are underlined):

$$f(\underline{a}) \hookrightarrow_{\mu} \underline{f(c(f(a)))} \hookrightarrow_{\mu} f(\underline{a}) \hookrightarrow_{\mu} \dots$$

Indeed, we must add the following *collapsing* dependency pair:

$$F(\underline{c(x)}) \rightarrow x$$

which would not be allowed in Arts and Giesl’s approach [AG00] because the right-hand side is a variable.

1.3. Plan of the Thesis

After some preliminaries in Chapter 2, we develop the material in the thesis in three main parts:

1. We investigate the structure of infinite context-sensitive rewrite sequences. This analysis is essential to provide an appropriate definition of context-sensitive dependency pair, and the related notions of chains, graph, etc. Section 3.1 provides appropriate notions of *minimal* non- μ -terminating terms and introduces the main properties of such terms. Section 3.2 introduces the notion of *hidden term* and *hiding context* in a CS-TRS. This notion turns to be essential for the appropriate treatment of collapsing dependency pairs. Section 3.3 investigates the structure of infinite context-sensitive rewrite sequences starting from minimal non- μ -terminating terms.

2. We define the notions of *context-sensitive dependency pair* and *context-sensitive chain of pairs* and show how to use them to *characterize* termination of CSR. Chapter 4 introduce the general framework to compute and use context-sensitive dependency pairs for proving termination of CSR. The introduction of *collapsing* dependency pairs leads to a notion of context-sensitive dependency *chain*, which is quite differente from the standard one. In Section 4.2 we prove that our *context-sensitive dependency pairs approach* fully characterize termination of CSR.
3. We describe a suitable *framework* for dealing with proofs of termination of CSR by using the previous results. Chapter 5 provides an adaptation of the *dependency pair framework* [GTSK04, GTSKF06] to CSR by defining appropriate notions of *CS-termination problem* and *CS-processor* which rely in the notions and results investigated in the second part of the thesis. Chapter 6 introduces some CS-processors for removing or transforming collapsing pairs from CS-termination problems as the use of *μ -reduction pair* ordering to achieve proofs of termination of CSR.

We end with an experimental evaluation of our techniques in Chapter 7 and Chapter 8 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].

2.1. Abstract Reduction Systems

Let A be a set and $R \subseteq A \times A$ be a binary relation on A . An *abstract reduction system* is a pair (A, R) . In the literature [BN98, Ohl02, TeR03], the binary relation, called *reduction*, is usually represented by an arrow (\rightarrow) . If $a, b \in A$, we write $a R b$, *a reduces to b in one step*, instead of $(a, b) \in R$. A *reduction sequence* with respect to R is a finite or infinite sequence $a_0 R a_1 R a_2 R a_3 R \dots$. 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 reduction sequence $a_1 R a_2 R a_3 \dots$. A reflexive and transitive relation R is a quasi-ordering.

2.2. Signatures, Terms, and Positions

Throughout the thesis, \mathcal{X} denotes a countable set of variables and \mathcal{F} denotes a signature, i.e., a set of function symbols $\{\mathbf{f}, \mathbf{g}, \mathbf{from}, \mathbf{primes}, \mathbf{sel} \dots\}$, each having a fixed arity given by a mapping $ar : \mathcal{F} \rightarrow \mathbb{N}$. The set of terms built from \mathcal{F} and \mathcal{X} is $\mathcal{T}(\mathcal{F}, \mathcal{X})$. A term is *ground* (\mathbf{primes} , for example) 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, \dots are represented by chains of positive natural numbers used to address subterms of t . The empty chain id denoted by Λ . Given positions p, q , their concatenation is denoted 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 $\mathcal{Pos}(t)$. 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 .

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

2.3. Substitutions

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

Remark 4 In this thesis, it is *not* impose that the domain of the substitutions is finite. This is usual practice in the dependency pairs 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 $\text{Dom}(\sigma) \cap \text{Dom}(\sigma') = \emptyset$, for substitutions σ, σ' , $\sigma \cup \sigma'$ is denoted, a substitution such that $(\sigma \cup \sigma')(x) = \sigma(x)$ if $x \in \text{Dom}(\sigma)$ and $(\sigma \cup \sigma')(x) = \sigma'(x)$ if $x \in \text{Dom}(\sigma')$.

2.4. 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, it is assumed that $\text{Var}(\rho)$ 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 $t \leq t' \Leftrightarrow \exists \sigma. t' = \sigma(t)$. The fact that $\sigma(x) \leq \sigma'(x)$ is denoted as $\sigma \leq \sigma'$ 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 ; it is also said 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 $R \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X}) \times \mathcal{T}(\mathcal{F}, \mathcal{X})$ on terms is *stable* if for all terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, and substitutions σ , we have $\sigma(s) R \sigma(t)$ whenever $s R t$.

2.5. Rewrite Systems and Term Rewriting

A rewrite rule is an ordered pair (l, r) , written $l \rightarrow r$, with $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $l \notin \mathcal{X}$ and $\text{Var}(r) \subseteq \text{Var}(l)$. The left-hand side (*lhs*) of the rule is l and r is the right-hand side (*rhs*). A rewrite rule $l \rightarrow r$ 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 $\mathcal{D} = \{\text{root}(l) \mid l \rightarrow r \in R\}$ and $\mathcal{C} = \mathcal{F} \setminus \mathcal{D}$.

Example 5

Consider again the TRS in Example 1. The symbols `primes`, `sieve`, `from`, `head`, `tail`, `if` and `filt` are defined, and `s`, `0`, `cons`, `true`, `false` and `div` are constructors.

For simplicity, we often write $l \rightarrow r \in \mathcal{R}$ instead of $l \rightarrow r \in R$ to express that the rule $l \rightarrow r$ is a rule of \mathcal{R} .

A term $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ rewrites to s (at position p), written $t \xrightarrow{p}_{\mathcal{R}} s$ (or just $t \rightarrow s$, or $t \rightarrow_{\mathcal{R}} s$), if $t|_p = \sigma(l)$ and $s = t[\sigma(r)]_p$, for some rule $l \rightarrow r \in R$, $p \in \text{Pos}(t)$ and substitution σ . We write $t \xrightarrow{\geq p}_{\mathcal{R}} s$ if $t \xrightarrow{q}_{\mathcal{R}} s$ for some $q > p$. \rightarrow is written instead of $\rightarrow_{\mathcal{R}}$ if \mathcal{R} is clear for the context. A TRS \mathcal{R} is *terminating* if its one step rewrite relation $\rightarrow_{\mathcal{R}}$ is terminating.

2.6. Narrowing

Narrowing combines rewriting with unification. Narrowing a term means to apply to it the minimum substitution such that the resulting term is not in normal form and then reducing it one step [Hul80].

Given a TRS $\mathcal{R} = (\mathcal{F}, R)$. A term t *narrows* to a term s (written $t \rightsquigarrow_{\mathcal{R}, \theta} s$), if there exists a nonvariable position $p \in \text{Pos}(t)$, θ is the most general unifier of $t|_p$ and l for some rewrite rule $l \rightarrow r \in R$ (sharing no variables with t), and $s = \theta(t[r]_p)$.

2.7. Context-Sensitive Rewriting

A mapping $\mu : \mathcal{F} \rightarrow \wp(\mathbb{N})$ is a *replacement map* (or \mathcal{F} -map) if $\forall f \in \mathcal{F}, \mu(f) \subseteq \{1, \dots, \text{ar}(f)\}$ [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 μ_{\top} be the replacement map given by $\mu_{\top}(f) = \{1, \dots, \text{ar}(f)\}$ for all $f \in \mathcal{F}$ (i.e., no replacement restrictions are specified).

A binary relation R on terms is μ -monotonic if whenever $t R s$ means that $f(t_1, \dots, t_{i-1}, t, \dots, t_k) R f(t_1, \dots, t_{i-1}, s, \dots, t_k)$ for all $f \in \mathcal{F}, i \in \mu(f)$, and $t, s, t_1, \dots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If R is μ_{\top} -monotonic, then 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 $\mathcal{P}os^{\mu}(t) = \{\Lambda\} \cup \bigcup_{i \in \mu(\text{root}(t))} i.\mathcal{P}os^{\mu}(t|_i)$, 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 6 [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 \triangleright_{μ} is given by $t \triangleright_{\mu} s$ if there is $p \in \mathcal{P}os^{\mu}(t)$ such that $s = t|_p$. $t \triangleright_{\mu} s$ is written if $t \triangleright_{\mu} s$ and $t \neq s$. $t \triangleright_{\mu} s$ is written to denote that s is a non- μ -replacing (hence strict) subterm of t : $t \triangleright_{\mu} s$ if there is $p \in \mathcal{P}os(t) \setminus \mathcal{P}os^{\mu}(t)$ 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 $\mathcal{V}ar^{\mu}(t) = \{x \in \mathcal{V}ar(t) \mid t \triangleright_{\mu} x\}$. The set of non- μ -replacing variables of t , i.e., variables occurring at some non- μ -replacing position in t , is $\mathcal{V}ar^{\#}(t) = \{x \in \mathcal{V}ar(t) \mid t \not\triangleright_{\mu} x\}$. Note that $\mathcal{V}ar^{\mu}(t)$ and $\mathcal{V}ar^{\#}(t)$ do not need to be disjoint.

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

Example 7

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

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

Since in the given system the second argument of `cons` is not μ -replacing, then $2 \notin \mathcal{P}os^{\mu}(\text{cons}(0, \text{from}(s(0))))$, and the redex `from(s(0))` cannot be μ -rewritten.

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

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

3

Structure of Infinite μ -Rewrite Sequences

In term rewriting, every term in an infinite rewrite sequence has a subterm which is a *minimal nonterminating term* [HM04, HM07]. By minimal nonterminating we mean that all its proper subterms are terminating. Hence, this term can be thought as an appropriate representative of the infinite computation (the infinite sequence). If we are able to capture these terms, then studying the terminating behavior of a system becomes easier. In this chapter, we study in depth the structure of minimal nonterminating terms and infinite μ -rewrite sequences starting from these terms.

3.1. Minimal non- μ -terminating terms

Given a TRS $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$, the *minimal nonterminating terms* associated to \mathcal{R} are nonterminating terms t whose proper subterms u (i.e., $t \triangleright u$) are terminating; \mathcal{T}_∞ is the set of minimal nonterminating terms associated to \mathcal{R} . Minimal nonterminating terms have two important properties:

1. Every nonterminating term s contains a minimal nonterminating term $t \in \mathcal{T}_\infty$ (i.e., $s \triangleright t$), and
2. minimal nonterminating terms t are always rooted by a *defined* symbol $f \in \mathcal{D}$:
 $\forall t \in \mathcal{T}_\infty, \text{root}(t) \in \mathcal{D}$.

Considering the structure of the infinite rewrite sequences starting from a minimal nonterminating term $t = f(t_1, \dots, t_k) \in \mathcal{T}_\infty$ is helpful to come to the notion of dependency pair. Such sequences proceed as follows (see, e.g., [HM04]):

1. a finite number of reductions can be performed *below* the root of t , thus rewriting t into t' ; then

2. a rule $f(l_1, \dots, l_k) \rightarrow r$ applies *at the root* of t' (i.e., $t' = \sigma(f(l_1, \dots, l_k))$) for some substitution σ); and
3. there is a minimal nonterminating term $u \in \mathcal{T}_\infty$ (hence $\text{root}(u) \in \mathcal{D}$) at some position p of $\sigma(r)$ satisfying that $p \in \mathcal{Pos}_{\mathcal{F}}(r)$, (i.e., p is a *nonvariable position* of r) which ‘continues’ the infinite sequence initiated by t in a similar way.

This means that considering the occurrences of defined symbols in the right-hand sides of the rewrite rules suffices to ‘catch’ *every possible infinite rewrite sequence starting from $\sigma(r)$* . In particular, no infinite sequence can be issued *below* the variables of r (more precisely: all bindings $\sigma(x)$ are *terminating* terms). This is summarized as follows:

Proposition 8 [HM04, Lemma 1] *Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS. For all $t \in \mathcal{T}_\infty$, there exist $l \rightarrow r \in R$, a substitution σ and a term $u \in \mathcal{T}_\infty$ such that $\text{root}(u) \in \mathcal{D}$, $t \xrightarrow{>\Lambda^*} \sigma(l) \xrightarrow{\Lambda} \sigma(r) \triangleright u$ and there is a nonvariable subterm v of r , $r \triangleright v$, such that $u = \sigma(v)$.*

The standard definition of dependency pair relies on (2) and (3) above: after marking $t = f(t_1, \dots, t_k)$ as $t^\sharp = f^\sharp(t_1, \dots, t_k)$, only reductions below the root of t are possible; then, such rewritings transform t^\sharp into $\sigma(f^\sharp(l_1, \dots, l_k))$ for some substitution σ and rule $f(l_1, \dots, l_k) \rightarrow r$ of the TRS. The set of dependency pairs $f^\sharp(l_1, \dots, l_k) \rightarrow v_i^\sharp$ for $1 \leq i \leq n$ associated to such a rule represent all possible ways to continue the infinite sequence initiated by t with a minimal nonterminating term $\sigma(v_i)$.

3.1.1. Minimal non- μ -terminating terms

Before starting our discussion about (minimal) non- μ -terminating terms, we provide an obvious auxiliary result about μ -terminating terms.

Lemma 9 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, and $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If t is μ -terminating, then:*

1. *If $t \triangleright_\mu s$, then s is μ -terminating.*
2. *If $t \hookrightarrow_{\mathcal{R}, \mu}^* s$, then s is μ -terminating.*

Proof. In both cases, we proceed by contradiction.

1. If s is not μ -terminating, then there is an infinite μ -rewrite sequence starting from s and, since $t \succeq_{\mu} s$ there exists an infinite μ -rewrite sequence starting from t . This leads to a contradiction.
2. If s is not μ -terminating, then there is an infinite μ -rewrite sequence starting from s . Since $t \hookrightarrow_{\mathcal{R},\mu}^* s$, there exists an infinite μ -rewrite sequence starting from t , leading to a contradiction.

■

Given a TRS $\mathcal{R} = (\mathcal{F}, R)$ and a replacement map $\mu \in M_{\mathcal{F}}$, maybe the most straightforward definition of minimal non- μ -terminating terms is the following: let $\mathcal{T}_{\infty,\mu}$ be a set of minimal non- μ -terminating terms in the following sense: t belongs to $\mathcal{T}_{\infty,\mu}$ if t is non- μ -terminating and every strict subterm u (i.e., $t \triangleright u$) is μ -terminating. It is obvious that $\text{root}(t) \in \mathcal{D}$ for all $t \in \mathcal{T}_{\infty,\mu}$. We also have:

Lemma 10 ([AGL08]) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, and $s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If s is not μ -terminating, then there is a subterm t of s ($s \triangleright t$) such that $t \in \mathcal{T}_{\infty,\mu}$.*

Then, if there is an infinite μ -rewrite sequence $t_1 \hookrightarrow_{\mathcal{R},\mu} t_2 \hookrightarrow_{\mathcal{R},\mu} t_3 \hookrightarrow_{\mathcal{R},\mu} \dots$ we can extract a sequence of the form $t_1 \triangleright t'_1 \hookrightarrow_{\mathcal{R},\mu} s'_2 \triangleright t'_2 \hookrightarrow_{\mathcal{R},\mu} s'_3 \triangleright t'_3 \hookrightarrow_{\mathcal{R},\mu} \dots$ where all terms t'_i are minimal non- μ -terminating, i.e., $t'_i \in \mathcal{T}_{\infty,\mu}$.

Example 11

Consider the TRS \mathcal{R} in Example 3. If we have the infinite μ -rewrite sequence (redexes are underlined):

$$f(\underline{\mathbf{a}}) \hookrightarrow_{\mu} f(\underline{c(f(\mathbf{a}))}) \hookrightarrow_{\mu} f(\underline{\mathbf{a}}) \hookrightarrow_{\mu} \dots$$

we can extract the following sequence:

$$f(\underline{\mathbf{a}}) \hookrightarrow_{\mu} f(\underline{c(f(\mathbf{a}))}) \triangleright f(\underline{\mathbf{a}}) \hookrightarrow_{\mu} f(\underline{c(f(\mathbf{a}))}) \triangleright f(\underline{\mathbf{a}}) \hookrightarrow_{\mu} \dots$$

where $f(\mathbf{a}) \in \mathcal{T}_{\infty,\mu}$. Unfortunately, to obtain this sequence we have to pick subterms from some frozen positions.

This approximation to the context-sensitive case doesn't seem an appropriate generalization to the rewriting case. We just show another example.

Example 12

Consider the following TRS \mathcal{R} :

$$\begin{array}{ll} a \rightarrow c(\underline{f(a)}, b) & f(c(x), y) \rightarrow x \\ b \rightarrow b \end{array}$$

together with $\mu(c) = \mu(b) = \emptyset$ and $\mu(f) = \{1\}$. This system is non- μ -terminating and an infinite μ -rewriting is (redexes are underlined):

$$f(\underline{a}, a) \hookrightarrow_{\mu} \underline{f(c(f(a), b), a)} \hookrightarrow_{\mu} f(\underline{a}, b) \hookrightarrow_{\mu} \dots$$

We can extract the following infinite minimal sequence:

$$f(\underline{a}, a) \hookrightarrow_{\mu} \underline{f(c(f(a), b), a)} \triangleright \underline{b} \hookrightarrow_{\mu} \underline{b} \triangleright \underline{b} \hookrightarrow_{\mu} \dots$$

where \underline{b} is a minimal non- μ -terminating term that has no incidence in the infinite μ -rewrite sequence.

Therefore, this kind of minimal non- μ -terminating terms are not the most natural ones because they could occur at non- μ -replacing positions, where no μ -rewriting step is possible. So, this simple notion would not lead to an appropriate generalization of Proposition 8 to CSR. Still, we use them advantageously below; for this reason we pay them some attention here.

There is a suitable generalization of Proposition 8 to CSR (see Proposition 26 below) based on the following notion.

Definition 13 (Minimal non- μ -terminating term) *Let $\mathcal{M}_{\infty, \mu}$ be a set of minimal non- μ -terminating terms in the following sense: t belongs to $\mathcal{M}_{\infty, \mu}$ if t is non- μ -terminating and every strict μ -replacing subterm s of t (i.e., $t \triangleright_{\mu} s$) is μ -terminating.*

Note that $\mathcal{T}_{\infty, \mu} \subseteq \mathcal{M}_{\infty, \mu}$. Obviously, if $t \in \mathcal{M}_{\infty, \mu}$, then $\text{root}(t)$ is a defined symbol.

In the following we often say that terms in $\mathcal{T}_{\infty, \mu}$ are *strongly minimal* non- μ -terminating terms. Now we have the following.

Lemma 14 ([AGL08]) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, and $s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If s is not μ -terminating, then there is a μ -replacing subterm t of s such that $t \in \mathcal{M}_{\infty, \mu}$.*

Using the new definition, if there is an infinite μ -rewrite sequence starting from a term t_1 , we can now extract a sequence of the form $t_1 \triangleright_{\mu} t'_1 \hookrightarrow_{\mathcal{R}, \mu} s'_2 \triangleright_{\mu} t'_2 \hookrightarrow_{\mathcal{R}, \mu} s'_3 \triangleright_{\mu} t'_3 \hookrightarrow_{\mathcal{R}, \mu} \dots$ where all terms t'_i are minimal, $t'_i \in \mathcal{M}_{\infty, \mu}$.

Example 15

Now, continuing Example 12, from the non- μ -terminating infinite μ -rewrite sequence:

$$f(\underline{a}, a) \hookrightarrow_{\mu} \underline{f(c(f(a, b)), a)} \hookrightarrow_{\mu} f(\underline{a}, b) \hookrightarrow_{\mu} \dots$$

We can extract the following infinite minimal sequence:

$$\begin{aligned} f(\underline{a}, a) \hookrightarrow_{\mu} \underline{f(c(f(a, b)), a)} \triangleright_{\mu} \underline{f(c(f(a, b)), a)} \hookrightarrow_{\mu} f(a, b) \triangleright_{\mu} f(\underline{a}, b) \hookrightarrow_{\mu} \\ f(c(f(a, b)), b) \triangleright_{\mu} \underline{f(c(f(a, b)), b)} \hookrightarrow_{\mu} f(a, b) \triangleright_{\mu} f(\underline{a}, b) \hookrightarrow_{\mu} \dots \end{aligned}$$

where all subterm steps now can be only given over active positions (in this case all are at the root positions).

Since μ -terminating terms are preserved under μ -rewriting (Lemma 9), it follows that $\mathcal{M}_{\infty, \mu}$ is preserved under *inner* μ -rewritings in the following sense.

Lemma 16 ([AGL08]) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, and $t \in \mathcal{M}_{\infty, \mu}$. If $t \xrightarrow{\Lambda^*}_{\mathcal{R}, \mu} u$ and u is non- μ -terminating, then $u \in \mathcal{M}_{\infty, \mu}$.*

Lemma 16 does *not* hold for $\mathcal{T}_{\infty, \mu}$: consider the CS-TRS (\mathcal{R}, μ) in Example 3. We have that $f(a) \in \mathcal{T}_{\infty, \mu}$. Now, $f(a) \hookrightarrow_{\mu} f(c(f(a)))$ and $f(c(f(a)))$ is not μ -terminating. However, $f(c(f(a))) \notin \mathcal{T}_{\infty, \mu}$ as shown in Example 11.

3.2. Hidden terms and hiding contexts in minimal μ -rewrite sequences

To go ahead with the study of infinite minimal μ -rewrite sequences we need the definitions of *hidden term* and *hiding context*. As we show in the next section, they play an important role in infinite minimal μ -rewrite sequences associated to \mathcal{R} . Given a CS-TRS (\mathcal{R}, μ) the *hidden terms* are nonvariable terms occurring on some frozen position in the right-hand side of some rule of \mathcal{R} .

Definition 17 (Hidden symbols and terms) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. We say that $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \setminus \mathcal{X}$ is a hidden term if there is a rule $l \rightarrow r \in R$ such that $r \triangleright_{\mu} t$. Let $\mathcal{HT}(\mathcal{R}, \mu)$ (or just \mathcal{HT} , if \mathcal{R} and μ are clear for the context) be the set of all hidden terms in (\mathcal{R}, μ) . We 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}, μ) .*

In the following, we also use $\mathcal{DHT} = \{t \in \mathcal{HT} \mid \text{root}(t) \in \mathcal{D}\}$ for the set of hidden terms which are rooted by a *defined* symbol. The *hiding contexts* are formed by symbols occurring in the right hand side of a rule at frozen positions where one of its arguments in an active position is a variable or a define symbol.

Definition 18 (Hiding context [AEF⁺08]) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. We say that the function symbol f hides position i if $f(r_1, \dots, r_i, \dots, r_n) \in \mathcal{HT}(\mathcal{R}, \mu)$, $i \in \mu(f)$, and r_i contains a defined symbol or a variable at an active position (i.e., $\text{Pos}_{\mathcal{D}}^{\mu}(r_i) \cup \text{Pos}_{\mathcal{X}}^{\mu}(r_i) \neq \emptyset$). A context $C[\square]$ is hiding if $C[\square] = \square$ or $C[\square] = f(t_1, \dots, t_{i-1}, C'[\square], t_{i+1}, \dots, t_n)$, where f hides position i and $C'[\square]$ is a hiding context.*

Example 19

For \mathcal{R} and μ as in Example 1, the maximal hidden terms of the CS-TRS are enclosed in boxes:

$$\begin{aligned}
\text{primes} &\rightarrow \text{sieve}(\text{from}(\text{s}(\text{s}(0)))) \\
\text{from}(x) &\rightarrow \text{cons}(x, \boxed{\text{from}(\text{s}(x))}) \\
\text{head}(\text{cons}(x, y)) &\rightarrow x \\
\text{if}(\text{true}, x, y) &\rightarrow x \\
\text{if}(\text{false}, x, y) &\rightarrow y \\
\text{tail}(\text{cons}(x, y)) &\rightarrow y \\
\text{sieve}(\text{cons}(x, y)) &\rightarrow \text{cons}(x, \boxed{\text{filt}(x, \text{sieve}(y))}) \\
\text{filt}(\text{s}(\text{s}(x)), \text{cons}(y, z)) &\rightarrow \text{if}(\text{div}(\text{s}(\text{s}(x)), y), \text{filt}(\text{s}(\text{s}(x)), z), \boxed{\text{cons}(y, \text{filt}(\text{s}(\text{s}(x)), z))})
\end{aligned}$$

The hidden symbols are `from`, `filt`, `sieve`, `cons` and `s`. Symbol `from` hides position 1; symbol `s` hides position 1; symbol `filt` hides positions 1 and 2; symbol `sieve` hides position 1, and symbol `cons` hides position 1.

Example 20

Consider the following nonterminating TRS \mathcal{R} which computes the zip of two lists by applying the quotient of its components [Bor03]; the auxiliary functions `tail` and `head` are also available through their usual definition by means of rewrite rules.

$$\begin{aligned}
\text{from}(x) &\rightarrow \text{cons}(x, \boxed{\text{from}(\text{s}(x))}) \\
\text{sel}(0, \text{cons}(x, \underline{\text{x}s})) &\rightarrow x
\end{aligned}$$

$$\begin{aligned}
\text{sel}(\text{s}(n), \text{cons}(x, \underline{xs})) &\rightarrow \text{sel}(n, xs) \\
\text{minus}(x, 0) &\rightarrow 0 \\
\text{minus}(\text{s}(x), \text{s}(y)) &\rightarrow \text{minus}(x, y) \\
\text{quot}(0, \text{s}(y)) &\rightarrow 0 \\
\text{quot}(\text{s}(x), \text{s}(y)) &\rightarrow \text{s}(\text{quot}(\text{minus}(x, y), \text{s}(y))) \\
\text{zWquot}(xs, \text{nil}) &\rightarrow \text{nil} \\
\text{zWquot}(\text{nil}, xs) &\rightarrow \text{nil} \\
\text{zWquot}(\text{cons}(x, \underline{xs}), \text{cons}(y, \underline{ys})) &\rightarrow \text{cons}(\text{quot}(x, y), \underline{\text{zWquot}(xs, ys)}) \\
\text{head}(\text{cons}(x, \underline{xs})) &\rightarrow x \\
\text{tail}(\text{cons}(x, \underline{xs})) &\rightarrow xs
\end{aligned}$$

Consider the replacement map μ given by $\mu(\text{cons}) = \{1\}$ and $\mu(f) = \{1, \dots, \text{ar}(f)\}$ for all $f \in (\mathcal{F} \setminus \{\text{cons}\})$ where subterms at frozen positions have been underlined in the left- and right-hand sides of the rewrite rule. The maximal hidden terms of the CS-TRS are $\text{from}(\text{s}(x))$ and $\text{zWquot}(xs, ys)$. The hidden symbols are from , s and zWquot . Symbol from hides position 1, symbol s hides position 1 and symbol zWquot hides positions 1 and 2.

The following lemma says that frozen subterms t rooted by a defined symbol in the contractum $\sigma(r)$ of a redex $\sigma(l)$ which do *not* contain t , are (at least partly) ‘introduced’ by a hidden term with a hiding context in the right-hand side r of the involved rule $l \rightarrow r$.

Lemma 21 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ be such that $\text{root}(t) \in \mathcal{D}$, $C[\square]$ be a (possibly empty) context with a μ -replacing hole, and σ be a substitution. If there is a rule $l \rightarrow r \in R$ such that $\sigma(l) \not\triangleright t$ and $\sigma(r) \triangleright_{\mu} C[t]$, then there is no $x \in \text{Var}(r)$ such that $\sigma(x) \triangleright t$. Furthermore, there is a term $t' \in \mathcal{DHT}$, a hiding context $C'[\square]$ such that $r \triangleright_{\mu} C'[t']$, $C[\square] = \sigma(C')[\square]$ and $t = \sigma(t')$.*

Proof. By contradiction. If there is $x \in \text{Var}(r)$ such that $\sigma(x) \triangleright t$, then since variables in l are always below some function symbol we have $\sigma(l) \triangleright t$, leading to a contradiction.

Since there is no $x \in \text{Var}(r)$ such that $\sigma(x) \triangleright t$ but we have that $\sigma(r) \triangleright_{\mu} C[t]$, then there are nonvariable positions p, q where $p \in \text{Pos}_{\mathcal{F}}(r) \setminus \text{Pos}_{\mathcal{F}}^{\mu}(r)$ and $q \in \text{Pos}_{\mathcal{F}}^{\mu}(r|_p)$, such that $C[t] = C[t]_q = \sigma(r|_p) \triangleright_{\mu} \sigma(r|_{pq}) = t$. Then, we let $t' = r|_{pq}$

and $C'[\square] = r|_p[\square]_q$. Note that $t' \in \mathcal{DHT}$ and $C'[\square]$ is formed by hidden symbols. Furthermore, we have $C[\square]_q = \sigma(r|_p)[\square]_q = \sigma(r|_p[\square]_q) = \sigma(C')[\square]_q = \sigma(C')[\square]$ and $\sigma(r|_{pq}) = t = \sigma(t')$ as desired. To prove that $C'[\square]$ is a hiding context we proceed by induction on $C'[\square]$:

- If $C'[\square] = \square$, then $C'[\square]$ is a hiding context, by definition.
- If $C'[\square] = f(t_1, \dots, t_{i-1}, C''[\square], t_{i+1}, \dots, t_n)$ then we have $\sigma(r) \triangleright_{\mu} \sigma(C'')[\sigma(t')] = C'''[t]$ where $C'''[\square] = \sigma(C'')[\square]$. By the induction hypothesis, $C''[\square]$ is a hiding context. Since $C'[t]_q = r|_p \in \mathcal{HT}$, $i \in \mu(f)$, $q = i.q' \in \mathcal{Pos}_{\mathcal{D}}^{\mu}(r|_p)$ and $C'[\square]$ is a context with a μ -replacing hole, we have that f hides position i , and hence $C'[\square]$ is a hiding context. ■

Now, we show how the rules construct the hiding context.

Lemma 22 *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $C[\square]$ be a context with a μ -replacing hole and σ be a substitution. If there is a rule $l \rightarrow r \in R$ such that $l \triangleright_{\mu} x$, $r \triangleright_{\mu} r|_p = C'[x]$ for a position $p \in \mathcal{Pos}(r) \setminus \mathcal{Pos}^{\mu}(r)$, and $\sigma(r|_p) = C[t]$, then $C'[\square]$ is a hiding context, $C[\square] = \sigma(C')[\square]$ and $t = \sigma(x)$.*

Proof. By induction on $C'[\square]$:

- If $C'[\square] = \square$, then $C'[\square]$ is a hiding context, by definition.
- If $C'[\square] = f(t_1, \dots, t_{i-1}, C''[\square], t_{i+1}, \dots, t_n)$ then we have $\sigma(r) \triangleright_{\mu} \sigma(C'')[\sigma(x)] = C'''[t]$ where $C'''[\square] = \sigma(C'')[\square]$. By the induction hypothesis, $C''[\square]$ is a hiding context and $t = \sigma(x)$. Since $C'[x]_q = r|_p \in \mathcal{HT}$, $i \in \mu(f)$, $q = i.q' \in \mathcal{Pos}_{\mathcal{X}}^{\mu}(r|_p)$ and $C'[\square]$ is a context with a μ -replacing hole, we have that f hides position i , and hence $C'[\square]$ is a hiding context. ■

Then, if we have a minimal non- μ -terminating term t in an infinite minimal μ -rewrite sequence with a minimal non- μ -terminating term $\sigma(t')$ at a frozen position, there is a hiding context $C'[\square]$ such that $t \triangleright_{\mu} \sigma(C')[\sigma(t')]$.

Example 23

Consider the following example:

$$f(x) \rightarrow g(c(f(x)))$$

$$\begin{aligned}
\mathbf{g}(c(x)) &\rightarrow \mathbf{g}(d(x)) \\
\mathbf{g}(x) &\rightarrow x \\
\mathbf{g}(x) &\rightarrow \mathbf{h}(e(x)) \\
\mathbf{h}(x) &\rightarrow x
\end{aligned}$$

with $\mu(\mathbf{f}) = \mu(\mathbf{g}) = \mu(\mathbf{h}) = \mu(\mathbf{d}) = \emptyset$ and $\mu(\mathbf{c}) = \mu(\mathbf{e}) = \{1\}$. If we consider the sequence starting from $\mathbf{f}(x)$, we have that:

$$\mathbf{f}(x) \hookrightarrow_{\mu} \mathbf{g}(c(\mathbf{f}(x)))$$

Since \mathbf{c} appears in the right hand side of a rule in a frozen position above a defined symbol, then, \mathbf{c} hides position 1. Notice that $\mathbf{f}(x) \in \mathcal{M}_{\infty, \mu}$ and Lemma 21 fits.

If we apply the second rule, then, we notice that \mathbf{d} is not a hiding context:

$$\mathbf{g}(c(\mathbf{f}(x))) \hookrightarrow_{\mu} \mathbf{g}(d(\mathbf{f}(x)))$$

but, we don't have to care because the first position of \mathbf{d} freeze its argument and then, if we apply the third rule then:

$$\mathbf{g}(d(\mathbf{f}(x))) \hookrightarrow_{\mu} \mathbf{d}(\mathbf{f}(x))$$

and no infinite chain can happened.

If instead of applying the second rule, we apply the fourth rule, then \mathbf{e} is a hiding context and Lemma 22 fits:

$$\mathbf{g}(c(\mathbf{f}(x))) \hookrightarrow_{\mu} \mathbf{h}(e(c(\mathbf{f}(x))))$$

If we apply the fifth rule, then the minimal term $\mathbf{f}(x)$ appear in an active position and only a hiding context is above this term.

$$\mathbf{h}(e(c(\mathbf{f}(x)))) \hookrightarrow_{\mu} \mathbf{e}(c(\mathbf{f}(x)))$$

The following lemma establishes that minimal non- μ -terminating and non- μ -replacing subterms occurring in a μ -rewrite sequence involving only minimal terms directly come from the first term in the sequence or are instances of a hidden term with a hiding context.

Lemma 24 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. Let A be a μ -rewrite sequence $t_1 \hookrightarrow t_2 \hookrightarrow \dots \hookrightarrow t_n$ with $t_i \in \mathcal{M}_{\infty, \mu}$ for all i , $1 \leq i \leq n$ and $n \geq 1$. If there is a term $t \in \mathcal{M}_{\infty, \mu}$ such that $t_1 \not\triangleright_\mu t$ and $t_n \triangleright_\mu C[t]$ where $C[\square]$ is a context with a μ -replacing hole, then $C[t] = \sigma(C')[\sigma(s)]$ for some $s \in \mathcal{DHT}$, hiding context $C'[\square]$ and substitution σ .*

We use the previous results to investigate infinite sequences that combine μ -rewriting steps on minimal non- μ -terminating terms and the extraction of such subterms as μ -replacing subterms of (instances of) right-hand sides of the rules.

Proposition 25 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. Let A be a finite or infinite sequence of the form $t_1 \xrightarrow{\Lambda} s_1 \triangleright_\mu t'_2 \xrightarrow{\triangleright_{\mathcal{R}, \mu}^* \Lambda} t_2 \xrightarrow{\Lambda} s_2 \triangleright_\mu t'_3 \xrightarrow{\triangleright_{\mathcal{R}, \mu}^* \Lambda} t_3 \dots$ with $t_i, t'_i \in \mathcal{M}_{\infty, \mu}$ for all $i \geq 1$. If there is a term $t \in \mathcal{M}_{\infty, \mu}$ such that $t_1 \not\triangleright_\mu t$ and $t_i \triangleright_\mu C[t]$ for some context $C[\square]$ with a μ -replacing hole, and some $i \geq 1$, then $C[t] = \sigma(C')[\sigma(s)]$ for some $s \in \mathcal{DHT}$, some hiding context $C'[\square]$ and substitution σ .*

3.3. Infinite μ -rewrite sequences starting from minimal terms

The following proposition establishes that, given a minimal non- μ -terminating term $t \in \mathcal{M}_{\infty, \mu}$, there are only two ways for an infinite 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. The second one is by showing up ‘hidden’ non- μ -terminating subterms with a hiding context which are activated by *migrating* variables in a rule $l \rightarrow r$, i.e., variables $x \in \mathcal{Var}^\mu(r) \setminus \mathcal{Var}^\mu(l)$ which are *not* μ -replacing in the left-hand side l but become μ -replacing in the right-hand side r .

Proposition 26 ([AGL08]) *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. Then for all $t \in \mathcal{M}_{\infty, \mu}$, there exist $l \rightarrow r \in R$, a substitution σ and a term $u \in \mathcal{M}_{\infty, \mu}$ such that $t \xrightarrow{\triangleright_{\mathcal{R}, \mu}^* \Lambda} \sigma(l) \xrightarrow{\Lambda} \sigma(r) \triangleright_\mu u$ and either*

1. *there is a μ -replacing subterm s of r , $r \triangleright_\mu s$, such that $u = \sigma(s)$, or*
2. *there is $x \in \mathcal{Var}^\mu(r) \setminus \mathcal{Var}^\mu(l)$ such that $\sigma(x) = C[u]$ for a possible empty context $C[\square]$ with a μ -replacing hole.*

Proposition 26 entails the following result, which establishes some properties of infinite sequences starting from minimal non- μ -terminating terms.

Corollary 27 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. For all $t \in \mathcal{M}_{\infty, \mu}$, there is an infinite sequence*

$$t \xrightarrow{>\Lambda^*_{\mathcal{R}, \mu}} \sigma_1(l_1) \xrightarrow{\Lambda} \sigma_1(r_1) \triangleright_{\mu} t_1 \xrightarrow{>\Lambda^*_{\mathcal{R}, \mu}} \sigma_2(l_2) \xrightarrow{\Lambda} \sigma_2(r_2) \triangleright_{\mu} t_2 \xrightarrow{>\Lambda^*_{\mathcal{R}, \mu}} \dots$$

where, for all $i \geq 1$, $l_i \rightarrow r_i \in R$ are rewrite rules, σ_i are substitutions, and terms $t_i \in \mathcal{M}_{\infty, \mu}$ are minimal non- μ -terminating terms such that either

1. $t_i = \sigma_i(s_i)$ for some s_i such that $r_i \triangleright_{\mu} s_i$, or
2. $\sigma_i(x_i) = C_i[t_i]$ for some $x_i \in \mathcal{V}ar^{\mu}(r_i) \setminus \mathcal{V}ar^{\mu}(l_i)$ and some possible empty context $C_i[\square]$ with a μ -replacing hole.

Since every minimal non- μ -terminating term contains a strongly minimal non- μ -terminating subterm (see Lemma 10), in the following we investigate infinite minimal non- μ -terminating sequences starting by one of these terms. In this way, we ensure the absence of non- μ -terminating terms in frozen positions in the initial term.

3.3.1. Infinite μ -rewrite sequences starting from strongly minimal terms

In the following, we consider a function REN^{μ} which *independently* renames all occurrences of μ -replacing variables within a term t by using new fresh variables which are not in $\mathcal{V}ar(t)$:

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

Note that $\text{REN}^{\mu}(t)$ renames all μ -replacing positions of variables in t by new fresh variables y but keeps variables at non- μ -replacing positions untouched. Note that, in contrast to a *renaming* substitution (often denoted by ρ), REN^{μ} is *not* a substitution: it will replace different μ -replacing occurrences of the same variable by different variables.

Proposition 28 ([AGL08]) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \setminus \mathcal{X}$ be a nonvariable term and σ be a substitution such that $\sigma(t) \in \mathcal{M}_{\infty, \mu}$. Then, $\text{REN}^{\mu}(t)$ is μ -narrowable.*

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

$$\mathcal{NHT}(\mathcal{R}, \mu) = \{t \in \mathcal{DHT} \mid \text{NARR}^{\mu}(\text{REN}^{\mu}(t))\}$$

be the set of *hidden terms* which are rooted by a *defined* symbol, and that, after applying REN^{μ} , become μ -narrowable. As a consequence of the previous results, we have the following main result which we will use later.

Theorem 29 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. For all $t \in \mathcal{T}_{\infty, \mu}$, there is an infinite sequence*

$$t = t_0 \xrightarrow{\text{>}_{\mathcal{R}, \mu} \Lambda^*} \sigma_1(l_1) \xrightarrow{\Lambda} \sigma_1(r_1) \triangleright_{\mu} t_1 \xrightarrow{\text{>}_{\mathcal{R}, \mu} \Lambda^*} \sigma_2(l_2) \xrightarrow{\Lambda} \sigma_2(r_2) \triangleright_{\mu} t_2 \xrightarrow{\text{>}_{\mathcal{R}, \mu} \Lambda^*} \dots$$

where, for all $i \geq 1$, $l_i \rightarrow r_i \in R$ are rewrite rules, σ_i are substitutions, and terms $t_i \in \mathcal{M}_{\infty, \mu}$ are minimal non- μ -terminating terms such that either

1. $t_i = \sigma_i(s_i)$ for some s_i such that $r_i \triangleright_{\mu} s_i$, or
2. $\sigma_i(x_i) = C_i[t_i]$ for some $x_i \in \text{Var}^{\mu}(r_i) \setminus \text{Var}^{\mu}(l_i)$ and $C_i[t_i] = \theta_i(C'_i[t'_i])$ for some $t'_i \in \mathcal{NHT}$, some hiding context $C'_i[\square]$ and substitution θ_i .

Proof. Since $\mathcal{T}_{\infty, \mu} \subseteq \mathcal{M}_{\infty, \mu}$, by Corollary 27, we have a sequence

$$t = t_0 \xrightarrow{\text{>}_{\mathcal{R}, \mu} \Lambda^*} \sigma_1(l_1) \xrightarrow{\Lambda} \sigma_1(r_1) \triangleright_{\mu} t_1 \xrightarrow{\text{>}_{\mathcal{R}, \mu} \Lambda^*} \sigma_2(l_2) \xrightarrow{\Lambda} \sigma_2(r_2) \triangleright_{\mu} t_2 \xrightarrow{\text{>}_{\mathcal{R}, \mu} \Lambda^*} \dots$$

where, for all $i \geq 1$, $l_i \rightarrow r_i \in R$, σ_i are substitutions, $t_i \in \mathcal{M}_{\infty, \mu}$, and either (1) $t_i = \sigma_i(s_i)$ for some s_i such that $r_i \triangleright_{\mu} s_i$ or (2) $\sigma_i(x_i) = C_i[t_i]$ for some $x_i \in \text{Var}^{\mu}(r_i) \setminus \text{Var}^{\mu}(l_i)$ (and hence $\sigma(l_i) \triangleright_{\mu} t_i$ and $\sigma(r_i) \triangleright_{\mu} t_i$ as well) and some context $C_i[\square]$ with a μ -replacing hole. We only need to prove that for the terms $C_i[t_i]$, terms t_i are instances of hidden terms in \mathcal{NHT} and contexts $C_i[\square]$ are instances of hiding contexts whenever the second condition holds. By Proposition 25, for all such terms t_i , we have that either (A) $\sigma_1(l_1) \triangleright_{\mu} t_i$ or (B) $C[t_i] = \theta_i(C'[t'_i])$ for some $t'_i \in \mathcal{DHT}$, hiding context $C'_i[\square]$ and substitution θ_i . In the second case (B), we are done by just considering Proposition 28, which ensures that $t'_i \in \mathcal{NHT}$. In the first one (A),

since $t \xrightarrow[\mathcal{R}, \mu]{>\Lambda^*} \sigma_1(l_1)$ and $\sigma_1(l_1)$ is not μ -terminating, by Lemma 16 all terms u_j in the μ -rewrite sequence

$$t = u_1 \xrightarrow{>\Lambda} u_2 \xrightarrow{>\Lambda} \dots \xrightarrow{>\Lambda} u_m = \sigma_1(l_1)$$

for $m \geq 1$, belong to $\mathcal{M}_{\infty, \mu}$: $u_j \in \mathcal{M}_{\infty, \mu}$ for all j , $1 \leq j \leq m$. Since $t \in \mathcal{T}_{\infty, \mu}$, all its strict subterms (disregarding their μ -replacing character) are μ -terminating. Therefore, $t \not\triangleright t_i$ (because t_i is not μ -terminating) and by Lemma 24, $C[t_i] = \theta_i(C'[t'_i])$ for some $t'_i \in \mathcal{DHT}$, hiding context $C'_i[\square]$ and substitution θ_i . Again, by Proposition 28 we have $t'_i \in \mathcal{NHT}$. ■

4

Context-Sensitive Dependency Pairs

Lemma 10 and Theorem 29 are the basis for our definition of *Context-Sensitive Dependency Pairs* (and the corresponding *chains*). Together, they show that every non- μ -terminating term s has an associated infinite μ -rewrite sequence starting from a strongly minimal subterm $t \in \mathcal{T}_{\infty, \mu}$ (i.e., $s \geq t$). Such a sequence proceeds by first performing some μ -rewriting steps *below the root* of t to obtain a term t' (i.e., $t \xrightarrow{\geq}_{\mathcal{R}, \mu}^* t'$) and then applying a rule $l \rightarrow r$ at the *topmost* position of t' (i.e., $t' = \sigma(l)$ for some matching substitution σ). According to Proposition 26, the application of such a rule either

1. *introduces* a new minimal non- μ -terminating subterm u having a prefix s which is a μ -replacing subterm of r (i.e., $r \geq_{\mu} s$ and $u = \sigma(s)$). Furthermore, by Proposition 28, $\text{REN}^{\mu}(s)$ must be μ -narrowable; or else
2. *takes* a minimal non- μ -terminating and non- μ -replacing subterm u of t' (i.e., $t' \triangleright_{\mu} u$) and
 - a) brings it up to an *active* position by means of the binding $\sigma(x)$ for some *migrating variable* x in $l \rightarrow r$, $\sigma(x) = C[u]$ for some $x \in \text{Var}^{\mu}(r) \setminus \text{Var}^{\mu}(l)$ and a context $C[\square]$ with a μ -replacing hole.
 - b) At this point, we know that u , which is rooted by a defined symbol due to $u \in \mathcal{M}_{\infty, \mu}$, is an instance of a hidden term $u' \in \mathcal{NHT}$ and $C[\square]$ is an instance of a hiding context $C'[\square]$: $C[u] = \theta(C')[\theta(u')]$ for some substitution θ .
 - c) Afterwards, further *inner* μ -rewritings on u lead to match the left-hand-side l' of a new rule $l' \rightarrow r'$ and everything starts again.

This process is abstracted in the following definition of *context-sensitive dependency pairs* and in the definition of chain below.

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 $t = f(t_1, \dots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, we write t^\sharp to denote the *marked* term $f^\sharp(t_1, \dots, t_k)$. Conversely, given a marked term $t = f^\sharp(t_1, \dots, t_k)$, where $t_1, \dots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, we write t^\flat to denote the term $f(t_1, \dots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Let $\mathcal{T}^\sharp(\mathcal{F}, \mathcal{X}) = \{t^\sharp \mid t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \setminus \mathcal{X}\}$ be the set of marked terms.

Definition 30 (Context-Sensitive Dependency Pairs) *Let $\mathcal{R} = (\mathcal{F}, R) = (C \uplus \mathcal{D}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. We define $\text{DP}(\mathcal{R}, \mu) = \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu) \cup \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$ to be the set of context-sensitive dependency pairs (CSDPs) where:*

$$\begin{aligned} \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu) &= \{l^\sharp \rightarrow s^\sharp \mid l \rightarrow r \in R, r \succeq_\mu s, \text{root}(s) \in \mathcal{D}, l \not\prec_\mu s, \text{NARR}^\mu(\text{REN}^\mu(s))\} \\ \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu) &= \begin{cases} \emptyset & \text{if } \mathcal{NHT} = \emptyset \\ \{l^\sharp \rightarrow x \mid l \rightarrow r \in R, x \in \text{Var}^\mu(r) \setminus \text{Var}^\mu(l)\} & \text{if } \mathcal{NHT} \neq \emptyset \end{cases} \end{aligned}$$

We extend $\mu \in M_{\mathcal{F}}$ into $\mu^\sharp \in M_{\mathcal{F} \cup \mathcal{D}^\sharp}$ by $\mu^\sharp(f) = \mu(f)$ if $f \in \mathcal{F}$, and $\mu^\sharp(f^\sharp) = \mu(f)$ if $f \in \mathcal{D}$.

The CSDPs $u \rightarrow v \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$ in Definition 30, consisting of collapsing rules only, are called the *collapsing* CSDPs.

If we apply in an infinite minimal μ -rewrite sequence a rule $l \rightarrow r \in R$ with a migrating variable x instantiated to an infinite minimal term with a substitution σ , we know that this instantiated variable has the form $\sigma(x) = C[t]$, where t is an instance of a hidden term t' and $C[\square]$ is an instance of a hiding context $C'[\square]$. Hence, if there is no any narrowable hidden term, then it is impossible to have an infinite minimal μ -rewrite sequence where a rule with a migrating variable is applied and the migrating variable is instantiated to a minimal non- μ -terminating term.

A rule $l \rightarrow r$ of a TRS \mathcal{R} is μ -conservative if $\text{Var}^\mu(r) \subseteq \text{Var}^\mu(l)$, i.e., it does not contain migrating variables; \mathcal{R} is μ -conservative if all its rules are (see [Luc96, Luc06]). The following fact is obvious from Definition 30.

Proposition 31 ([AGL06]) *Let \mathcal{R} be a TRS and $\mu \in M_{\mathcal{R}}$. If (\mathcal{R}, μ) is μ -conservative, then $\text{DP}(\mathcal{R}, \mu) = \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$.*

Therefore, in order to deal with μ -conservative TRSs \mathcal{R} we only need to consider the ‘classical’ dependency pairs in $\text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$.

Example 32

Consider the following TRS \mathcal{R} :

$$\begin{array}{ll} g(x) \rightarrow h(x) & h(d) \rightarrow g(c) \\ c \rightarrow d & \end{array}$$

together with $\mu(g) = \mu(h) = \emptyset$ [Zan97, Example 1]. Note that \mathcal{R} is μ -conservative. $\text{DP}(\mathcal{R}, \mu)$ consists of the following (noncollapsing) CSDPs:

$$\begin{array}{ll} G(x) \rightarrow H(x) & H(d) \rightarrow G(c) \end{array}$$

with $\mu^\sharp(G) = \mu^\sharp(H) = \emptyset$.

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.

Example 33

For the CS-TRS (\mathcal{R}, μ) in Example 1, we have six CSDPs: (1.1), (1.2), and (1.9) as in Example 2 plus the following three *collapsing* CSDPs:

$$\text{TAIL}(\text{cons}(x, y)) \rightarrow y \tag{4.1}$$

$$\text{IF}(\text{true}, x, y) \rightarrow x \tag{4.2}$$

$$\text{IF}(\text{false}, x, y) \rightarrow y \tag{4.3}$$

Example 34

For the CS-TRS (\mathcal{R}, μ) in Example 20, we have six CSDPs:

$$\text{SEL}(s(n), \text{cons}(x, \underline{xs})) \rightarrow \text{SEL}(n, xs) \tag{4.4}$$

$$\text{MINUS}(s(x), s(y)) \rightarrow \text{MINUS}(x, y) \tag{4.5}$$

$$\text{QUOT}(s(x), s(y)) \rightarrow \text{QUOT}(\text{minus}(x, y), s(y)) \tag{4.6}$$

$$\text{QUOT}(s(x), s(y)) \rightarrow \text{MINUS}(x, y) \tag{4.7}$$

$$\text{ZwQUOT}(\text{cons}(x, \underline{xs}), \text{cons}(y, \underline{ys})) \rightarrow \text{QUOT}(x, y) \tag{4.8}$$

$$\text{TAIL}(\text{cons}(x, y)) \rightarrow y \tag{4.9}$$

4.1. Chains of CSDPs

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, GTSKF06]. As we prove in Section 4.2, this is also true for CSR when CSDPs are considered. First, we have to introduce a suitable notion of chain which can be used with CSDPs. As in the DP-framework [GTSK04, GTSKF06], where the precedence 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 CSDPs, by just taking $\mathcal{P} = \text{DP}(\mathcal{R}, \mu)$.

Definition 35 (Chain of pairs - Minimal chain) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. A $(\mathcal{P}, \mathcal{R}, \mu)$ -chain is a finite or infinite sequence of pairs $u_i \rightarrow v_i \in \mathcal{P}$, together with a substitution σ satisfying that, for all $i \geq 1$:*

1. *if $v_i \notin \text{Var}(u_i) \setminus \text{Var}^\mu(u_i)$, then $\sigma(v_i) \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_{i+1})$, and*
2. *if $v_i \in \text{Var}(u_i) \setminus \text{Var}^\mu(u_i)$, then there is $s_i \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and a context $C_i[\square]$ with a μ -replacing hole such that $\sigma(v_i) = C_i[s_i]$ and $s_i^\sharp \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_{i+1})$.*

As usual, we assume that different occurrences of dependency pairs do not share any variable (renaming substitutions are used if necessary).

A $(\mathcal{P}, \mathcal{R}, \mu)$ -chain is called minimal if for all $i \geq 1$,

1. *if $v_i \notin \text{Var}(u_i) \setminus \text{Var}^\mu(u_i)$, then $\sigma(v_i)$ is (\mathcal{R}, μ) -terminating, and*
2. *if $v_i \in \text{Var}(u_i) \setminus \text{Var}^\mu(u_i)$, then s_i^\sharp is (\mathcal{R}, μ) -terminating, $\exists \bar{s}_i \in \mathcal{NHT}(\mathcal{R}, \mu)$ and a hiding context $\bar{C}_i[\square]$ such that $C_i[s_i] = \sigma(\bar{C}_i[\bar{s}_i])$.*

Note that the condition $v_i \in \text{Var}(u_i) \setminus \text{Var}^\mu(u_i)$ in Definition 35 implies that v_i is a variable. Furthermore, since each $u_i \rightarrow v_i \in \mathcal{P}$ is a rewrite rule (i.e., $\text{Var}(v_i) \subseteq \text{Var}(u_i)$), v_i is a *migrating variable* in the rule $u_i \rightarrow v_i$. In the following, the pairs in a CS-TRS (\mathcal{P}, μ) , where $\mathcal{P} = (\mathcal{G}, P)$, are partitioned according to its role in Definition 35 as follows:

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

Remark 36 (Notation for chains) In general, a $(\mathcal{P}, \mathcal{R}, \mu)$ -chain can be written as follows:

$$\sigma(u_1) \hookrightarrow_{\mathcal{P}, \mu} \circ \underset{\mu}{\triangleright}^\sharp t_1 \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_2) \hookrightarrow_{\mathcal{P}, \mu} \circ \underset{\mu}{\triangleright}^\sharp t_2 \hookrightarrow_{\mathcal{R}, \mu}^* \dots$$

where, for all $i \geq 1$ and $u_i \rightarrow v_i \in \mathcal{P}$,

1. if $u_i \rightarrow v_i \notin \mathcal{P}_{\mathcal{X}}$, then $t_i = \sigma(v_i)$,
2. if $u_i \rightarrow v_i \in \mathcal{P}_{\mathcal{X}}$, then $t_i = s_i^\sharp$ for some term s_i such that $\sigma(v_i) \succeq_\mu s_i$.

The relation \succeq_μ^\sharp is defined as follows:

- $s \succeq_\mu^\sharp t$ is equivalent to $s \succeq_\mu t^\sharp$ if $s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and $t \in \mathcal{T}^\sharp(\mathcal{F}, \mathcal{X})$, and
- $s \succeq_\mu^\sharp t$ is equivalent to $s = t$ for $s, t \in \mathcal{T}^\sharp(\mathcal{F}, \mathcal{X})$.

■

In the following, we let $\mathcal{NHT}_{\mathcal{P}}(\mathcal{R}, \mu) \subseteq \mathcal{NHT}(\mathcal{R}, \mu)$ (or just $\mathcal{NHT}_{\mathcal{P}}$ if \mathcal{R} and μ are clear from the context) be as follows:

$$\mathcal{NHT}_{\mathcal{P}}(\mathcal{R}, \mu) = \{t \in \mathcal{NHT}(\mathcal{R}, \mu) \mid \exists u \rightarrow v \in \mathcal{P}, \exists \theta, \theta', \theta(t^\sharp) \xrightarrow[\mathcal{R}, \mu]{*} \theta'(u)\}$$

This set contains the narrowable hidden terms which ‘connect’ with some pair in \mathcal{P} .

The following proposition establishes some important ‘basic’ cases of (absence of) infinite context-sensitive chains of pairs which are used later.

4.2. Characterizing termination of CSR using chains of CSDPs

The following result establishes the soundness of the context-sensitive dependency pairs 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 CSDPs are available when necessary.

Theorem 37 (Soundness) *Let \mathcal{R} be a TRS and $\mu \in M_{\mathcal{R}}$. If there is no infinite minimal $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ -chain, then \mathcal{R} is μ -terminating.*

Proof. By contradiction. If \mathcal{R} is not μ -terminating, then by Lemma 10 there is $t \in \mathcal{T}_{\infty, \mu}$. By Theorem 29, there are rules $l_i \rightarrow r_i \in \mathcal{R}$, matching substitutions σ_i , and terms $t_i \in \mathcal{M}_{\infty, \mu}$, for $i \geq 1$ such that

$$t = t_0 \xrightarrow[\mathcal{R}, \mu]{> \Lambda^*} \sigma_1(l_1) \xrightarrow{\Lambda} \sigma_1(r_1) \succeq_\mu t_1 \xrightarrow[\mathcal{R}, \mu]{> \Lambda^*} \sigma_2(l_2) \xrightarrow{\Lambda} \sigma_2(r_2) \succeq_\mu t_2 \xrightarrow[\mathcal{R}, \mu]{> \Lambda^*} \dots$$

where either (D1) $t_i = \sigma_i(s_i)$ for some s_i such that $r_i \succeq_\mu s_i$ or (D2) $\sigma_i(x_i) = C_i[t_i]$ for some $x_i \in \text{Var}^\mu(r_i) \setminus \text{Var}^\mu(l_i)$, $C_i[t_i] = \theta_i(C'_i[t'_i])$ for some $t'_i \in \mathcal{NHT}$ and some

hiding context $C'_i[\square]$. Furthermore, since $t_{i-1} \xrightarrow{\Lambda^*}_{\mathcal{R},\mu} \sigma_i(l_i)$ and $t_{i-1} \in \mathcal{M}_{\infty,\mu}$ (in particular, $t_0 = t \in \mathcal{T}_{\infty,\mu} \subseteq \mathcal{M}_{\infty,\mu}$), by Lemma 16, $\sigma_i(l_i) \in \mathcal{M}_{\infty,\mu}$ for all $i \geq 1$. Note that, since $t_i \in \mathcal{M}_{\infty,\mu}$, we have that t_i^\sharp is μ -terminating (with respect to \mathcal{R}), because all μ -replacing subterms of t_i (hence of t_i^\sharp as well) are μ -terminating and $\text{root}(t_i^\sharp)$ is not a defined symbol of \mathcal{R} .

First, note that $\text{DP}(\mathcal{R}, \mu)$ 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 35. Furthermore, $\mathcal{P}_{\mathcal{G}} = \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$ and $\mathcal{P}_{\mathcal{X}} = \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$. We can define an infinite minimal $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ -chain using CSDPs $u_i \rightarrow v_i$ for $i \geq 1$, where $u_i = l_i^\sharp$ and

1. $v_i = s_i^\sharp$ if (D1) holds. Since $t_i \in \mathcal{M}_{\infty,\mu}$, we have that $\text{root}(s_i) \in \mathcal{D}$ and, because $t_i = \sigma_i(s_i)$, by Proposition 28 $\text{REN}^\mu(s_i)$ 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 $\sigma_i(s_i) = t_i \in \mathcal{M}_{\infty,\mu}$) contradicts that $\sigma_i(l_i) \in \mathcal{M}_{\infty,\mu}$. Thus, $l_i \not\triangleright_\mu s_i$. Hence, $u_i \rightarrow v_i \in \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$. Furthermore, $t_i^\sharp = \sigma_i(v_i)$ is μ -terminating. Finally, since $t_i = \sigma_i(s_i) \xrightarrow{\Lambda^*}_{\mathcal{R},\mu} \sigma_{i+1}(l_{i+1})$ and μ^\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 $\sigma_i(v_i) = \sigma_i(s_i^\sharp) \xrightarrow{*}_{\mathcal{R},\mu^\sharp} \sigma_{i+1}(u_{i+1})$.
2. $v_i = x_i$ if (D2) holds. Since $\mathcal{NHT} \neq \emptyset$ (by assumption), we have that $u_i \rightarrow v_i \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$. As discussed above, t_i^\sharp is μ -terminating. Since $\sigma_i(x_i) = C_i[t_i]$, we have that $\sigma_i(v_i) = C_i[t_i]$ for a context $C_i[\square]$ with a μ -replacing hole. Finally, since $t_i \xrightarrow{\Lambda^*}_{\mathcal{R},\mu} \sigma_{i+1}(l_{i+1})$, again we have that $u_i^\sharp \xrightarrow{*}_{\mathcal{R},\mu^\sharp} \sigma_{i+1}(u_{i+1})$. Furthermore, $C_i[t_i] = \theta_i(C'_i[t'_i])$ for some $t'_i \in \mathcal{NHT}$ and some hiding context $C'_i[\square]$.

Regarding σ , w.l.o.g. we can assume that $\text{Var}(l_i) \cap \text{Var}(l_j) = \emptyset$ for all $i \neq j$, and therefore $\text{Var}(u_i) \cap \text{Var}(u_j) = \emptyset$ as well. Then, σ is given by $\sigma(x) = \sigma_i(x)$ whenever $x \in \text{Var}(u_i)$ for $i \geq 1$. From the discussion in points (1) and (2) above, we conclude that the CSDPs $u_i \rightarrow v_i$ for $i \geq 1$ together with σ define an infinite minimal $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ -chain which contradicts our initial assumption. \blacksquare

Now we prove that the previous CS-dependency pairs approach is not only correct but also complete for proving termination of CSR.

Theorem 38 (Completeness) *Let \mathcal{R} be a TRS and $\mu \in M_{\mathcal{R}}$. If \mathcal{R} is μ -terminating, then there is no infinite $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ -chain.*

Proof. By contradiction. If there is an infinite $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ -chain, then there is a substitution σ and dependency pairs $u_i \rightarrow v_i \in \text{DP}(\mathcal{R}, \mu)$ such that

1. If $u_i \rightarrow v_i \in \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$, then $\sigma(v_i) \hookrightarrow_{\mathcal{R}, \mu^\sharp}^* \sigma(u_{i+1})$, and
2. if $u_i \rightarrow v_i = u_i \rightarrow x_i \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$, then there is $s_i \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and a context $C_i[\square]$ with a μ -replacing hole such that $\sigma(x_i) = C_i[s_i]$ and $s_i^\sharp \hookrightarrow_{\mathcal{R}, \mu^\sharp}^* \sigma(u_{i+1})$.

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

1. If $u_1 \rightarrow v_1 \in \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$, then v_1^\sharp is a μ -replacing subterm of the right-hand-side r_1 of a rule $l_1 \rightarrow r_1$ in \mathcal{R} . Therefore, $r_1 = D_1[v_1^\sharp]_{p_1}$ for some $p_1 \in \text{Pos}^\mu(r_1)$ and we can perform the μ -rewriting step $t_1 = \sigma(u_1) \hookrightarrow_{\mathcal{R}, \mu} \sigma(r_1) = \sigma(D_1)[\sigma(v_1^\sharp)]_{p_1} = s_1$, where $\sigma(v_1^\sharp)^\sharp = \sigma(v_1) \hookrightarrow_{\mathcal{R}, \mu^\sharp}^* \sigma(u_2)$ and $\sigma(u_2)$ initiates an infinite $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ -chain. Note that $p_1 \in \text{Pos}^\mu(s_1)$.
2. If $u_1 \rightarrow x \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$, then there is a rule $l_1 \rightarrow r_1$ in \mathcal{R} such that $u_1 = l_1^\sharp$, and $x \in \text{Var}^\mu(r_1) \setminus \text{Var}^\mu(l_1)$, i.e., $r_1 = D_1[x]_{q_1}$ for some $q_1 \in \text{Pos}^\mu(r_1)$. Furthermore, since there is a subterm s and a context $C_1[\square]$ with a μ -replacing hole such that $\sigma(x) = C_1[s]$ and $s^\sharp \hookrightarrow_{\mathcal{R}, \mu^\sharp}^* \sigma(u_2)$. Therefore, we can perform the μ -rewriting step $t_1 = \sigma(l_1) \hookrightarrow_{\mathcal{R}, \mu} \sigma(r_1) = \sigma(D_1)[C_1[s]] = s_1$ where $s^\sharp \hookrightarrow_{\mathcal{R}, \mu^\sharp}^* \sigma(u_2)$ (hence $s \xrightarrow{\Lambda^*}_{\mathcal{R}, \mu} u_2^\sharp$) and $\sigma(u_2)$ initiates an infinite $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ -chain.

Since $\mu^\sharp(f^\sharp) = \mu(f)$, and C_1 is a context with a μ -replacing hole, we have that $s_1 \hookrightarrow_{\mathcal{R}, \mu}^* t_2[\sigma(u_2)]_{p_1} = t_2$ and $p_1 \in \text{Pos}^\mu(t_2)$. Therefore, we can build in that way an infinite μ -rewrite sequence

$$t_1 \hookrightarrow_{\mathcal{R}, \mu} s_1 \hookrightarrow_{\mathcal{R}, \mu}^* t_2 \hookrightarrow_{\mathcal{R}, \mu} \dots$$

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

Corollary 39 (Characterization of μ -termination) *Let \mathcal{R} be a TRS and $\mu \in M_{\mathcal{R}}$. Then, we have that \mathcal{R} is μ -terminating if and only if there is no infinite minimal $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ -chain.*

5

Mechanizing proofs of μ -termination using CSDPs

During the last ten years, the dependency pairs method has evolved to a powerful technique for proving termination of TRSs in practice. From the already classical Arts and Giesl's article [AG00] to the last developments corresponding to the so-called *dependency pair framework* [GTSK04, GTSKF06, Thi07] many new improvements have been introduced.

In the DP-approach [AG00], the starting point is a TRS \mathcal{R} from which a set of dependency pairs $\text{DP}(\mathcal{R})$ is obtained. Then, such dependency pairs are organized in a dependency graph $\text{DG}(\mathcal{R})$ and the cycles of the graph are analyzed to show that no infinite chains of DPs can be obtained from them. The dependency pairs approach emphasizes (at least theoretically) a 'linear' (although somehow *modular*, see [GAO02]) procedure for proving termination. In this sense, the treatment of strongly connected components of the graph (SCCs) instead of cycles, as suggested by Hirokawa and Middeldorp [HM04, HM05], brought an important improvement in its practical use because it provides a way to make the proofs more incremental without running out of the basic DP-approach. In the DP-approach, dependency pairs are considered as components of the chains (or cycles). Since they only make sense when an underlying TRS is given as the source of the dependency pairs, transforming DPs is possible (the *narrowing* transformation is already described in [AG00]) but only as a final step because, afterwards, they are not dependency pairs of the original TRS anymore.

The dependency pair framework solves these problems in a clean way, leading to a more powerful mechanization of termination proofs.

5.1. Mechanizing termination proofs with the dependency pair framework

An appealing aspect of the DP-framework [GTSK04, GTSKF06] is that the precedence of *pairs* does not matter; they can be *independent* from the considered TRS. The notion of chain is parametric on both a TRS \mathcal{R} and a set of pairs \mathcal{P} (a TRS, actually) which are connected by using \mathcal{R} -rewrite sequences. Regarding termination proofs, the central notion now 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 $\mathcal{P} = \text{DP}(\mathcal{R})$. 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.

Processors are used in a pipe (more precisely, a *tree*) to incrementally simplify the original DP-termination problem as much as possible, possibly decomposing it into smaller pieces which are then independently treated in the very same way. The trivial case of this *iterative* process comes when the set of pairs \mathcal{P} becomes empty. Then, no infinite chain is possible and we can provide a *positive* answer **yes** to the DP-termination problem which is propagated upwards to the original problem in the root of the tree. In some cases it is also possible to witness the existence of infinite chains for a given DP-termination problem; then a *negative* answer **no** can be provided and propagated upwards. Of course, DP-termination problems are undecidable (in general), thus **don’t know** answers can also be generated (for instance by a time-out system which interrupts the usually complex search processes which are involved in the proofs). When all answers are collected, a final conclusion about the whole DP-termination problem can be given:

1. if we have positive answers (**yes**) for all problems in the leaves of the tree, then we conclude **yes** as well;
2. if a negative answer (**no**) was raised somewhere and the DP-processors which

were used in the path from the root to the node producing the negative answer were *complete*, then we conclude *no* as well;

3. Otherwise, the conclusion is don't know.

The notions of graph, cycles, SCCs, etc., are also part of the framework but (1) they are incorporated as *processors* now, and (2) they do *not* refer to dependency pairs anymore, but rather to the pairs in the (different) sets of pairs which are obtained through the process sketched above. In this way, we obtain a much more flexible framework to mechanize termination proofs and also to benefit from new future developments which could lead to the introduction of new processors.

In the following, we adapt these ideas to CSR to provide a suitable framework for mechanizing proofs of termination of CSR using CSDPs.

5.2. CS-termination problems and processors

The following definition adapts the notion of (DP-)termination problem defined in [GTSKF06] to CSR. In our definition, we prefer to avoid ‘*DP*’ because, as discussed above, dependency pairs (as such) are relevant in the theoretical framework only for investigating a particular problem (termination of TRSs), whereas some transformations can yield sets of pairs which are not dependency pairs of the underlying TRS anymore.

Definition 40 (CS-termination problems) *A CS-termination problem τ is a tuple $\tau = (\mathcal{P}, \mathcal{R}, \mu)$, where $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ are TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. A CS-termination problem is finite if there is no infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain.*

Finite CS-termination problems correspond to those generating a positive answer *yes* in the full proof process sketched above. Accordingly, CS-termination problems which are *not* finite correspond to a negative answer *no*.

Remark 41 According to Corollary 39, we can say now that a TRS \mathcal{R} is μ -terminating if and only if the CS-termination problem $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ is finite. ■

The following definition adapts the notion of *processor* [GTSKF06] to CSR.

Definition 42 (CS-processor) *A CS-processor Proc is a mapping from CS-termination problems into sets of CS-termination problems. A CS-processor Proc is*

- sound *if for all CS-termination problems τ , τ is finite whenever τ' is finite for all $\tau' \in \text{Proc}(\tau)$.*
- complete *if for all CS-termination problems τ , whenever τ is finite, then τ' is finite for all $\tau' \in \text{Proc}(\tau)$.*

In the following sections we describe a number of sound and (most of them) complete CS-processors.

6

CS-Termination Processors

In this chapter, we develop different processors that allow us to solve CS-termination problems. As explained in the previous chapter, these processors can be combined to treat CS-termination problems in such a way that after a finite number of processing steps we (hopefully) get a final response (finite or infinite).

6.1. Context-Sensitive Dependency Graph

CS-termination problems focus our attention on the analysis of *infinite minimal* $(\mathcal{P}, \mathcal{R}, \mu)$ -chains. In general, an infinite sequence $S = a_1, a_2, \dots, a_n, \dots$ of objects a_i belonging to a set A can be represented as a path in a *graph* G whose nodes are the objects in A , and whose arcs among them are appropriately established to represent S (in particular, an arc from a_i to a_{i+1} should be established if we want to be able to capture the sequence above). Actually, if A is *finite*, then the infinite sequence S defines at least one *cycle* in G : since there is a finite number of different objects $a_i \in A$ in S , there is an infinite tail $S' = a_m, a_{m+1}, \dots$ of S where *all objects* a_i occur *infinitely many times* for all $i \geq m$. This clearly corresponds to a cycle in G .

In the dependency pairs approach [AG00], a *dependency graph* $DG(\mathcal{R})$ is associated to the TRS \mathcal{R} . The nodes of the dependency graph are the dependency pairs in $DP(\mathcal{R})$; there is an arc from a dependency pair $u \rightarrow v$ to a dependency pair $u' \rightarrow v'$ if there are substitutions θ and θ' such that $\theta(v) \rightarrow_{\mathcal{R}}^* \theta'(u')$.

In more recent approaches, the analysis of infinite chains of dependency pairs as such is just a starting point. Many often, chains of dependency pairs are *transformed* into chains of more general *pairs* which cannot be considered dependency pairs anymore. This is the case for the *narrowing* or *instantiation* transformations, among others, see [GTSKF06] for instance. Still, the analysis of the cycles in the graph build out from such pairs is useful to investigate the existence of infinite (minimal)

chains of pairs. Thus, a more general notion of *graph of pairs* $DG(\mathcal{P}, \mathcal{R})$ associated to a set of pairs \mathcal{P} and a TRS \mathcal{R} is considered; the pairs in \mathcal{P} are used now as the nodes of the graph but they are connected by \mathcal{R} -rewriting in the same way [GTSKF06, Definition 7].

In the following section we take into account these points to provide an appropriate definition of context-sensitive (dependency) graph.

6.1.1. Definition of the context-sensitive dependency graph

According to the discussion above, our starting point are two TRSs $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ together with a replacement map $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. Our aim is obtaining a notion of graph which is able to represent all infinite *minimal* chains of pairs as given in Definition 35.

When considering pairs $u \rightarrow v \in \mathcal{P}_{\mathcal{G}}$, we can proceed as in the standard case to define appropriate connections to other pairs $u' \rightarrow v' \in \mathcal{P}$: there is an arc from $u \rightarrow v$ to $u' \rightarrow v'$ if $\theta(v) \hookrightarrow_{\mathcal{R}, \mu}^* \theta'(u')$ for some substitutions θ and θ' . When considering collapsing pairs $u \rightarrow v \in \mathcal{P}_{\mathcal{X}}$, we know that such pairs can only be followed by a pair $u' \rightarrow v' \in \mathcal{P}$ such that $\theta(t^\sharp) \hookrightarrow_{\mathcal{R}, \mu}^* \theta'(u')$ for some $t \in \mathcal{NHT}$ and substitutions θ and θ' (see Definition 35).

Definition 43 (Context-Sensitive Graph of Pairs) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. The context-sensitive (CS-)graph associated to \mathcal{R} and \mathcal{P} (denoted $G(\mathcal{P}, \mathcal{R}, \mu)$) has \mathcal{P} as the set of nodes and arcs which connect them as follows:*

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

In termination proofs, we are concerned with the so-called *strongly connected components* (SCCs) of the dependency graph, rather than with the cycles themselves (which are exponentially many) [HM05]. 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 $(\mathcal{P}, \mathcal{R}, \mu)$ -chains.

Theorem 44 (SCC processor [AGL08]) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. Then, the processor Proc_{SCC} given by*

$$\text{Proc}_{SCC}(\mathcal{P}, \mathcal{R}, \mu) = \{(\mathcal{Q}, \mathcal{R}, \mu) \mid \mathcal{Q} \text{ contains the pairs of an SCC in } \mathbb{G}(\mathcal{P}, \mathcal{R}, \mu)\}$$

is sound and complete.

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

Now we can use these notions to introduce the context-sensitive dependency graph.

Definition 45 (Context-Sensitive Dependency Graph) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. The Context-Sensitive Dependency Graph associated to \mathcal{R} and μ is $\text{DG}(\mathcal{R}, \mu) = \mathbb{G}(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$.*

6.1.2. Estimating the CS-dependency graph

In general, the (context-sensitive) dependency graph of a TRS is *not* computable: it involves reachability of $\theta'(u')$ from $\theta(v)$ (for $u \rightarrow v \in \mathcal{P}_{\mathcal{G}}$) or $\theta(t^\sharp)$ (for $t \in \mathcal{NHT}_{\mathcal{P}}$) using CSR; as in the unrestricted case, the reachability problem for CSR is undecidable. So, we need to use some approximation of it. Following [AG00], we describe how to approximate the CS-dependency graph of a CS-TRS.

Given a set Δ of ‘defined’ symbols, we let $\text{CAP}_{\Delta}^{\mu}$ be as follows:

$$\begin{aligned} \text{CAP}_{\Delta}^{\mu}(x) &= x \text{ if } x \text{ is a variable} \\ \text{CAP}_{\Delta}^{\mu}(f(t_1, \dots, t_k)) &= \begin{cases} y & \text{if } f \in \Delta \\ f([t_1]_1^f, \dots, [t_k]_1^f) & \text{otherwise} \end{cases} \end{aligned}$$

where y is intended to be a new, fresh variable which has not yet been used and given a term s , $[s]_i^f = \text{CAP}_{\Delta}^{\mu}(s)$ if $i \in \mu(f)$ and $[s]_i^f = s$ if $i \notin \mu(f)$.

Function $\text{CAP}_{\Delta}^{\mu}$ is intended to provide a suitable approximation of reachability problems $\theta(s) \xrightarrow{*}_{\mathcal{R}, \mu} \theta'(t)$ by means of unification. The idea is obtaining the maximal prefix context $C[\square]$ of s (i.e., $s = C[s_1, \dots, s_n]$ for some terms s_1, \dots, s_n) which we know (without any ‘look-ahead’ for applicable rules) that cannot be changed by any reduction starting from s . Furthermore, terms s_1, \dots, s_n above must be rooted by defined symbols (i.e., $\text{root}(s_i) \in \Delta$ for $i \in \{1, \dots, n\}$). Now, we replace those subterms s_i which are at μ -replacing positions (i.e., $s_i = s|_{p_i}$ for some $p_i \in \text{Pos}^{\mu}(s)$) by some variable x , and we leave untouched the non- μ -replacing ones.

The following result whose proof is similar to that of [AG00, Theorem 21] (we only need to take into account the replacement restrictions indicated by the replacement map μ) formalizes the soundness of this approach.

Proposition 46 ([AGL08]) *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ be such that $\text{Var}(s) \cap \text{Var}(t) = \emptyset$ and θ, θ' be substitutions. If $\theta(s) \hookrightarrow_{\mathcal{R}, \mu}^* \theta'(t)$, then $\text{REN}^{\mu}(\text{CAP}_{\mathcal{D}}^{\mu}(s))$ and t unify.*

Example 47

Consider the well-known Toyama's example [Toy87]. Note that REN^{μ} is necessary to simulate reachability. Otherwise:

$$\begin{aligned} f(\mathbf{a}, \mathbf{b}, x) &\rightarrow f(x, x, x) \\ c &\rightarrow a \\ c &\rightarrow b \end{aligned}$$

The only dependency pair for this system is:

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

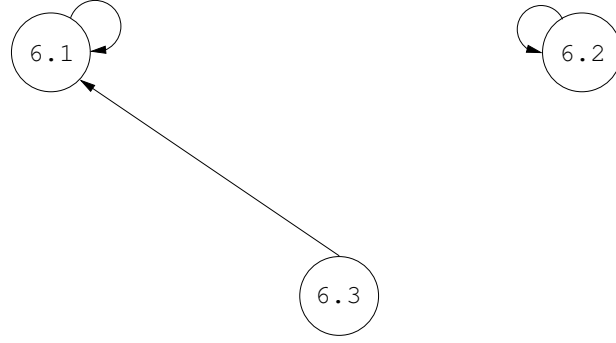
Without considering REN^{μ} , $F(x, x, x)$ does not unify with $F(\mathbf{a}, \mathbf{b}, y)$, but it is possible to rewrite $F(\mathbf{c}, \mathbf{c}, \mathbf{c})$ into $F(\mathbf{a}, \mathbf{b}, \mathbf{c})$.

According to Proposition 46, given terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and substitutions θ, θ' , the reachability of $\theta'(t)$ from $\theta(s)$ by μ -rewriting can be *approximated* as unification of $\text{REN}^{\mu}(\text{CAP}_{\mathcal{D}}^{\mu}(s))$ and t . So, we have the following.

Definition 48 (Estimated Context-Sensitive Graph of Pairs) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. The estimated CS-graph associated to \mathcal{R} and \mathcal{P} (denoted $\text{EG}(\mathcal{P}, \mathcal{R}, \mu)$) has \mathcal{P} as the set of nodes and arcs which connect them as follows:*

1. *There is an arc from $u \rightarrow v \in \mathcal{P}_{\mathcal{G}}$ to $u' \rightarrow v' \in \mathcal{P}$ if $\text{REN}^{\mu}(\text{CAP}_{\mathcal{D}}^{\mu}(v))$ and u' unify.*
2. *There is an arc from $u \rightarrow v \in \mathcal{P}_{\mathcal{X}}$ to $u' \rightarrow v' \in \mathcal{P}$ if there is $t \in \mathcal{NHT}(\mathcal{R}, \mu)$ such that $\text{REN}^{\mu}(\text{CAP}_{\mathcal{D}}^{\mu}(t^{\sharp}))$ and u' unify.*

According to Definition 43, we would have the corresponding one for the *estimated* CSDG: $\text{EDG}(\mathcal{R}, \mu) = \text{EG}(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^{\sharp})$.

Figure 6.1: Estimated CSDG for the CS-TRS (\mathcal{R}, μ) in Example 49**Example 49**

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

$$\begin{array}{ll}
 f(x) \rightarrow \text{cons}(x, f(g(x))) & \text{sel}(0, \text{cons}(x, y)) \rightarrow x \\
 g(0) \rightarrow s(0) & \text{sel}(s(x), \text{cons}(y, z)) \rightarrow \text{sel}(x, z) \\
 g(s(x)) \rightarrow s(s(g(x))) &
 \end{array}$$

with $\mu(0) = \emptyset$, $\mu(f) = \mu(g) = \mu(s) = \mu(\text{cons}) = \{1\}$, and $\mu(\text{sel}) = \{1, 2\}$. Then, $\text{DP}(\mathcal{R}, \mu)$ is:

$$G(s(x)) \rightarrow G(x) \quad (6.1) \quad \text{SEL}(s(x), \text{cons}(y, z)) \rightarrow z \quad (6.3)$$

$$\text{SEL}(s(x), \text{cons}(y, z)) \rightarrow \text{SEL}(x, z) \quad (6.2)$$

and $\mathcal{NHT} = \{f(g(x)), g(x)\}$. Regarding pairs (6.1) and (6.2) in $\text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$, there is an arc from (6.1) to itself and another one from (6.2) to itself. Regarding the only collapsing pair (6.3), we have that $\text{REN}^{\mu}(\text{CAP}_{\mathcal{D}}^{\mu}(F(g(x)))) = F(y)$ and $\text{REN}^{\mu}(\text{CAP}_{\mathcal{D}}^{\mu}(G(x))) = G(y)$. Since $F(y)$ does not unify with the left-hand side of any pair, and $G(y)$ unifies with the left-hand side $G(s(x))$ of (6.1), there is an arc from (6.3) to (6.1), see Figure 6.1. Thus, there are two cycles: $\{(6.1)\}$ and $\{(6.2)\}$.

Note that Proposition 46 also provides a way to *estimate* the set $\mathcal{NHT}_{\mathcal{P}}$: if $t \in \mathcal{NHT}_{\mathcal{P}}$, then $\text{REN}^{\mu}(\text{CAP}_{\mathcal{D}}^{\mu}(t^{\sharp}))$ and u unify for some $u \rightarrow v \in \mathcal{P}$. In the following, our presentations of $\mathcal{NHT}_{\mathcal{P}}$ in the examples are computed in this way.

Example 50

Consider again the CS-TRS (\mathcal{R}, μ) in Example 1. Note that

$$\mathcal{NHT}_{\text{DP}(\mathcal{R}, \mu)}(\mathcal{R}, \mu^{\sharp}) = \{\text{filt}(x, \text{sieve}(y)), \text{filt}(s(s(x)), z)\}$$

The CSDG is shown in Figure 6.2 and has *no* cycle. By Theorem 44 we transform the CS-problem $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ into a singleton $\{(\emptyset, \mathcal{R}, \mu^\sharp)\}$ containing a finite CS-termination problem. Thus, we conclude that \mathcal{R} is μ -terminating.

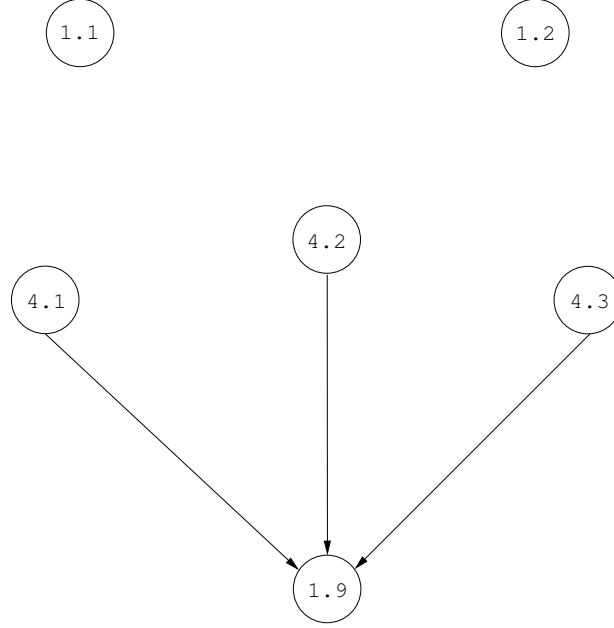


Figure 6.2: Context-Sensitive Dependency Graph for the CS-TRS (\mathcal{R}, μ) in Example 1

Example 51

Consider again the CS-TRS (\mathcal{R}, μ) in Example 20. Note that

$$\mathcal{NHT}_{\text{DP}(\mathcal{R}, \mu)}(\mathcal{R}, \mu^\sharp) = \{\text{quot}(xs, ys)\}$$

The CSDG is shown in Figure 6.3 and has three cycles. By Theorem 44 we transform the CS-problem $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$ into a set $\{(\mathcal{P}_1, \mathcal{R}, \mu^\sharp), (\mathcal{P}_2, \mathcal{R}, \mu^\sharp), (\mathcal{P}_3, \mathcal{R}, \mu^\sharp)\}$ of three CS-termination problems where:

$$\begin{aligned} \mathcal{P}_1 &= \{\{\text{SEL}(s(n), \text{cons}(x, xs)) \rightarrow \text{SEL}(n, xs)\}\} \\ \mathcal{P}_2 &= \{\{\text{MINUS}(s(x), s(y)) \rightarrow \text{MINUS}(x, y)\}\} \\ \mathcal{P}_3 &= \{\{\text{QUOT}(s(x), s(y)) \rightarrow \text{QUOT}(\text{minus}(x, y), s(y))\}\} \end{aligned}$$

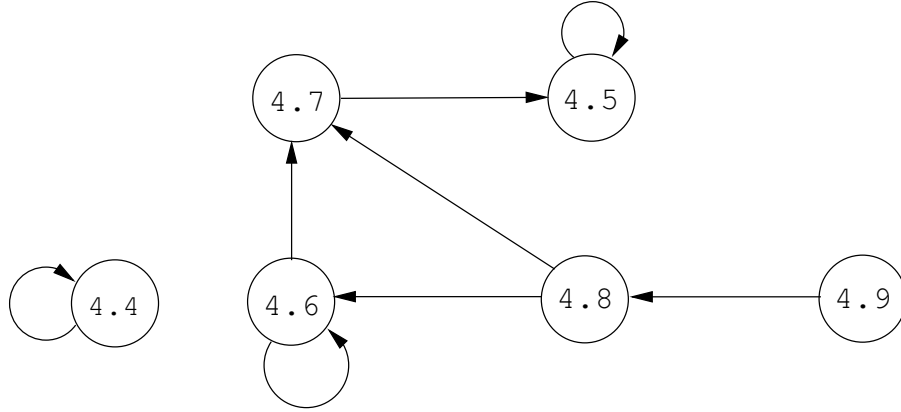


Figure 6.3: Context-Sensitive Dependency Graph for the CS-TRS (\mathcal{R}, μ) in Example 20

6.2. Treating collapsing pairs

The following result introduces a sound and complete CS-processor to transform collapsing pairs into noncollapsing ones. In order to appropriately formulate it, we first define the notion of hiding replacement map. Given a CS-TRS (\mathcal{R}, μ) , it collects the arguments which are hidden by each symbol in the signature \mathcal{F} of \mathcal{R} (see Definition 18).

Definition 52 (Hiding replacement map) Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. We define the hiding context map $\mu_{\mathcal{H}, \mathcal{R}, \mu}$ as a mapping $\mu_{\mathcal{H}, \mathcal{R}, \mu} : \mathcal{F} \rightarrow \wp(\mathbb{N})$ such that:

$$\mu_{\mathcal{H}, \mathcal{R}, \mu}(f) = \{i \mid i \in \mu(f) \text{ and } f \text{ hides position } i \text{ in } (\mathcal{R}, \mu)\}$$

Note that $\mu_{\mathcal{H}, \mathcal{R}, \mu} \subseteq \mu$.

Theorem 53 (Collapsing pairs transformation processor) Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$.

Let $u \rightarrow x \in \mathcal{P}_{\mathcal{X}}$ and

$$P_u = \{u \rightarrow \mathbf{U}(x)\} \cup \{\mathbf{U}(f(x_1, \dots, x_k)) \rightarrow \mathbf{U}(x_i) \mid f \in \mathcal{F}, i \in \mu_{\mathcal{H}, \mathcal{R}, \mu}(f)\} \\ \cup \{\mathbf{U}(t) \rightarrow t^\sharp \mid t \in \mathcal{NHT}_{\mathcal{P}}\}$$

where \mathbf{U} is a new fresh symbol. Let $\mathcal{P}' = (\mathcal{G} \cup \{\mathbf{U}\}, P')$ where $P' = (P \setminus \{u \rightarrow x\}) \cup P_u$, and μ' which extends μ by $\mu'(\mathbf{U}) = \emptyset$.

Then, the processor $\text{Proc}_{e\text{Coll}}$ given by

$$\text{Proc}_{e\text{Coll}}(\mathcal{P}, \mathcal{R}, \mu) = \{(\mathcal{P}', \mathcal{R}, \mu')\}$$

is sound and complete.

Proof. Soundness. We prove first that the existence of an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain implies the existence of an infinite minimal $(\mathcal{P}', \mathcal{R}, \mu')$ -chain.

First, note that \mathcal{P}' is well-defined as a TRS. Consider an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain A :

$$\sigma(u_1) \hookrightarrow_{\mathcal{P}, \mu} \circ \succeq_{\mu}^{\#} t_1 \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_2) \hookrightarrow_{\mathcal{P}, \mu} \circ \succeq_{\mu}^{\#} t_2 \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_3) \hookrightarrow_{\mathcal{P}, \mu} \circ \succeq_{\mu}^{\#} \dots$$

for some substitution σ , where, for all $i \geq 1$, t_i is μ -terminating and, (1) if $u_i \rightarrow v_i \in \mathcal{P}_{\mathcal{G}}$, then $t_i = \sigma(v_i)$ and (2) if $u_i \rightarrow v_i = u_i \rightarrow x_i \in \mathcal{P}_{\mathcal{X}}$, then $t_i = s_i^{\#}$ for some s_i such that $\sigma(x_i) = C[s_i]$ and $C[s_i] = \theta_i(\bar{C}[\bar{s}_i])$ for some $\bar{s}_i \in \mathcal{NHT}$, some hiding context $\bar{C}[\square]$ and substitution θ_i ; actually, since $t_i = s_i^{\#} = \theta_i(\bar{s}_i)^{\#} = \theta_i(\bar{s}_i^{\#})$ and $t_i \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_{i+1})$, we can further say that $\bar{s}_i \in \mathcal{NHT}_{\mathcal{P}}$. Hence, we obtain:

$$\begin{array}{ll} \sigma(u_i) & \\ \xrightarrow{\Lambda}_{\mathcal{P}', \mu'} & \text{U}(\sigma(x_i)) \quad \text{a collapsing pair is applied} \\ = & \text{U}(C[s_i]) \quad C[\square] \text{ is an instance of a hiding context} \\ \xrightarrow{\Lambda^*}_{\mathcal{P}', \mu'} & \text{U}(s_i) \quad \text{we remove the hiding context with the U rules} \\ \xrightarrow{\Lambda}_{\mathcal{P}', \mu'} & s_i^{\#} \quad s_i \text{ is an instance of a term in } \mathcal{NHT} \end{array}$$

Thus, we obtain an infinite minimal $(\mathcal{P}', \mathcal{R}, \mu')$ -chain, as desired. In particular, we note that all steps with \mathcal{P}_u are performed at the root, we do not require any reduction below symbol U , hence $\mu'(\text{U}) = \emptyset$ is enough to perform them.

Completeness. By contradiction. If there is an infinite $(\mathcal{P}', \mathcal{R}, \mu')$ -chain, then there is a substitution σ and pairs $u_i \rightarrow v_i \in \mathcal{P}'$ such that

1. If $u_i \rightarrow v_i \in \mathcal{P}' \setminus \mathcal{P}_u$ and $\sigma(v_i) \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_{i+1})$, then $u_i \rightarrow v_i \in \mathcal{P}$ and $\sigma(v_i) \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_{i+1})$,
2. if $u_i \rightarrow v_i = u_i \rightarrow \text{U}(x_i) \in \mathcal{P}'$ and $\sigma(u_i) \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(v_{i+1})$ where $\sigma(v_{i+1}) = \text{U}(v'_{i+1})$, then there is a pair $u_i \rightarrow x_i \in \mathcal{P}$ such that $\sigma(x_i) = v'_{i+1}$,
3. if $u_i \rightarrow v_i = \text{U}(f(x_1, \dots, x_n)) \rightarrow \text{U}(x_i) \in \mathcal{P}'$ and $\sigma(v_i) \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(v_{i+1})$ where $\sigma(v_i) = \text{U}(v'_i)$ and $\sigma(v_{i+1}) = \text{U}(v'_{i+1})$, then $i \in \mu_{\mathcal{H}, \mathcal{R}, \mu}(f)$ and $v'_i = C_i[v'_{i+1}]$ where $C'_i[\square]$ is an instance of a hiding context, and

4. if $u_i \rightarrow v_i = U(t) \rightarrow t^\sharp \in \mathcal{P}'$ and $\sigma(v_i) \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_{i+1})$, then there is $t \in \mathcal{NHT}$ such that $\sigma(t)^\sharp \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_{i+1})$.

Hence, we can build in that way an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain which contradicts the μ -termination of \mathcal{R} . ■

Example 54

Consider the following TRS [AEF⁺08]:

$$\begin{array}{llll}
\text{gt}(0, y) & \rightarrow & \text{false} & \quad \text{p}(0) \rightarrow 0 \\
\text{gt}(s(x), 0) & \rightarrow & \text{true} & \quad \text{p}(s(x)) \rightarrow x \\
\text{gt}(s(x), s(y)) & \rightarrow & \text{gt}(x, y) & \quad \text{minus}(x, y) \rightarrow \text{if}(\text{gt}(y, 0), \text{minus}(\text{p}(x), \text{p}(y)), x) \\
\text{if}(\text{true}, x, y) & \rightarrow & x & \quad \text{div}(0, s(y)) \rightarrow 0 \\
\text{if}(\text{false}, x, y) & \rightarrow & y & \quad \text{div}(s(x), s(y)) \rightarrow s(\text{div}(\text{minus}(x, y), s(y)))
\end{array}$$

with $\mu(\text{if}) = \{1\}$ and $\mu(f) = \{1, \dots, \text{ar}(f)\}$ for all other symbols f .

We obtain the following CS-DPs.

$$\text{GT}(s(x), s(y)) \rightarrow \text{GT}(x, y) \tag{6.4}$$

$$\text{DIV}(s(x), s(y)) \rightarrow \text{MINUS}(x, y) \tag{6.5}$$

$$\text{DIV}(s(x), s(y)) \rightarrow \text{DIV}(\text{minus}(x, y), s(y)) \tag{6.6}$$

$$\text{IF}(\text{true}, x, y) \rightarrow x \tag{6.7}$$

$$\text{IF}(\text{false}, x, y) \rightarrow y \tag{6.8}$$

$$\text{MINUS}(x, y) \rightarrow \text{GT}(y, 0) \tag{6.9}$$

$$\text{MINUS}(x, y) \rightarrow \text{IF}(\text{gt}(y, 0), \text{minus}(\text{p}(x), \text{p}(y)), x) \tag{6.10}$$

Applying the SCCs processor to the CS-problem $(\text{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu^\sharp)$, we obtain the graph shown in Figure 6.4. And the resulting problems is a set of CS-problems $\{(\mathcal{P}_1, \mathcal{R}, \mu^\sharp), (\mathcal{P}_2, \mathcal{R}, \mu^\sharp), (\mathcal{P}_3, \mathcal{R}, \mu^\sharp)\}$ where set $\mathcal{P}_1 = \{(6.4)\}$, set $\mathcal{P}_2 = \{(6.6)\}$, and set $\mathcal{P}_3 = \{(6.7), (6.8), (6.10)\}$. If we apply the collapsing pairs transformation processor to the CS-termination problem $(\mathcal{P}_3, \mathcal{R}, \mu^\sharp)$, we obtain the CS-termination problem $(\{(6.10)\} \cup \mathcal{P}_u, \mathcal{R}, \mu')$ where \mathcal{P}_u is formed by the following pairs:

$$\text{IF}(\text{true}, x, y) \rightarrow U(x) \tag{6.11}$$

$$\text{IF}(\text{false}, x, y) \rightarrow U(y) \tag{6.12}$$

$$\text{MINUS}(x, y) \rightarrow \text{GT}(y, 0) \tag{6.13}$$

$$\text{MINUS}(x, y) \rightarrow \text{IF}(\text{gt}(y, 0), \text{minus}(\text{p}(x), \text{p}(y)), x) \tag{6.14}$$

$$U(\text{p}(x)) \rightarrow U(x) \tag{6.15}$$

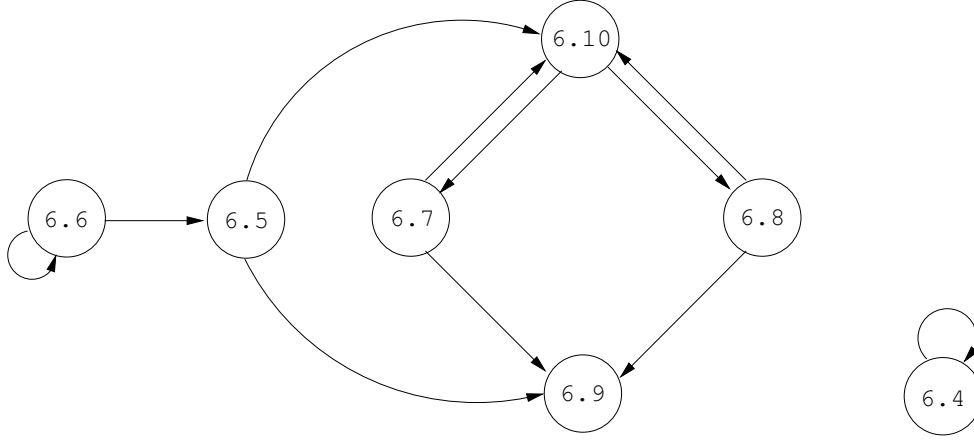


Figure 6.4: Dependency Graph for the CS-DPs in Example 54

$$U(p(y)) \rightarrow U(y) \quad (6.16)$$

$$U(\text{minus}(x, y)) \rightarrow U(x) \quad (6.17)$$

$$U(\text{minus}(p(x), p(y))) \rightarrow \text{MINUS}(p(x), p(y)) \quad (6.18)$$

$$U(\text{minus}(x, y)) \rightarrow U(y) \quad (6.19)$$

$$U(p(x)) \rightarrow P(x) \quad (6.20)$$

where μ' extends μ^\sharp with $\mu'(U) = \emptyset$.

6.3. Use of μ -reduction pairs

A reduction pair (\succsim, \sqsupset) consists of a stable and monotonic quasi-ordering \succsim , and a stable and well-founded ordering \sqsupset satisfying either $\succsim \circ \sqsupset \subseteq \sqsupset$ or $\sqsupset \circ \succsim \subseteq \sqsupset$ [KNT99].

The absence of infinite chains of (dependency) pairs can be ensured by finding a *reduction pair* (\succsim, \sqsupset) which is compatible with the rules and the dependency pairs [AG00]: $l \succsim r$ for all rewrite rules $l \rightarrow r$ and $u \succsim v$ or $u \sqsupset v$ for all dependency pairs $u \rightarrow v$. In the dependency pair framework [GTSK04, GTSKF06] (but also in [GAO02, HM04, HM05, HM07]), they are used to obtain *smaller* sets of pairs $\mathcal{P}' \subseteq \mathcal{P}$ by removing the *strict* pairs, i.e., those pairs $u \rightarrow v \in \mathcal{P}$ such that $u \sqsupset v$.

Stability is required both for \succsim and \sqsupset because, although we only check the left- and right-hand sides of the rewrite rules $l \rightarrow r$ (with \succsim) and pairs $u \rightarrow v$ (with \succsim or \sqsupset), the chains of pairs involve *instances* $\sigma(l)$, $\sigma(r)$, $\sigma(u)$, and $\sigma(v)$ of rules and pairs and we aim at concluding $\sigma(l) \succsim \sigma(r)$, and $\sigma(u) \succsim \sigma(v)$ or $\sigma(u) \sqsupset \sigma(v)$, respectively.

Monotonicity is required for \succsim to deal with the application of rules $l \rightarrow r$ to an arbitrary depth in terms. Since the pairs are ‘applied’ only at the root level, no monotonicity is required for \sqsupset (but, for this reason, we cannot compare the rules in \mathcal{R} using \sqsupset). Recently, Endrullis et al. noticed that *transitivity* is not necessary for the strict component \sqsupset because it is somehow ‘simulated’ by the compatibility requirement above [EWZ08].

In our setting, since we are interested in μ -rewriting steps only, we can relax the monotonicity requirements as follows.

Definition 55 (μ -reduction pair) *Let \mathcal{F} be a signature and $\mu \in M_{\mathcal{F}}$. A μ -reduction pair (\succsim, \sqsupset) consists of a stable and μ -monotonic quasi-ordering \succsim and a well-founded stable relation \sqsupset on terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$ which are compatible, i.e., $\succsim \circ \sqsupset \subseteq \sqsupset$ or $\sqsupset \circ \succsim \subseteq \sqsupset$.*

We say that (\succsim, \sqsupset) is μ -monotonic if \sqsupset is μ -monotonic.

Reduction pairs are often used in combination with *argument filterings*, which discard subexpressions from constraints $s \succsim t$ or $s \sqsupset t$ in such a way that $\pi(s) \succsim \pi(t)$ (resp. $\pi(s) \sqsupset \pi(t)$) is often simpler to prove [AG00, GTSKF06].

6.3.1. Argument filterings for CSR

An argument filtering π for a signature \mathcal{F} is a mapping that assigns to every k -ary function symbol $f \in \mathcal{F}$ an argument position $i \in \{1, \dots, k\}$ or a (possibly empty) list $[i_1, \dots, i_m]$ of argument positions with $1 \leq i_1 < \dots < i_m \leq k$ [KNT99]. In the following, by the *trivial* argument filtering π_{\top} for \mathcal{F} , we mean the one given by $\pi_{\top}(f) = [1, \dots, k]$ for each k -ary symbol $f \in \mathcal{F}$. It corresponds to the argument filtering which does nothing.

We can use an argument filtering π to ‘filter’ either the signature \mathcal{F} or any replacement map $\mu \in M_{\mathcal{F}}$. In the following, we assume that:

1. The signature \mathcal{F}_{π} consists of all function symbols f such that $\pi(f)$ is some list $[i_1, \dots, i_m]$, where, in \mathcal{F}_{π} , the arity of f is m . As usual, we give the same name to the version of $f \in \mathcal{F}$ which belongs to \mathcal{F}_{π} .
2. The replacement map $\mu_{\pi} \in M_{\mathcal{F}_{\pi}}$ is given as follows: for all $f \in \mathcal{F}$ such that $f \in \mathcal{F}_{\pi}$ and $\pi(f) = [i_1, \dots, i_m]$:

$$\mu_{\pi}(f) = \{j \in \{1, \dots, m\} \mid i_j \in \mu(f)\}$$

An argument filtering induces a mapping from $\mathcal{T}(\mathcal{F}, \mathcal{X})$ to $\mathcal{T}(\mathcal{F}_\pi, \mathcal{X})$, also denoted by π :

$$\pi(t) = \begin{cases} t & \text{if } t \text{ is a variable} \\ \pi(t_i) & \text{if } t = f(t_1, \dots, t_k) \text{ and } \pi(f) = i \\ f(\pi(t_{i_1}), \dots, \pi(t_{i_m})) & \text{if } t = f(t_1, \dots, t_k) \text{ and } \pi(f) = [i_1, \dots, i_m] \end{cases}$$

Note that, for the *top* filtering π_\top , we have that $\mathcal{F}_{\pi_\top} = \mathcal{F}$, $\mu_{\pi_\top} = \mu$ for all $\mu \in M_{\mathcal{F}}$, and $\pi_\top(t) = t$ for all $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$.

In the following, given a substitution σ and an argument filtering π , we let σ_π be the substitution defined by $\sigma_\pi(x) = \pi(\sigma(x))$ for all $x \in \mathcal{X}$. The following auxiliary results are used later.

Lemma 56 ([AGL08]) *Let \mathcal{F} be a signature, π be an argument filtering for \mathcal{F} and σ be a substitution. If $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, then, $\pi(\sigma(t)) = \sigma_\pi(\pi(t))$.*

Proposition 57 ([AGL08]) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, and π be an argument filtering for \mathcal{F} . Let (\succsim, \sqsupset) be a μ_π -reduction pair such that $\pi(l) \succsim \pi(r)$ for all $l \rightarrow r \in R$, and let $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. If $s \hookrightarrow_{\mathcal{R}, \mu}^* t$, then $\pi(s) \succsim \pi(t)$.*

6.3.2. Removing pairs using μ -reduction orderings

For a given TRS $\mathcal{R} = (\mathcal{F}, R)$, set of pairs $\mathcal{P} = (\mathcal{G}, P)$, and replacement map $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$, checking the absence of infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chains can often be ‘simplified’ to checking the absence of infinite minimal $(\mathcal{P}', \mathcal{R}, \mu)$ -chains for a proper subset $\mathcal{P}' \subset \mathcal{P}$ by finding appropriate μ -reduction pairs (\succsim, \sqsupset) . The presence of *collapsing* pairs $u \rightarrow v = u \rightarrow x \in \mathcal{P}_{\mathcal{X}}$ imposes some additional requirements on the μ -reduction pairs. Basically,

1. We need to ensure that the quasi-ordering \succsim is able to ‘look’ for a μ -replacing subterm $s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ inside the instantiation $\sigma(x) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ of a migrating variable x : we know that $\sigma(x) = C[s]$ where $C[\square]$ is an instance of a hiding context. Hence we require that for all $f \in \mathcal{F}$ and for all position $i \in \mu_{\mathcal{H}, \mathcal{R}, \mu}(f)$ we have that $f(x_1, \dots, x_i, \dots, x_n) \succsim x_i$, where $\text{ar}(f) = n$.
2. We need to connect the marked version s^\sharp of s (which is known to be an instance of a hidden term $t \in \mathcal{NHT}_{\mathcal{P}}$, i.e., $s = \theta(t)$ for some substitution θ) with an instance $\sigma(u)$ of the left-hand side u of a pair; hence the requirement $t \succsim t^\sharp$ or $t \sqsupset t^\sharp$ for all $t \in \mathcal{NHT}_{\mathcal{P}}$ which, by stability, becomes $s \succsim s^\sharp$ or $s \sqsupset s^\sharp$.

The following theorem formalizes a generic processor to remove pairs from \mathcal{P} by using argument filterings and μ -reduction pairs.

Theorem 58 (μ -reduction pair processor) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs, $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. Let π be an argument filtering for $\mathcal{F} \cup \mathcal{G}$ and (\succsim, \sqsupset) be a $\mu\pi$ -reduction pair such that*

1. $\pi(\mathcal{R}) \subseteq \succsim$, $\pi(\mathcal{P}) \subseteq \succsim \cup \sqsupset$, and
2. whenever $\mathcal{NHT}_{\mathcal{P}} \neq \emptyset$ and $\mathcal{P}_{\mathcal{X}} \neq \emptyset$, we have that
 - a) for all $f \in \mathcal{F}$, either $\pi(f) = [i_1, \dots, i_m]$ and $\mu_{\mathcal{H}, \mathcal{R}, \mu}(f) \subseteq \pi(f)$, or $\pi(f) = i$ and $\mu(f) = \{i\}$,
 - b) $\pi(f(x_1, \dots, x_i, \dots, x_n)) \succsim x_i$ for all $f \in \mathcal{F}$ and for all position $i \in \mu_{\mathcal{H}, \mathcal{R}, \mu}(f)$, and
 - c) $\pi(t) (\succsim \cup \sqsupset) \pi(t^\sharp)$ for all $t \in \mathcal{NHT}_{\mathcal{P}}$,

Let $\mathcal{P}_{\sqsupset} = \{u \rightarrow v \in \mathcal{P} \mid \pi(u) \sqsupset \pi(v)\}$. Then, the processor Proc_{RP} given by

$$\text{Proc}_{RP}(\mathcal{P}, \mathcal{R}, \mu) = \begin{cases} \{(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathcal{R}, \mu)\} & \text{if (1) and (2) hold} \\ \{(\mathcal{P}, \mathcal{R}, \mu)\} & \text{otherwise} \end{cases}$$

is sound and complete.

Proof. We have to prove that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain if and only if there is an infinite minimal $(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathcal{R}, \mu)$ -chain. The *if* part is obvious. For the *only if* part, we proceed by contradiction. Assume that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain A , but that there is no infinite minimal $(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathcal{R}, \mu)$ -chain. Due to the finiteness of \mathcal{P} , we can assume that there is $\mathcal{Q} \subseteq \mathcal{P}$ such that A has a tail B

$$\sigma(u_1) \hookrightarrow_{\mathcal{Q}, \mu} \circ \triangleright_{\mu}^{\sharp} t_1 \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_2) \hookrightarrow_{\mathcal{Q}, \mu} \circ \triangleright_{\mu}^{\sharp} t_2 \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_3) \hookrightarrow_{\mathcal{Q}, \mu} \circ \triangleright_{\mu}^{\sharp} \dots$$

for some substitution σ , where all pairs in \mathcal{Q} are infinitely often used, and, for all $i \geq 1$, (1) if $u_i \rightarrow v_i \in \mathcal{Q}_{\mathcal{G}}$, then $t_i = \sigma(v_i)$ and (2) if $u_i \rightarrow v_i = u_i \rightarrow x_i \in \mathcal{Q}_{\mathcal{X}}$, then $t_i = s_i^{\sharp}$ for some s_i and some context $C[\square]$ with a μ -replacing hole such that $\sigma(x_i) = C[s_i]$, and $C[s_i] = \theta_i(\bar{C}[\bar{s}_i])$ for some $\bar{s}_i \in \mathcal{NHT}$, some hiding context $\bar{C}[\square]$ and substitution θ_i ; actually, since $t_i = s_i^{\sharp} = \theta_i(\bar{s}_i)^{\sharp} = \theta_i(s_i^{\sharp})$ and $t_i \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_{i+1})$, we can further say that $\bar{s}_i \in \mathcal{NHT}_{\mathcal{Q}}$.

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

No pair $u \rightarrow v \in \mathcal{Q}$ satisfies that $\pi(u) \sqsupset \pi(v)$. Otherwise, we get a contradiction by considering the following two cases:

1. If $u_i \rightarrow v_i \in \mathcal{Q}_G$, then $t_i = \sigma(v_i) \xrightarrow{*}_{\mathcal{R},\mu} \sigma(u_{i+1})$ and by Proposition 57, $\pi(t_i) \succeq \pi(\sigma(u_{i+1}))$. By Lemma 56, $\pi(t_i) \succeq \sigma_\pi(\pi(u_{i+1}))$. Since we have $\sigma_\pi(\pi(u_i)) (\succeq \cup \sqsupset) \sigma_\pi(\pi(v_i)) = \pi(\sigma(v_i)) = \pi(t_i)$ (using Lemma 56), by using transitivity of \succeq and compatibility between \succeq and \sqsupset , we conclude that $\sigma_\pi(\pi(u_i)) (\succeq \cup \sqsupset) \sigma_\pi(\pi(u_{i+1}))$.
2. If $u_i \rightarrow v_i = u_i \rightarrow x_i \in \mathcal{Q}_X$, then $\sigma(v_i) = \sigma(x_i) = C[s_i]$ for some context $C[\square]$ with a μ -replacing hole. Since $C[\square]$ is an instance of a hiding context $\bar{C}[\square]$ and $i \in \mu_{\mathcal{H},\mathcal{R},\mu}(f)$ implies that $i \in \pi(f)$, we can say that $\pi(\sigma(x)) = \sigma_\pi(x) = C_\pi[\pi(s_i)]$. Since $\pi(f(x_1, \dots, x_i, \dots, x_n)) \succeq \pi(x_i)$ for all $f \in \mathcal{F}$ and all position $i \in \mu_{\mathcal{H},\mathcal{R},\mu}(f)$ we have by stability $\sigma_\pi(\pi(u_i)) (\succeq \cup \sqsupset) \sigma_\pi(\pi(v_i)) = \sigma_\pi(x_i) \succeq \pi(s_i)$. Furthermore, we are assuming that $\pi(t) (\succeq \cup \sqsupset) \pi(t^\sharp)$ for all $t \in \mathcal{NHT}_{\mathcal{Q}} \subseteq \mathcal{NHT}_{\mathcal{P}}$. Since $s_i = \theta_i(\bar{s}_i)$, we have that $\pi(s_i) = \pi(\theta_i(\bar{s}_i)) = \theta_{i,\pi}(\pi(\bar{s}_i))$ (using Lemma 56 again) and, similarly, $\pi(s_i^\sharp) = \theta_{i,\pi}(\pi(\bar{s}_i^\sharp))$. By stability we have that $\pi(s_i) (\succeq \cup \sqsupset) \pi(s_i^\sharp)$. Hence, by transitivity of \succeq (and compatibility of \succeq and \sqsupset), we have $\sigma_\pi(\pi(v_i)) = \sigma_\pi(x_i) (\succeq \cup \sqsupset) \pi(s_i^\sharp)$. Finally, since $\pi(s_i^\sharp) = \pi(t_i)$ and $t_i \xrightarrow{*}_{\mathcal{R},\mu} \sigma(u_{i+1})$ for all $i \geq 1$, by Proposition 57 and Lemma 56, $\pi(t_i) \succeq \sigma_\pi(\pi(u_{i+1}))$. Therefore, again by transitivity of \succeq and compatibility of \succeq and \sqsupset , we conclude that $\sigma_\pi(\pi(u_i)) (\succeq \cup \sqsupset) \sigma_\pi(\pi(u_{i+1}))$.

Since $u \rightarrow v$ occurs infinitely often in B , there is an infinite set $\mathcal{I} \subseteq \mathbb{N}$ such that $\sigma_\pi(\pi(u_i)) \sqsupset \sigma_\pi(\pi(u_{i+1}))$ for all $i \in \mathcal{I}$. And we have $\sigma_\pi(\pi(u_i)) (\succeq \cup \sqsupset) \sigma_\pi(\pi(u_{i+1}))$ for all other $u_i \rightarrow v_i \in \mathcal{Q}$. Thus, by using the compatibility conditions of the μ_π -reduction pair, we obtain an infinite decreasing \sqsupset -sequence which contradicts well-foundedness of \sqsupset .

Therefore, $\mathcal{Q} \subseteq (\mathcal{P} \setminus \mathcal{P}_{\sqsupset})$, which means that B is an infinite minimal $(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathcal{R}, \mu)$ -chain, thus leading to a contradiction. \blacksquare

The following example shows that the ‘compatibility’ between $\mu_{\mathcal{H},\mathcal{R},\mu}$ and the argument filtering π which is required when collapsing pairs are present is necessary in Theorem 58.

Example 59

Consider the following TRS:

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

$$f(c(x)) \rightarrow x$$

together with the replacement map μ given by $\mu(f) = \mu(h) = \{1\}$ and $\mu(c) = \emptyset$. Then, $\text{DP}(\mathcal{R}, \mu)$ consists of a single (collapsing) CSDP:

$$F(c(x)) \rightarrow x$$

and $\mathcal{NHT}_{\text{DP}(\mathcal{R}, \mu)} = \{f(\mathbf{a}), \mathbf{a}\}$. Note that \mathcal{R} is *not* μ -terminating:

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

However, by using the argument filtering π given by $\pi(h) = []$, $\pi(F) = \pi(f) = [1]$ and $\pi(c) = 1$, we would get the constraints:

$$\begin{array}{rcl} \pi(\mathbf{a}) & = & \mathbf{a} \quad \succeq \quad \mathbf{h} = \pi(c(h(f(\mathbf{a}), \mathbf{b}))) \\ \pi(f(c(x))) & = & f(x) \quad \succeq \quad x = \pi(x) \\ \pi(F(c(x))) & = & F(x) \quad \sqsupset \quad x = \pi(x) \end{array}$$

which are easily satisfiable (by an RPO with precedence $\mathbf{a} \succ \mathbf{h}$, for instance). Thus, we would wrongly conclude μ -termination of \mathcal{R} . Note that $\pi(c) = 1$ but $\mu(c) = \emptyset$ and that $\pi(h) = []$ but $\mu(h) = \{1\}$. Note also that $\mu_\pi(f) = \mu_\pi(F) = \{1\}$ and $\mu_\pi(\mathbf{a}) = \mu_\pi(\mathbf{h}) = \emptyset$.

Example 60

Consider the TRS \mathcal{R} [Zan97, Example 5]:

$$\begin{array}{ll} \text{if}(\text{true}, x, y) \rightarrow x & f(x) \rightarrow \text{if}(x, c, f(\text{true})) \\ \text{if}(\text{false}, x, y) \rightarrow y & \end{array}$$

with $\mu(\text{if}) = \{1, 2\}$. Then, $\text{DP}(\mathcal{R}, \mu)$ consists of a CSDP in $\text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$ and another one in $\text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$:

$$F(x) \rightarrow \text{IF}(x, c, f(\text{true})) \qquad \text{IF}(\text{false}, x, y) \rightarrow y$$

with $\mu^\sharp(F) = \{1\}$ and $\mu(\text{IF}) = \{1, 2\}$. The μ -reduction pair $(\succeq, >)$ induced by the polynomial interpretation

$$\begin{array}{lll} [c] = [\text{true}] = 0 & [f](x) = x & [F](x) = x \\ [\text{false}] = 1 & [\text{if}](x, y, z) = x + y + z & [\text{IF}](x, y, z) = x + z \end{array}$$

can be used to prove the μ -termination of \mathcal{R} . Consider $\mathcal{P} = \text{DP}(\mathcal{R}, \mu)$. We have $\mathcal{NHT}_{\mathcal{P}} = \{f(\text{true})\}$. Since no symbol f hides any position i , we don't have to

impose that the quasi-ordering fullfills any subterm condition. Now we can see that the condition on the only hidden term in $\mathcal{NHT}_{\mathcal{P}}$ is also fulfilled:

$$[f(\text{true})] = 0 \geq 0 = [F(\text{true})]$$

Finally, for the three rules in \mathcal{R} and the two pairs in \mathcal{P} , we have:

$$\begin{array}{rclcl} [f(x)] & = & x & \geq & x & = & [f(x, c, f(\text{true}))] \\ [f(\text{true}, x, y)] & = & x + y & \geq & x & = & [x] \\ [f(\text{false}, x, y)] & = & x + y & \geq & y & = & [y] \\ [F(x)] & = & x & \geq & x & = & [F(x, c, f(\text{true}))] \\ [IF(\text{false}, x, y)] & = & y + 1 & > & y & = & [y] \end{array}$$

So, we remove the pair $IF(\text{false}, x, y) \rightarrow y$ from \mathcal{P} . With the remaining pair $F(x) \rightarrow IF(x, c, f(\text{true}))$ no infinite chain is possible. Thus, the μ -termination of \mathcal{R} is proved.

Our last result establishes that if we are able to provide a strict comparison between unmarked and marked versions of the (filtered) hidden terms in $\mathcal{NHT}_{\mathcal{P}}$, then we can remove *all* collapsing pairs at the same time.

Theorem 61 (μ -reduction pair processor for collapsing pairs) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs, $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. Let π be an argument filtering for $\mathcal{F} \cup \mathcal{G}$ and (\succsim, \sqsupset) be a μ_{π} -reduction pair such that*

1. $\pi(\mathcal{R}) \subseteq \succsim$, $\pi(\mathcal{P}) \subseteq \succsim \cup \sqsupset$, and
2. $\pi(t) \sqsupset \pi(t^{\#})$ for all $t \in \mathcal{NHT}_{\mathcal{P}}$ and
 - a) for all $f \in \mathcal{F}$, either $\pi(f) = [i_1, \dots, i_m]$ and $\mu_{\mathcal{H}, \mathcal{R}, \mu}(f) \subseteq \pi(f)$, or $\pi(f) = i$ and $\mu(f) = \{i\}$,
 - b) $\pi(f(x_1, \dots, x_i, \dots, x_n)) \succsim x_i$ for all $f \in \mathcal{F}$ and for all position $i \in \mu_{\mathcal{H}, \mathcal{R}, \mu}(f)$.

Then, the processor Proc_{RPc} given by

$$\text{Proc}_{RPc}(\mathcal{P}, \mathcal{R}, \mu) = \begin{cases} \{(\mathcal{P}_{\mathcal{G}}, \mathcal{R}, \mu)\} & \text{if (1) and (2) hold} \\ \{(\mathcal{P}, \mathcal{R}, \mu)\} & \text{otherwise} \end{cases}$$

is sound and complete.

Proof. As in the proof of Theorem 58, we proceed by contradiction. We assume that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain A , but that there is no infinite

minimal $(\mathcal{P}_G, \mathcal{R}, \mu)$ -chain. Thus, there is $\mathcal{Q} \subseteq \mathcal{P}$ such that $\mathcal{Q} \cap \mathcal{P}_X \neq \emptyset$ and A has a tail B as in the proof of Theorem 58. Now, we assume the notation as in the first paragraph of such a proof.

We have $\sigma_\pi(\pi(u_i)) (\succeq \cup \sqsupset) \pi(t_i)$ and $\pi(t_i) \succeq \sigma_\pi(\pi(u_{i+1}))$ for all pairs $u_i \rightarrow v_i \in \mathcal{P}_G$. If $u_i \rightarrow v_i = u_i \rightarrow x_i \in \mathcal{Q}_X$, then by applying the considerations in the corresponding item of the proof of Theorem 58 and taking into account that $\pi(t) \sqsupset \pi(t^\sharp)$ for all $t \in \mathcal{NHT}_{\mathcal{P}}$, we have now that $\sigma_\pi(\pi(u_i)) (\succeq \cup \sqsupset) \sigma_\pi(x_i) \sqsupset \pi(t_i) \succeq \sigma_\pi(\pi(u_{i+1}))$. Since pairs $u_i \rightarrow v_i \in \mathcal{Q}_X$ occur infinitely often in B , by using the compatibility conditions of the μ_π -reduction pair, we obtain an infinite decreasing \sqsupset -sequence which contradicts well-foundedness of \sqsupset . ■

6.3.3. Usable Rules for CSR

Until now, we are imposing that all the rules of the CS-termination problem must be included in \succeq to apply a μ -reduction pair processor, but it is desirable to consider only the necessary rules to capture all possible infinite sequence instead of all the rules in the CS-termination problem. *Usable rules* [AG00, HM04, TGSK04] allow us to obtain this 'minimal' set. To prove that this set of rules is enough to prove the absence of minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chains, we are going to develop one interpretation that allows us to simulate every infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain as a chain over the set of the usable rules. As in rewriting [HM04, TGSK04], we need \mathcal{C}_ε -compatibility to this task, i.e. $\mathbf{g}(x, y) \succeq x$ and $\mathbf{g}(x, y) \succeq y$ for a fresh symbol \mathbf{g} .

Usable rules were introduced by Arts and Giesl in [AG00] in connection with innermost termination. Hirokawa and Middeldorp [HM04] and (independently) Thiemann et al. [TGSK04] showed how to use them to prove termination. The difference between usable rules in rewriting and in context-sensitive rewriting are related directly with the associated replacement map. The replacement map function changes the normal path, forbidding certain reductions (on non-replacing positions). For that reason, the set of usable rules is different.

Definition 62 (Direct dependency [HM04]) *Given a TRS $\mathcal{R} = (\mathcal{F}, R)$, we say that $f \in \mathcal{F}$ directly depends on $g \in \mathcal{F}$, written $f \triangleright_d g$, if there is a rule $l \rightarrow r \in \mathcal{R}$ with*

- $f = \text{root}(l)$ and
- g occurs in r .

The set of defined function symbols in a term t is $\mathcal{DFun}(t) = \{f \mid \exists p \in \mathcal{Pos}(t), f = \text{root}(t|_p) \in \mathcal{D}\}$. Let \triangleright_d^* be the transitive and reflexive closure of \triangleright_d . Then, we have:

Definition 63 (Usable rules [HM04]) For a set Δ of symbols we denote by $\mathcal{R} \mid \Delta$ the set of rewriting rules $l \rightarrow r \in \mathcal{R}$ with $\text{root}(l) \in \Delta$. The set $\mathcal{U}(\mathcal{R}, t)$ of usable rules of a term t is defined as $\mathcal{R} \mid \{g \mid f \triangleright_d^* g \text{ for some } f \in \mathcal{DFun}(t)\}$. If \mathcal{P} is a set of pairs then

$$\mathcal{U}(\mathcal{R}, \mathcal{P}) = \bigcup_{l \rightarrow r \in \mathcal{P}} \mathcal{U}(\mathcal{R}, r)$$

The set $\mathcal{U}(\mathcal{R}, \mathcal{P})$ can be used instead of \mathcal{R} when looking for a reduction pair that proves termination of \mathcal{R} [HM04, TGSK04].

Unfortunately, this does not work with CSR.

Example 64

Consider the CS-TRS \mathcal{R} in Example 3. We have only one CS-termination problem:

$$(\{F(c(x)) \rightarrow x\}, \mathcal{R}, \mu)$$

where $\mathcal{U}(\mathcal{R}, \{F(c(x)) \rightarrow x\}) = \emptyset$ according to Definition 63. It is very easy to find a polynomial interpretation inducing a μ -reduction pair which is compatible with the pair of the CS-termination problem and thus, wrongly suggesting that the CS-termination problem is finite:

$$F(\underline{a}) \hookrightarrow_{\mathcal{R}, \mu} \underline{F(c(f(\underline{a})))} \hookrightarrow_{\{F(c(x)) \rightarrow x\}, \mu} f(\underline{a}) \triangleright_{\mu}^{\#} F(\underline{a}) \hookrightarrow_{\mathcal{R}, \mu} \dots$$

In the following, we discuss suitable notions of usable rules for CSR.

Basic usable rules for CSR

A first attempt to give a notion of usable rules in proofs of termination of (innermost) context-sensitive rewriting has been given in [AL07]. Although the results in [AL07] are not completely general, they show that the straightforward adaptation to CSR of the standard notion of usable rules (see Definition 66 below) applies to prove termination of *conservative* CS-termination problems.

The application of these rules are related with the appearance of new symbols. These symbols only appear in the right hand side of the DPs in the infinite chain. Rules headed by those symbols are usable.

With conservative CS-termination problems we can relax the previous dependency relation for some symbols thanks to the replacement map.

Then, the μ -dependency relation is:

Definition 65 (Basic μ -dependency) *Given a TRS $\mathcal{R} = (\mathcal{F}, R)$ and a replacement map $\mu \in M_{\mathcal{F}}$, we say that $f \in \mathcal{F}$ has a basic μ -dependency on $g \in \mathcal{F}$, written $f \blacktriangleright_{d,\mu} g$, if there is $l \rightarrow r \in R$ with*

- $f = \text{root}(l)$ and
- g occurs in r at a μ -replacing position.

This leads to a straightforward extension of Definition 63. The set of μ -replacing defined function symbols in a term t is $\mathcal{DFun}^{\mu}(t) = \{f \mid \exists p \in \mathcal{Pos}^{\mu}(t), f = \text{root}(t|_p) \in \mathcal{D}\}$. Then, we have

Definition 66 (Basic context-sensitive usable rules) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. The set $\mathcal{U}_B(\mathcal{R}, \mu, t)$ of basic context-sensitive usable rules of a term t is defined as $\mathcal{R} \mid \{g \mid f \blacktriangleright_{d,\mu}^* g \text{ for some } f \in \mathcal{DFun}^{\mu}(t)\}$, where $\blacktriangleright_{d,\mu}^*$ is the transitive and reflexive closure of $\blacktriangleright_{d,\mu}$. If $(\mathcal{P}, \mathcal{R}, \mu)$ is a CS-termination problem then:*

$$\mathcal{U}_B(\mathcal{R}, \mu, \mathcal{P}) = \bigcup_{l \rightarrow r \in \mathcal{P}} \mathcal{U}_B(\mathcal{R}, \mu, r)$$

Example 67

Consider the following CS-TRS \mathcal{R} :

$$f(\mathbf{a}, x, x) \rightarrow f(x, \mathbf{b}, \mathbf{b}) \quad (6.21) \qquad \mathbf{b} \rightarrow \mathbf{a} \quad (6.22)$$

together with $\mu(f) = \emptyset$. We have the following CS-termination problem:

$$(\{F(\mathbf{a}, x, x) \rightarrow F(x, \mathbf{b}, \mathbf{b})\}, \mathcal{R}, \mu)$$

Since \mathbf{b} is at frozen positions in the right-hand side of the pair, there is no usable rule following Definition 66.

However, in sharp contrast with [AL07], Definition 65 does not lead to a correct approach for proving termination of CSR, even for conservative TRSs.

Example 68

Consider the following TRS \mathcal{R} [AL07]:

$$f(c(x), x) \rightarrow f(x, x) \quad (6.23) \qquad \mathbf{b} \rightarrow c(\mathbf{b}) \quad (6.24)$$

together with $\mu(f) = \{1, 2\}$ and $\mu(c) = \emptyset$. Note that \mathcal{R} is μ -conservative. We have the following CS-termination problem:

$$(\{F(c(x), x) \rightarrow F(x, x)\}, \mathcal{R}, \mu)$$

where $\mathcal{U}(\mathcal{R}, \mu, \{F(c(x), x) \rightarrow F(x, x)\}) = \emptyset$ according to Definition 66. We can find a polynomial interpretation inducing a μ -reduction pair which is compatible with the pair of the CS-termination problem and thus, wrongly suggesting that the CS-termination problem is finite:

$$\underline{F(c(\mathbf{b}), \mathbf{b})} \hookrightarrow_{\{F(c(x), x) \rightarrow F(x, x)\}, \mu} \underline{F(\mathbf{b}, \mathbf{b})} \hookrightarrow_{\mathcal{R}, \mu} \underline{F(c(\mathbf{b}), \mathbf{b})} \hookrightarrow_{\{F(c(x), x) \rightarrow F(x, x)\}, \mu} \dots$$

Termination with basic usable rules

According to the discussion in the previous subsection, the notion of basic usable rule is not correct even for conservative systems. Still, since $\mathcal{U}_B(\mathcal{R}, \mu, \mathcal{P})$ is contained (and is usually smaller than) $\mathcal{U}(\mathcal{R}, \mathcal{P})$, it is interesting to identify a class of CS-TRSs where basic usable rules can be safely used. Then, we consider a more restrictive kind of conservative CS-TRSs: the *strongly conservative* CS-TRSs, in which the problem illustrated by Example 68 is not possible.

Definition 69 (Strongly conservative [GLU08]) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{F}}$. A rule $l \rightarrow r \in R$ is strongly conservative if it is conservative and $\mathcal{V}ar^{\mu}(l) \cap \mathcal{V}ar^{\sharp}(l) = \mathcal{V}ar^{\mu}(r) \cap \mathcal{V}ar^{\sharp}(r) = \emptyset$.*

Linear CS-TRSs trivially satisfy $\mathcal{V}ar^{\mu}(l) \cap \mathcal{V}ar^{\sharp}(l) = \mathcal{V}ar^{\mu}(r) \cap \mathcal{V}ar^{\sharp}(r) = \emptyset$. Hence, linear conservative CS-TRSs are strongly conservative. For instance, Example 67 is strongly conservative, but Example 68 is not.

Theorem 79 below shows that basic usable rules in Definition 66 can be used to improve proofs of termination of CSR for strongly conservative CS-TRSs. In

[GTSK04, HM04], an *interpretation* of terms as sequences of their possible reducts is used¹. The definition of the transformation requires adding new fresh (list constructor) symbols $\perp, \mathbf{g} \notin \mathcal{F}$ and the (projection) rules $\mathcal{C}_\varepsilon = \{\mathbf{g}(x, y) \rightarrow x, \mathbf{g}(x, y) \rightarrow y\}$. In this way, infinite minimal $(\mathcal{P}, \mathcal{R})$ -chains can be represented as infinite $(\mathcal{P}, \mathcal{U}(\mathcal{R}, \mathcal{P}) \cup \mathcal{C}_\varepsilon)$ -chains. We recall here the interpretation definition.

Definition 70 (Interpretation [GTSK04, HM04]) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\Delta \subseteq \mathcal{F}$. Let $>$ be an arbitrary total ordering over $\mathcal{T}(\mathcal{F} \cup \{\perp, \mathbf{g}\}, \mathcal{X})$ where \perp is a new constant symbol and \mathbf{g} is a new binary symbol. The interpretation I_Δ is a mapping from terminating terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$ to terms in $\mathcal{T}(\mathcal{F} \cup \{\perp, \mathbf{g}\}, \mathcal{X})$ defined as follows:*

$$I_\Delta(t) = \begin{cases} t & \text{if } t \in \mathcal{X} \\ f(I_\Delta(t_1), \dots, I_\Delta(t_n)) & \text{if } t = f(t_1 \dots t_n) \text{ and } f \notin \Delta \\ \mathbf{g}(f(I_\Delta(t_1), \dots, I_\Delta(t_n)), t') & \text{if } t = f(t_1 \dots t_n) \text{ and } f \in \Delta \end{cases}$$

$$\begin{aligned} \text{where } t' &= \text{order}(\{I_\Delta(u) \mid t \rightarrow_{\mathcal{R}} u\}) \\ \text{order}(T) &= \begin{cases} \perp, & \text{if } T = \emptyset \\ \mathbf{g}(t, \text{order}(T \setminus \{t\})) & \text{if } t \text{ is minimal in } T \text{ w.r.t. } > \end{cases} \end{aligned}$$

The set of symbols $\Delta \subseteq \mathcal{F}$ in Definition 70 is intended to represent the set of ‘non-usable symbols’, i.e., symbols which do not occur in the usable rules of the considered set of pairs \mathcal{P} .

We provide an interpretation of terms akin to Hirokawa and Middeldorp’s [HM04] with the difference that we treat $\mathcal{M}_{\infty, \mu}$. Hence, we pay special attention to non- μ -replacing positions where possibly infinite μ -rewrite sequence might occur.

Definition 71 (Basic μ -interpretation) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$ and $\Delta \subseteq \mathcal{F}$. Let $>$ be an arbitrary total ordering over $\mathcal{T}(\mathcal{F} \cup \{\perp, \mathbf{g}\}, \mathcal{X})$ where \perp is a new constant symbol and \mathbf{g} is a new binary symbol. The interpretation $I'_{\Delta, \mu}$ is a mapping from μ -terminating terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$ into $\mathcal{T}(\mathcal{F} \cup \{\perp, \mathbf{g}\}, \mathcal{X})$ defined as follows:*

$$I'_{\Delta, \mu}(t) = \begin{cases} t & \text{if } t \in \mathcal{X} \\ f(I'_{\Delta, \mu, f, 1}(t_1), \dots, I'_{\Delta, \mu, f, n}(t_n)) & \text{if } t = f(t_1 \dots t_n) \text{ and } f \notin \Delta \\ \mathbf{g}(f(I'_{\Delta, \mu, f, 1}(t_1), \dots, I'_{\Delta, \mu, f, n}(t_n)), t') & \text{if } t = f(t_1 \dots t_n) \text{ and } f \in \Delta \end{cases}$$

¹This method goes back to [Gra94].

$$\begin{aligned}
\text{where } I'_{\Delta,\mu,f,i}(t) &= \begin{cases} I'_{\Delta,\mu}(t) & \text{if } i \in \mu(f) \\ t & \text{if } i \notin \mu(f) \end{cases} \\
t' &= \text{order} \left(\{I'_{\Delta,\mu}(u) \mid t \hookrightarrow_{\mathcal{R},\mu} u\} \right) \\
\text{order}(T) &= \begin{cases} \perp, & \text{if } T = \emptyset \\ \mathbf{g}(t, \text{order}(T \setminus \{t\})) & \text{if } t \text{ is minimal in } T \text{ w.r.t. } > \end{cases}
\end{aligned}$$

In contrast to [Gra94, HM04, Urb04], we do not interpret non- μ -replacing positions since we deal only with strongly conservative CS-TRSs.

The interpretation of a term $t = f(t_1, \dots, t_n)$, where $f \in \Delta$, is a sequence of its interpreted one-step- μ -reducts. It is possible to reach any of them by using a suitable $\hookrightarrow_{\mathcal{C}_\varepsilon^2, \mu}^* \circ \hookrightarrow_{\mathcal{C}_\varepsilon^1, \mu}$ -sequence where $\mathcal{C}_\varepsilon^1 = \{\mathbf{g}(x, y) \rightarrow x\}$ and $\mathcal{C}_\varepsilon^2 = \{\mathbf{g}(x, y) \rightarrow y\}$. It is easy to prove that the basic μ -interpretation is well-defined (finite) for all μ -terminating terms.

Lemma 72 (Well-definition of $I'_{\Delta,\mu}$) *For each μ -terminating term t , $I'_{\Delta,\mu}(t)$ is finite.*

Proof. By well-founded induction based on the fact that μ -replacing subterms are μ -terminating. The interpretation of frozen positions is always finite (they are not developed). Interpretation of active positions is always finite because there is no infinite μ -reduction for these subterms. Hence $I'_{\Delta,\mu}(t)$ is finite. \blacksquare

For the proof of our next theorem, we need some auxiliary definitions and results.

Definition 73 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, σ be a substitution and $\Delta \in \mathcal{F}$. We denote by $\sigma_{I'_{\Delta,\mu}} : \mathcal{T}(\mathcal{F}, \mathcal{X}) \rightarrow \mathcal{T}(\mathcal{F}, \mathcal{X})$ a function that, given a term t replaces occurrences of $x \in \mathcal{V}ar(t)$ at position p in t by either $I'_{\Delta,\mu}(\sigma(x))$ if $p \in \mathcal{P}os^\mu(t)$, or $\sigma(x)$ if $p \notin \mathcal{P}os^\mu(t)$.*

The following technical proposition is obvious from Definition 73.

Proposition 74 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, σ be a substitution and $\Delta \subseteq \mathcal{F}$. Let t be a term such that $\mathcal{V}ar^\mu(t) \cap \mathcal{V}ar^\mu(t) = \emptyset$. Let $\bar{\sigma}_{I'_{\Delta,\mu}, t}$ be a substitution given by*

$$\bar{\sigma}_{I'_{\Delta,\mu}, t}(x) = \begin{cases} I'_{\Delta,\mu}(\sigma(x)) & \text{if } x \in \mathcal{V}ar^\mu(t) \\ \sigma(x) & \text{otherwise} \end{cases}$$

Then, $\bar{\sigma}_{I'_{\Delta,\mu}, t}(t) = \sigma_{I_{\Delta,\mu}}(t)$.

Lemma 75 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$ and $\Delta \subseteq \mathcal{F}$. Let t be a term and σ be a substitution. If $\sigma(t)$ is terminating, then $I'_{\Delta, \mu}(\sigma(t)) \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}}^* \sigma_{I'_{\Delta, \mu}}(t)$. If t does not contain Δ -symbols, then $I'_{\Delta, \mu}(\sigma(t)) = \sigma_{I'_{\Delta, \mu}}(t)$.*

Proof. By structural induction on t :

- If t is a variable then $I'_{\Delta, \mu}(\sigma(t)) = \sigma_{I'_{\Delta, \mu}}(t)$.
- If $t = f(t_1, \dots, t_n)$ then
 - If $f \notin \Delta$ then $I'_{\Delta, \mu}(\sigma(t)) = f(I'_{\Delta, \mu, f, 1}(\sigma(t_1)), \dots, I'_{\Delta, \mu, f, n}(\sigma(t_n)))$. Terms $\sigma(t_i)$ are μ -terminating for $i \in \mu(f)$. By induction hypothesis, for all terms t_i s.t. $i \in \mu(f)$, we have $I'_{\Delta, \mu, f, i}(\sigma(t_i)) = I'_{\Delta, \mu}(\sigma(t_i)) \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}}^* \sigma_{I'_{\Delta, \mu}}(t_i)$. And for all t_i s.t. $i \notin \mu(f)$, we have $I'_{\Delta, \mu, f, i}(\sigma(t_i)) = \sigma(t_i)$. By definition of $\sigma_{I'_{\Delta, \mu}}$ we have $\sigma_{I'_{\Delta, \mu}}(f(t_1, \dots, t_n)) = f(t'_1, \dots, t'_n)$, where $t'_i = I'_{\Delta, \mu}(t_i)$ if $i \in \mu(f)$ and $t'_i = \sigma(t_i)$ if $i \notin \mu(f)$. Hence, $I'_{\Delta, \mu}(\sigma(t)) \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}}^* \sigma_{I'_{\Delta, \mu}}(t)$.
 - If $f \in \Delta$, $I'_{\Delta, \mu}(\sigma(t)) = \mathbf{g}(f(I'_{\Delta, \mu, f, 1}(\sigma(t_1)), \dots, I'_{\Delta, \mu, f, n}(\sigma(t_n))), t')$ for some t' . Using one step of $\mathcal{C}_{\varepsilon}$ ($\hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}} f(I'_{\Delta, \mu, f, 1}(\sigma(t_1)), \dots, I'_{\Delta, \mu, f, n}(\sigma(t_n)))$) and the preceding result we get $I'_{\Delta, \mu}(\sigma(t)) \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}}^* \sigma_{I'_{\Delta, \mu}}(t)$.

Then we conclude $I'_{\Delta, \mu}(\sigma(t)) \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}}^* \sigma_{I'_{\Delta, \mu}}(t)$. The second part of the lemma is easily proved by structural induction and using Definition 71. ■

Lemma 76 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, $\Delta \subseteq \mathcal{F}$ and $C[\]$ a context with n μ -replacing holes. If $t = C[t_1, \dots, t_n]$ is μ -terminating and the context $C[\]$ contains no Δ -symbols then $I'_{\Delta, \mu}(C[t_1, \dots, t_n]) = C[I'_{\Delta, \mu}(t_1), \dots, I'_{\Delta, \mu}(t_n)]$.*

Proof. By structural induction on t :

- If $C[\]$ has no holes then $I'_{\Delta, \mu}(C[\]) = C[\]$.
- If the context $C[\] = f(s_1, \dots, s_k)$ then $f \notin \Delta$, where $1 \leq k \leq ar(f)$. We have that $C[\] = C''[s_j]$ for a position $1 \leq j \leq k$ and $s_j = C'''[t_1, \dots, t_n]$:
 - If $j \notin \mu(f)$ then the holes are non- μ -replacing, contradicting the hypothesis.
 - If $j \in \mu(f)$ then by Definition 71 we have that $I'_{\Delta, \mu}(C''[s_j]) = C''[I'_{\Delta, \mu}(s_j)]$. Now, applying the induction hypothesis, $I'_{\Delta, \mu}(s_j) = I'_{\Delta, \mu}(C'''[t_1, \dots, t_n]) = C'''[I'_{\Delta, \mu}(t_1), \dots, I'_{\Delta, \mu}(t_n)]$.

And we can conclude that $I'_{\Delta,\mu}(C[t_1, \dots, t_n]) = C[I'_{\Delta,\mu}(t_1), \dots, I'_{\Delta,\mu}(t_n)]$. ■

The interpretation of a term $t = f(t_1, \dots, t_n)$, where $f \in \Delta$, is a sequence of its interpreted one-step- μ -reducts. It is possible to reach any of them by using a suitable \hookrightarrow_{μ} -sequence. In particular, we have the following result which is easily proved by structural induction:

Proposition 77 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$ and $\Delta \subseteq \mathcal{F}$. For all terms $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, $I'_{\Delta,\mu}(t) \hookrightarrow_{\mathcal{C}_{\varepsilon,\mu}}^* t$.*

Proof. By structural induction. ■

Lemma 78 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$ and $l \rightarrow r \in R$ is strongly conservative. Let $\Delta \subseteq \mathcal{F}$ such that if $\text{root}(l) \notin \Delta$ then $\mathcal{D}\text{Fun}^{\mu}(r) \notin \Delta$. If terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ are μ -terminating with respect to (\mathcal{R}, μ) and $s \hookrightarrow_{\{l \rightarrow r\}, \mu} t$, then $I'_{\Delta,\mu}(s) \hookrightarrow_{\{\{l \rightarrow r\} \cup \mathcal{C}_{\varepsilon,\mu}\}}^+ I'_{\Delta,\mu}(t)$ if $\text{root}(l) \notin \Delta$ or $I'_{\Delta,\mu}(s) \hookrightarrow_{\mathcal{C}_{\varepsilon,\mu}}^+ I'_{\Delta,\mu}(t)$ otherwise.*

Proof. Let $p \in \text{Pos}^{\mu}(s)$ be the position of the rewrite step $s \hookrightarrow_{\mathcal{R}, \mu} t$. There are two cases:

- If there is a function symbol from Δ at a position above or in p , we can write $s = C[s_1, \dots, s_i, \dots, s_n]$ and $t = C[s_1, \dots, t_i, \dots, s_n]$, with $s_i \hookrightarrow_{\mathcal{R}, \mu} t_i$, where s_i and t_i are at a μ -replacing hole, $\text{root}(s_i) \in \Delta$ and the context contains no Δ -symbol. We have $I'_{\Delta,\mu}(s_i) \hookrightarrow_{\mathcal{C}_{\varepsilon,\mu}} \text{order}(\bigcup_{s_i \hookrightarrow_{\mathcal{R}, \mu} u} I'_{\Delta,\mu}(u))$. Since $s_i \hookrightarrow_{\mathcal{R}, \mu} t_i$, applying appropriate $\mathcal{C}_{\varepsilon}$ -steps we extract $I'_{\Delta,\mu}(t_i)$ from the term $\text{order}(\bigcup_{s_i \hookrightarrow_{\mathcal{R}, \mu} u} I'_{\Delta,\mu}(u))$, so $I'_{\Delta,\mu}(s_i) \hookrightarrow_{\mathcal{C}_{\varepsilon,\mu}}^+ I'_{\Delta,\mu}(t_i)$. By Lemma 76 we get $I'_{\Delta,\mu}(s) \hookrightarrow_{\mathcal{C}_{\varepsilon,\mu}}^+ I'_{\Delta,\mu}(t)$.
- Otherwise, we can write $s = C[s_1, \dots, s_i, \dots, s_n]$ and $t = C[s_1, \dots, t_i, \dots, s_n]$ with $s_i \xrightarrow{\Delta}_{\mathcal{R}, \mu} t_i$, where $\text{root}(s_i) \notin \Delta$ and the context $C[\]$ contains no Δ -symbols. Since $\text{root}(s_i) \notin \Delta$ let σ be the substitution with $\text{Dom}(\sigma) \subseteq \text{Var}(l)$ s.t. $s_i = \sigma(l)$ and $t_i = \sigma(r)$. By lemma 75 we have $I'_{\Delta,\mu}(s_i) = I'_{\Delta,\mu}(\sigma(l)) \hookrightarrow_{\mathcal{C}_{\varepsilon,\mu}}^* \sigma_{I'_{\Delta,\mu}}(l)$. Since right-hand side of r do not contain Δ -symbols in μ -replacing positions, the same lemma yields $I'_{\Delta,\mu}(t_i) = \sigma_{I'_{\Delta,\mu}}(r)$. By strong conservativeness of $l \rightarrow r$ (in particular since $\text{Var}^{\mu}(r) \cap \text{Var}^{\neq \mu}(r) = \emptyset$), by Proposition 74, $\bar{\sigma}_{I'_{\Delta,\mu}, r}$ is well-defined and $\bar{\sigma}_{I'_{\Delta,\mu}, r}(r) = \sigma_{I'_{\Delta,\mu}}(r)$. Let $x \in \text{Var}(l)$, we have two cases:

1. If $x \in \mathcal{V}ar^\mu(r)$, then $\bar{\sigma}_{I'_{\Delta,\mu},r}(x) = \sigma_{I'_{\Delta,\mu}}(x)$, and by conservativeness of $l \rightarrow r$ (in particular because $\mathcal{V}ar^\mu(r) \subseteq \mathcal{V}ar^\mu(l)$), we get $x \in \mathcal{V}ar^\mu(l)$.
2. If $x \notin \mathcal{V}ar^\mu(r)$, then $\bar{\sigma}_{I'_{\Delta,\mu},r}(x) = \sigma(x)$.

By Proposition 77, $\sigma_{I'_{\Delta,\mu}}(x) = I'_{\Delta,\mu}(\sigma(x)) \hookrightarrow_{\mathcal{C}_\varepsilon,\mu}^* \sigma(x)$. According to 1 and 2 above, by structural induction on l we easily get $\sigma_{I'_{\Delta,\mu}}(l) \hookrightarrow_{\mathcal{C}_\varepsilon,\mu}^* \bar{\sigma}_{I'_{\Delta,\mu},r}(l)$. Here, the fact that $\mathcal{V}ar^\mu(l) \cap \mathcal{V}ar^\mu(r) = \emptyset$ (which is part of the definition of strong conservativeness) ensures that a non- μ -replacing variable x (at position q) in l is not instantiated to $\sigma_{I'_{\Delta,\mu}}(x)$ by $\bar{\sigma}_{I'_{\Delta,\mu},r}$, in which case we could not rewrite $\sigma_{I'_{\Delta,\mu}}(l)|_q = \sigma(x)$ into $\bar{\sigma}_{I'_{\Delta,\mu},r}(l)|_q = I'_{\Delta,\mu}(\sigma(x))$. Since $\bar{\sigma}_{I'_{\Delta,\mu},r}(l) \hookrightarrow_{\{l \rightarrow r\},\mu} \bar{\sigma}_{I'_{\Delta,\mu},r}(r) = \sigma_{I'_{\Delta,\mu}}(r)$, we obtain $I'_{\Delta,\mu}(s_i) \hookrightarrow_{\{l \rightarrow r\} \cup \mathcal{C}_\varepsilon,\mu}^+ I'_{\Delta,\mu}(t_i)$. By Lemma 76 we conclude that $I'_{\Delta,\mu}(s) \hookrightarrow_{\{l \rightarrow r\} \cup \mathcal{C}_\varepsilon,\mu}^+ I'_{\Delta,\mu}(t)$. ■

Theorem 79 (Basic usable rules processor) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. If $P \cup \mathcal{U}_B(\mathcal{R}, \mu, \mathcal{P})$ is strongly conservative and there exists a μ -reduction pair (\succsim, \sqsupset) such that*

$$(1) \mathcal{U}_B(\mathcal{R}, \mu, \mathcal{P}) \cup \mathcal{C}_\varepsilon \subseteq \succsim \text{ and } \mathcal{P} \subseteq \succsim \cup \sqsupset.$$

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

$$\text{Proc}_{RPbur}(\mathcal{P}, \mathcal{R}, \mu) = \begin{cases} \{(\mathcal{P} \setminus \mathcal{P}_\sqsupset, \mathcal{R}, \mu)\} & \text{if (1) holds} \\ \{(\mathcal{P}, \mathcal{R}, \mu)\} & \text{otherwise} \end{cases}$$

is sound and complete.

Proof. We have to prove that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain if there is an infinite minimal $(\mathcal{P} \setminus \mathcal{P}_\sqsupset, \mathcal{R}, \mu)$ -chain. The *if* part is obvious. For the *only if* part, we proceed by contradiction. Assume that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain A , but that there is no infinite minimal $(\mathcal{P} \setminus \mathcal{P}_\sqsupset, \mathcal{R}, \mu)$ -chain. Due to the finiteness of \mathcal{P} , we can assume that there is $\mathcal{Q} \subseteq \mathcal{P}$ such that A has a tail B

$$\sigma(u_1) \hookrightarrow_{\mathcal{Q},\mu} \sigma(v_1) \hookrightarrow_{\mathcal{R},\mu}^* \sigma(u_2) \hookrightarrow_{\mathcal{Q},\mu} \sigma(v_2) \hookrightarrow_{\mathcal{R},\mu}^* \sigma(u_3) \hookrightarrow_{\mathcal{Q},\mu} \dots$$

for some substitution σ , where all pairs in \mathcal{Q} are infinitely often used. No pair $u \rightarrow v \in \mathcal{Q}$ satisfies that $u \sqsupset v$. Otherwise, we get a contradiction.

Let Δ be the set of defined symbols of $\mathcal{R} \setminus \mathcal{U}_B(\mathcal{R}, \mu, \mathcal{P})$. We show that after applying the basic μ -interpretation $I'_{\Delta,\mu}$ we obtain an infinite $(\mathcal{Q}, \mathcal{U}_B(\mathcal{R}, \mu, \mathcal{P}) \cup \mathcal{C}_\varepsilon, \mu)$ -chain. All terms in the infinite $(\mathcal{Q}, \mathcal{R}, \mu)$ -chain are μ -terminating with respect to (\mathcal{R}, μ) and hence we can indeed apply the basic μ -interpretation $I'_{\Delta,\mu}$. Let $i \geq 1$.

- First consider the dependency pair step. There is a context-sensitive dependency pair $u_i \rightarrow v_i \in \mathcal{Q}$ and a substitution σ such that $\sigma(u_i) \rightarrow_{\mathcal{Q}} \sigma(v_i)$. We may assume that $\text{Dom}(\sigma) \subseteq \text{Var}(u_i)$. $\sigma(x)$ is μ -terminating for every variable $x \in \text{Var}^\mu(u_i)$. Using Lemma 75 we have $I'_{\Delta,\mu}(\sigma(u_i)) \hookrightarrow_{\mathcal{C}_\varepsilon,\mu}^* \sigma_{I'_{\Delta,\mu}}(u_i)$. Since right-hand sides of dependency pairs in \mathcal{Q} lack Δ -symbols, the same lemma also yields $I'_{\Delta,\mu}(\sigma(v_i)) = \sigma_{I'_{\Delta,\mu}}(v_i)$. By strong conservativeness of the rule $u_i \rightarrow v_i$ (in particular by using the fact that $\text{Var}^\mu(v_i) \cap \text{Var}^\mu(u_i) = \emptyset$) and by Proposition 74 we know that $\bar{\sigma}_{I'_{\Delta,\mu},v_i}$ is a well-defined substitution and that $\bar{\sigma}_{I'_{\Delta,\mu},v_i}(v_i) = \sigma_{I'_{\Delta,\mu}}(v_i)$. As in the proof of Lemma 78, according to Item 1 and Item 2, by structural induction on u_i we easily get $\sigma_{I'_{\Delta,\mu}}(u_i) \hookrightarrow_{\mathcal{C}_\varepsilon,\mu}^* \bar{\sigma}_{I'_{\Delta,\mu},v_i}(u_i)$. Here, we need to use the fact that $\text{Var}^\mu(u_i) \cap \text{Var}^\mu(u_i) = \emptyset$ (which is also part of the definition of strong conservativeness) to ensure that a non- μ -replacing variable x (at position q) in u_i is not instantiated to $\sigma_{I'_{\Delta,\mu}}(x)$ by the substitution $\bar{\sigma}_{I'_{\Delta,\mu},v_i}$, in which case we would not be able to rewrite $\sigma_{I'_{\Delta,\mu}}(u_i)|_q = \sigma(x)$ into $\bar{\sigma}_{I'_{\Delta,\mu},v_i}(u_i)|_q = I'_{\Delta,\mu}(\sigma(x))$. Since $\bar{\sigma}_{I'_{\Delta,\mu},v_i}(u_i) \hookrightarrow_{\mathcal{U}_B(\mathcal{R},\mu,\mathcal{P}),\mu}^* \bar{\sigma}_{I'_{\Delta,\mu},v_i}(v_i) = \sigma_{I'_{\Delta,\mu}}(v_i)$, we conclude:

$$\begin{array}{ccccc} I'_{\Delta,\mu}(\sigma(u_i)) & \hookrightarrow_{\mathcal{C}_\varepsilon,\mu}^* & \sigma_{I'_{\Delta,\mu}}(u_i) & \hookrightarrow_{\mathcal{C}_\varepsilon,\mu}^* & \bar{\sigma}_{I'_{\Delta,\mu},v_i}(u_i) \\ \rightarrow_{\mathcal{Q}} & & & & \\ \bar{\sigma}_{I'_{\Delta,\mu},v_i}(v_i) & = & \sigma_{I'_{\Delta,\mu}}(v_i) & = & I'_{\Delta,\mu}(\sigma(v_i)) \end{array}$$

- Next consider the rewrite sequence $\sigma(v_i) \hookrightarrow_{\mathcal{R},\mu}^* \sigma(u_{i+1})$. All terms in it are μ -terminating, then we get $I'_{\Delta,\mu}(\sigma(v_i)) \hookrightarrow_{\mathcal{U}_B(\mathcal{R},\mu,\mathcal{P}) \cup \mathcal{C}_\varepsilon,\mu}^* I'_{\Delta,\mu}(\sigma(u_{i+1}))$ by repeatedly applying Lemma 78.

So we obtain the infinite μ -rewrite sequence:

$$\sigma(u_1) \hookrightarrow_{\mathcal{Q},\mu} \sigma(v_1) \hookrightarrow_{\mathcal{U}_B(\mathcal{R},\mu,\mathcal{P}) \cup \mathcal{C}_\varepsilon,\mu}^* \sigma(u_2) \hookrightarrow_{\mathcal{Q},\mu} \sigma(v_2) \hookrightarrow_{\mathcal{U}_B(\mathcal{R},\mu,\mathcal{P}) \cup \mathcal{C}_\varepsilon,\mu}^* \dots$$

Using the premise of the theorem, it is transformed into an infinite sequence consisting of \succsim and infinitely many \sqsupset steps. Using the stability condition, this contradicts the well-foundedness of \sqsupset . Therefore, $\mathcal{Q} \subseteq (\mathcal{P} \setminus \mathcal{P}_{\sqsupset})$, which means that B is an infinite $(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathcal{R}, \mu)$ -chain, thus leading a contradiction. ■

Example 80

Consider the CS-TRS in Example 20. The CS-termination problems after applying the SCC processor are:

$$(\{\text{SEL}(s(n), \text{cons}(x, y)) \rightarrow \text{SEL}(n, y)\}, \mathcal{R}, \mu) \quad (6.25)$$

$$(\{\text{MINUS}(s(x), s(y)) \rightarrow \text{MINUS}(x, y)\}, \mathcal{R}, \mu) \quad (6.26)$$

$$(\{\text{QUOT}(s(x), s(y)) \rightarrow \text{QUOT}(\text{minus}(x, y), s(y))\}, \mathcal{R}, \mu) \quad (6.27)$$

Whereas the CS-termination problems (6.25) and (6.26) are easily shown harmless by using the subterm criterion (see 6.4), this is not possible with the CS-termination problem (6.27). Since this pair is conservative and left-linear, it is strongly conservative. Furthermore, the set $\mathcal{U}_B(\mathcal{R}, \mu, \{\text{QUOT}(s(x), s(y)) \rightarrow \text{QUOT}(\text{minus}(x, y), s(y))\})$ of basic usable rules for CS-termination problem (6.27) contains the following rules $\{\text{minus}(x, 0) \rightarrow 0, \text{minus}(s(x), s(y)) \rightarrow \text{minus}(x, y)\}$ which are strongly conservative as well. The following polynomial interpretation proves the absence of infinite $(\text{C3}, \mathcal{R}, \mu)$ -chains, and hence CS-termination problem (6.27) is finite:

$$\begin{array}{ll} [0] & = 0 & [\text{minus}](x, y) & = 0 \\ [s(x)] & = 1 & [\text{QUOT}](x, y) & = x \end{array}$$

Basic usable rules with collapsing pairs

Collapsing dependency pairs are strongly conservative by definition, but in an infinite minimal chain they are treated in a different way. Then, we can use an extended notion of basic usable rule if these pairs fulfill some properties.

The idea behind is that if all the terms in the set of $t \in \mathcal{NHT}_{\mathcal{P}}$, $\text{Var}^{\mu}(t) = \emptyset$, then there is no migrating variable in the application of the collapsing pair and hence, we can maintain as usable only those rules of symbols appeared in μ -replacing positions in the chain. Then, to capture the usable rules of a collapsing pairs is equivalent to capture the usable rules of the following set.

Definition 81 (marking rules set) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. We define the set $\mathcal{P}_{\mathcal{NHT}}$ as:*

$$\mathcal{P}_{\mathcal{NHT}} = \{t \rightarrow t^{\sharp} \mid t \in \mathcal{NHT}_{\mathcal{P}}\}$$

Definition 82 (Extended basic CS-usable rules) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. The set $\mathcal{U}_{eB}(\mathcal{R}, \mu, t)$ of extended basic context-sensitive usable rules of a term t is defined as $\mathcal{R} \mid \{g \mid f \blacktriangleright_{d, \mu}^* g \text{ for some } f \in$*

$\mathcal{DFun}^\mu(t)$, where $\blacktriangleright_{d,\mu}^*$ is the transitive and reflexive closure of $\blacktriangleright_{d,\mu}$. If $(\mathcal{P}, \mathcal{R}, \mu)$ is a CS-termination problem and $\mathcal{P}_\mathcal{X} \neq \emptyset$ then:

$$\mathcal{U}_{eB}(\mathcal{R}, \mu, \mathcal{P}) = \bigcup_{l \rightarrow r \in \mathcal{P}_G \cup \mathcal{P}_{\mathcal{NHT}}} \mathcal{U}_{eB}(\mathcal{R}, \mu, r)$$

Our processor is extended in the same way:

Theorem 83 (Extended basic usable rules processor) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. If $\mathcal{P}_G \cup \mathcal{P}_{\mathcal{NHT}} \cup \mathcal{U}_{eB}(\mathcal{R}, \mu, \mathcal{P})$ is strongly conservative and there exists a μ -reduction pair (\succsim, \sqsupset) such that*

$$(1) \mathcal{U}_{eB}(\mathcal{R}, \mu, \mathcal{P}) \cup \mathcal{C}_\varepsilon \subseteq \succsim \text{ and } \mathcal{P} \subseteq \succsim \cup \sqsupset,$$

(2) whenever $\mathcal{NHT}_\mathcal{P} \neq \emptyset$ and $\mathcal{P}_\mathcal{X} \neq \emptyset$, we have that

$$(a) f(x_1, \dots, x_i, \dots, x_n) \succsim x_i \text{ for all } f \in \mathcal{F} \text{ and for all position } i \in \mu_{\mathcal{H}, \mathcal{R}, \mu}(f),$$

and

$$(b) t (\succsim \cup \sqsupset) t^\# \text{ for all } t \in \mathcal{NHT}_\mathcal{P}.$$

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

$$\text{Proc}_{RPebur}(\mathcal{P}, \mathcal{R}, \mu) = \begin{cases} \{(\mathcal{P} \setminus \mathcal{P}_\sqsupset, \mathcal{R}, \mu)\} & \text{if (1) and (2) hold} \\ \{(\mathcal{P}, \mathcal{R}, \mu)\} & \text{otherwise} \end{cases}$$

is sound and complete.

Proof. We have to prove that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain if and only if there is an infinite minimal $(\mathcal{P} \setminus \mathcal{P}_\sqsupset, \mathcal{R}, \mu)$ -chain. The *if* part is obvious. For the *only if* part, we proceed by contradiction. Assume that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain A , but that there is no infinite minimal $(\mathcal{P} \setminus \mathcal{P}_\sqsupset, \mathcal{R}, \mu)$ -chain. Due to the finiteness of \mathcal{P} , we can assume that there is $\mathcal{Q} \subseteq \mathcal{P}$ such that A has a tail B

$$\sigma(u_1) \hookrightarrow_{\mathcal{Q}, \mu} \circ \triangleright_\mu^\# t_1 \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_2) \hookrightarrow_{\mathcal{Q}, \mu} \circ \triangleright_\mu^\# t_2 \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_3) \hookrightarrow_{\mathcal{Q}, \mu} \circ \triangleright_\mu^\# \dots$$

for some substitution σ , where all pairs in \mathcal{Q} are infinitely often used, and, for all $i \geq 1$, (1) if $u_i \rightarrow v_i \in \mathcal{Q}_G$, then $t_i = \sigma(v_i)$ and (2) if $u_i \rightarrow v_i = u_i \rightarrow x_i \in \mathcal{Q}_\mathcal{X}$, then $t_i = s_i^\#$ for some s_i and some context C_i with a μ -replacing hole such that $\sigma(x_i) = C_i[s_i]$ and $C_i[s_i] = \theta_i(\bar{C}_i[\bar{s}_i])$ for some $\bar{s}_i \in \mathcal{NHT}$, hiding context $\bar{C}_i[\square]$ and substitution θ_i ; actually, since $t_i = s_i^\# = \theta_i(\bar{s}_i)^\# = \theta_i(\bar{s}_i^\#)$ and $t_i \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_{i+1})$, we can further say that $\bar{s}_i \in \mathcal{NHT}_\mathcal{Q}$.

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

No pair $u \rightarrow v \in \mathcal{Q}$ satisfies that $u \sqsupset v$. Otherwise, we get a contradiction.

Let Δ be the set of defined symbols of $\mathcal{R} \setminus \mathcal{U}_{eB}(\mathcal{R}, \mu, \mathcal{P})$. We show that after applying the μ -interpretation $I'_{\Delta, \mu}$ we obtain an infinite $(\mathcal{Q}, \mathcal{U}_{eB}(\mathcal{R}, \mu, \mathcal{P}) \cup \mathcal{C}_\varepsilon, \mu)$ -chain. All terms in the infinite $(\mathcal{Q}, \mathcal{R}, \mu)$ -chain are μ -terminating with respect to (\mathcal{R}, μ) and hence we can indeed apply the μ -interpretation $I'_{\Delta, \mu}$. Let $i \geq 1$.

■ First consider the step $u_i \rightarrow_{\mathcal{Q}} \circ \triangleright_{\mu}^{\sharp} t_i$. We have two possibilities:

1. If there is $u_i \rightarrow v_i \in \mathcal{Q}_{\mathcal{F}}$, we follow the proof in Theorem 79.
2. If there is an $u_i \rightarrow v_i = u_i \rightarrow x \in \mathcal{Q}_{\mathcal{X}}$ and a substitution σ such that $\sigma(x) = C_i[t_i^{\sharp}]$. As in the proof of Lemma 78, according to Item 1 and Item 2, by structural induction on u_i we easily get $\sigma_{I'_{\Delta, \mu}}(u_i) = \bar{\sigma}_{I'_{\Delta, \mu, u_i}}(u_i)$. Since x is a migrating variable, $\bar{\sigma}_{I'_{\Delta, \mu, u_i}}(x) = \sigma(x)$. We have that $\bar{\sigma}_{I'_{\Delta, \mu, u_i}}(x) = C_i[t_i^{\sharp}]$. We know that $C_i[\square]$ is an instance of a hiding context $\bar{C}_i[\square]$, t_i^{\sharp} is an instance of a hidden term $\bar{s}_i \in \mathcal{NHT}_{\mathcal{Q}}$, $Var^{\mu}(\bar{s}_i) = \emptyset$, and $\mathcal{DFun}^{\mu}(\bar{s}_i^{\sharp}) \notin \Delta$, then $\sigma(\bar{s}_i^{\sharp}) = \bar{\sigma}_{I'_{\Delta, \mu, u_i}}(\bar{s}_i^{\sharp}) = \bar{\sigma}_{I'_{\Delta, \mu, \bar{s}_i^{\sharp}}}(\bar{s}_i^{\sharp}) = \sigma_{I'_{\Delta, \mu}}(\bar{s}_i^{\sharp}) = I'_{\Delta, \mu}(t_i)$. We obtain:

$$\begin{aligned} I'_{\Delta, \mu}(u_i) &= \bar{\sigma}_{I'_{\Delta, \mu, u_i}}(u_i) \rightarrow_{\mathcal{Q}_{\mathcal{X}}} \bar{\sigma}_{I'_{\Delta, \mu, u_i}}(x) = \sigma(x) = C_i[t_i^{\sharp}] \\ &\quad \text{and} \\ t_i &= \sigma(\bar{s}_i^{\sharp}) = \bar{\sigma}_{I'_{\Delta, \mu, \bar{s}_i^{\sharp}}}(\bar{s}_i^{\sharp}) = \sigma_{I'_{\Delta, \mu}}(\bar{s}_i^{\sharp}) = I'_{\Delta, \mu}(\sigma(\bar{s}_i^{\sharp})) = I'_{\Delta, \mu}(t_i) \end{aligned}$$

■ Next consider the rewrite sequence $t_i \xrightarrow{*}_{\mathcal{R}, \mu} \sigma(u_{i+1})$, we follow the proof in Theorem 79.

So we obtain the infinite rewrite sequence:

$$I'_{\Delta, \mu}(\sigma(u_1)) \xrightarrow{\mathcal{Q}, \mu} \circ \triangleright_{\mu}^{\sharp} I'_{\Delta, \mu}(t_1) \xrightarrow{*}_{\mathcal{U}_{eB}(\mathcal{R}, \mu, \mathcal{P}) \cup \mathcal{C}_\varepsilon, \mu} I'_{\Delta, \mu}(\sigma(u_2)) \xrightarrow{\mathcal{Q}, \mu} \circ \triangleright_{\mu}^{\sharp} \dots$$

Using the premises of the theorem, by monotonicity and stability of \succsim , we would have that $I'_{\Delta, \mu}(t_i) \succsim I'_{\Delta, \mu}(\sigma(u_{i+1}))$ for all $i \geq 1$. By stability of \sqsupset (and of \succsim), we have that $I'_{\Delta, \mu}(\sigma(u_i)) (\succsim \cup \sqsupset) I'_{\Delta, \mu}(t_i)$ for all $i \geq 1$ and $I'_{\Delta, \mu}(\sigma(u_i)) \sqsupset I'_{\Delta, \mu}(t_i)$ for all $j \in J$ for an infinite set $J = \{j_1, \dots, j_n, \dots\}$ of natural numbers $j_1 < j_2 < \dots < j_n < \dots$. We would obtain an infinite sequence consisting of infinitely many \sqsupset -steps, a contradiction to the well-foundedness of \sqsupset . Therefore, $\mathcal{Q} \subseteq (\mathcal{P} \setminus \mathcal{P}_{\sqsupset})$, thus leading a contradiction. ■

Usable rules for arbitrary CS-termination problems

In this section, we consider *arbitrary* CS-termination problems. In rewriting and in the previous subsection, when considering infinite minimal $(\mathcal{P}, \mathcal{R})$ -chains or $(\mathcal{P}, \mathcal{R}, \mu)$ -chains, we only deal with terminating terms over \mathcal{R} or (\mathcal{R}, μ) . The interpretations in Definition 70 and 71 is defined only for μ -terminating terms because non- μ -terminating terms would yield an infinite term which, actually, does *not* belong to $\mathcal{T}(\mathcal{F} \cup \{\perp, \mathbf{g}\}, \mathcal{X})$.

Similarly, we aim at defining a μ -interpretation $I_{\Delta, \mu}$ that allows us to associate an infinite $(\mathcal{P}, \mathcal{U}(\mathcal{R}, \mu, \mathcal{P}) \cup \mathcal{C}_{\varepsilon}, \mu)$ -chain to each infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain. Actually, the main problems are:

- $(\mathcal{P}, \mathcal{R}, \mu)$ -chains contain non- μ -terminating terms in frozen positions which are potentially able to reach μ -replacing positions: subterms at an active position are μ -terminating, but we do not know anything about subterms at frozen positions.
- The interpretation of nonterminating terms (independently if they are μ -terminating or not) generates infinite terms.

Hence, we have to define our μ -interpretation $I_{\Delta, \mu}$ both on μ -terminating and non- μ -terminating terms and avoid to interpret these nonterminating terms.

Intuitively, terms at frozen positions in the right-hand side of the rules are essential to track infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chains involving collapsing CS-DPs. These terms, by definition, are formed by *hidden symbols*. This observation gives us the key to properly generalize Definition 70. Following Definition 70, a μ -terminating but nonterminating term generates an infinite list. For this reason, I_{Δ} and $I'_{\Delta, \mu}$ (as a mapping from finite into finite terms) are *not* defined for nonterminating terms.

Regarding our μ -interpretation, if we consider the rules headed by hidden symbols (and those captured by the dependency relation) as *usable* and non- μ -terminating term t (at a frozen position) is treated as if its root symbol does not belong to Δ , because if it occurs in the $(\mathcal{P}, \mathcal{R}, \mu)$ -chain at a μ -replacing position, then $t \succeq_{\mu} s$ and $s^{\#}$ becomes the next term in the chain. Our new μ -interpretation is:

Definition 84 (μ -interpretation) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$ and $\Delta \subseteq \mathcal{F}$ be such that $\Delta \cap \mathcal{H}(\mathcal{R}, \mu) = \emptyset$. Let $>$ be an arbitrary total ordering over $\mathcal{T}(\mathcal{F} \cup \{\perp, \mathbf{g}\}, \mathcal{X})$ where \perp is a new constant symbol and \mathbf{g} is a new binary symbol (with $\mu(\mathbf{g}) = \{1, 2\}$). The μ -interpretation $I_{\Delta, \mu}$ is a mapping from arbitrary*

terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$ to terms in $\mathcal{T}(\mathcal{F} \cup \{\perp, \mathbf{g}\}, \mathcal{X})$ defined as follows:

$$I_{\Delta, \mu}(t) = \begin{cases} t & \text{if } t \in \mathcal{X} \\ f(I_{\Delta, \mu}(t_1), \dots, I_{\Delta, \mu}(t_n)) & \text{if } t = f(t_1 \dots t_n) \text{ and } f \notin \Delta \\ & \text{or } t \text{ is non-}\mu\text{-terminating} \\ \mathbf{g}(f(I_{\Delta, \mu}(t_1), \dots, I_{\Delta, \mu}(t_n)), t') & \text{if } t = f(t_1 \dots t_n) \text{ and } f \in \Delta \\ & \text{and } t \text{ is } \mu\text{-terminating} \end{cases}$$

where $t' = \text{order}(\{I_{\Delta, \mu}(u) \mid t \hookrightarrow_{(\mathcal{R}, \mu)} u\})$

$$\text{order}(T) = \begin{cases} \perp, & \text{if } T = \emptyset \\ \mathbf{g}(t, \text{order}(T \setminus \{t\})) & \text{if } t \text{ is minimal in } T \text{ w.r.t. } > \end{cases}$$

Now, we prove that $I_{\Delta, \mu}$ is well-defined. The most important difference (and essential in our proof) among our μ -interpretation and all previous ones [Gra94, GTSK04, HM04, Urb04] is that $I_{\Delta, \mu}$ is well-defined both for μ -terminating and non- μ -terminating terms.

Lemma 85 (Well-definition of $I_{\Delta, \mu}$) *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$ and let $\Delta \subseteq \mathcal{F} \setminus \mathcal{H}$. Then, $I_{\Delta, \mu}$ is well-defined.*

Proof. According to Definition 84, the only way to get an infinite term as the output of $I_{\Delta, \mu}(t)$ for a given term t is performing an infinite number of applications of the third item of the definition of $I_{\Delta, \mu}$ as part of the recursive calls to $I_{\Delta, \mu}$ when computing $I_{\Delta, \mu}(t)$. This means that, without losing generality, we can assume that t is μ -terminating and that there exists an infinite sequence of the form $t = u_1 \triangleright_{\mathcal{R}, \mu} t_1 \hookrightarrow_{\mathcal{R}, \mu} u_2 \triangleright_{\mathcal{R}, \mu} t_2 \hookrightarrow_{\mathcal{R}, \mu} u_3 \dots$ where $\text{root}(t_i) \in \Delta$ and t_i is μ -terminating for all $i \geq 1$. We can assume minimality of this sequence without loss of generality, that is, for all $i \geq 1$ there is no s_i such that $t_i \triangleright s_i$ and s_i generates an infinite sequence leading to an infinite definition of $I_{\Delta, \mu}$. Since t_{i+1} is not a subterm of t_i then there is a rule $l \rightarrow r$ and a position p , such that $t_i = C[\sigma(l)]_p \hookrightarrow_{\mathcal{R}, \mu} C[\sigma(r)]_p = u_{i+1} \triangleright_{\mathcal{R}, \mu} t_{i+1}$. We have only two possibilities:

- $t_{i+1} \triangleright_{\mathcal{R}, \mu} \sigma(r)$. Then $u_{i+1} \triangleright_{\mathcal{R}, \mu} t_{i+1}$ because $\sigma(r)$ is at an active position.
- $\sigma(r) \triangleright \sigma(s) = t_{i+1}$ for some s such that $s \notin \mathcal{X}$ and $r \triangleright s$. Since $\text{root}(s) \in \Delta$ and $\Delta \cap \mathcal{H} = \emptyset$ we have that $\text{root}(s) \notin \mathcal{H}$ and $r \triangleright_{\mathcal{R}, \mu} s$, hence $u_{i+1} \triangleright_{\mathcal{R}, \mu} t_{i+1}$.

Then, the resulting sequence is: $t = u_1 \triangleright_{\mathcal{R}, \mu} t_1 \hookrightarrow_{\mathcal{R}, \mu} u_2 \triangleright_{\mathcal{R}, \mu} t_2 \hookrightarrow_{\mathcal{R}, \mu} u_3 \dots$, which can be considered as an infinite μ -rewrite sequence starting from t , thus contradicting the μ -termination of t . ■

Now, we define an appropriate notion of direct μ -dependency. This is not straightforward as shown in the next example.

Example 86

Consider the following conservative TRS \mathcal{R} :

$$a(x, y) \rightarrow b(x, x) \quad (6.28) \qquad d(x, e) \rightarrow a(x, x) \quad (6.30)$$

$$b(x, c) \rightarrow d(x, x) \quad (6.29) \qquad c \rightarrow e \quad (6.31)$$

with $\mu(a) = \mu(d) = \{1, 2\}$, $\mu(b) = \{1\}$ and $\mu(c) = \mu(e) = \emptyset$. We have the following CS-termination problem:

$$\begin{aligned} (\mathcal{P} = \{ & A(x, y) \rightarrow B(x, x), \\ & B(x, c) \rightarrow D(x, x), \\ & D(x, e) \rightarrow A(x, x) \}, \mathcal{R}, \mu) \end{aligned}$$

According to Definition 66, we have no usable rules because the right-hand sides of the dependency pairs have no defined symbols.

Without considering the rule $c \rightarrow e$ we may conclude on termination of this CS-termination problem, but there is the following infinite $(\mathcal{P}, \mathcal{R}, \mu)$ -chain:

$$\underline{A(c, c)} \hookrightarrow_{\mathcal{P}, \mu} \underline{B(c, c)} \hookrightarrow_{\mathcal{P}, \mu} \underline{D(c, c)} \hookrightarrow_{\{c \rightarrow e\}, \mu} \underline{D(c, e)} \hookrightarrow_{\mathcal{P}, \mu} \underline{A(c, c)} \hookrightarrow_{\mathcal{P}, \mu} \dots$$

Hence, a first extension is to handle this problem in the dependency definition:

Definition 87 *Given a TRS $\mathcal{R} = (\mathcal{F}, R)$ and $\mu \in M_{\mathcal{F}}$, we say that $f \in \mathcal{F}$ directly μ -depends on $g \in \mathcal{F}$, written $f \triangleright_{d, \mu} g$, if there is a rule $l \rightarrow r \in R$ with*

1. $f = \text{root}(l)$ and
2. g occurs in r at a μ -replacing position or
3. g occurs in l at a non- μ -replacing position.

Remarkably, condition (3) in Definition 87 is not very problematic in practice because most programs are *constructor systems*, which means that no defined symbols occur below the root in the left-hand side of the rules.

Now, we are ready to define our notion of usable rules. To have a well-founded interpretation, the necessity of adding the rules related with hidden symbols as

usable is mandatory, as we see in Proposition 85. The set of *non-replacing* defined function symbols in a term t is $\mathcal{DFun}^\#(t) = \{f \mid \exists p \in \text{Pos}(t) \text{ and } p \notin \text{Pos}^\mu(t), f = \text{root}(t|_p) \in \mathcal{D}\}$.

Definition 88 (Context-sensitive usable rules) *Let $\mathcal{R} = (\mathcal{F}, R)$, $\mathcal{P} = (\mathcal{G}, P)$ be TRSs, $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. The set $\mathcal{U}(\mathcal{R}, \mu, \mathcal{P})$ of context-sensitive usable rules for the CS-termination problem $(\mathcal{P}, \mathcal{R}, \mu)$ is given by*

$$\mathcal{U}(\mathcal{R}, \mu, \mathcal{P}) = \bigcup_{l \rightarrow r \in \mathcal{P}} \mathcal{U}(\mathcal{R}, \mu, l \rightarrow r) \cup \mathcal{U}_{\mathcal{H}}(\mathcal{R}, \mu)$$

$$\begin{aligned} \text{where } \mathcal{U}(\mathcal{R}, \mu, l \rightarrow r) &= \mathcal{R} \mid \{g \mid f \triangleright_{\mathcal{d}, \mu}^* g \text{ for some } f \in \mathcal{DFun}(r) \cup \mathcal{DFun}^\#(l)\} \\ \mathcal{U}_{\mathcal{H}}(\mathcal{R}, \mu) &= \mathcal{R} \mid \{g \mid f \triangleright_{\mathcal{d}, \mu}^* g \text{ for some } f \in \mathcal{H}(\mathcal{R}, \mu)\} \end{aligned}$$

Note that $\mathcal{U}(\mathcal{R}, \mu, l \rightarrow r)$ extends the notion of usable rules in Definition 66, by taking into account not only dependencies with symbols on the right-hand side of the rules, but *also with some symbols in proper subterms of the left-hand sides*. On the other hand, $\mathcal{U}_{\mathcal{H}}(\mathcal{R}, \mu)$ is the set of usable rules corresponding to the hidden symbols. Now, we are ready to formulate and prove our main result in this subsection, using auxiliary results first:

Lemma 89 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$ and let $\Delta \subseteq \mathcal{F} \setminus \mathcal{H}$. If t does not contain any Δ -symbol in a frozen position then $\text{I}_{\Delta, \mu}(\sigma(t)) \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}}^* \sigma_{\text{I}_{\Delta, \mu}}(t)$ and if t does not contain Δ -symbols then $\text{I}_{\Delta, \mu}(\sigma(t)) = \sigma_{\text{I}_{\Delta, \mu}}(t)$.*

Proof. By induction on t . If t is a variable then $\text{I}_{\Delta, \mu}(\sigma(t)) = \sigma_{\text{I}_{\Delta, \mu}}(t)$. Let $t = f(t_1, \dots, t_n)$. By induction hypothesis $\text{I}_{\Delta, \mu}(\sigma(t_i)) \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}}^* \sigma_{\text{I}_{\Delta, \mu}}(t_i)$ for $1 \leq i \leq n$. Moreover, by hypothesis: whenever $i \notin \mu(f)$ then t_i contains no Δ -symbol. Then $\text{I}_{\Delta, \mu}(\sigma(t_i)) = \sigma_{\text{I}_{\Delta, \mu}}(t_i)$ for all $i \notin \mu(f)$ by induction hypothesis, and hence $f(\text{I}_{\Delta, \mu}(\sigma(t_1)), \dots, \text{I}_{\Delta, \mu}(\sigma(t_n))) \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}}^* f(\sigma_{\text{I}_{\Delta, \mu}}(t_1), \dots, \sigma_{\text{I}_{\Delta, \mu}}(t_n)) = \sigma_{\text{I}_{\Delta, \mu}}(t)$.

- If $f \notin \Delta$ or t is non- μ -terminating, $\text{I}_{\Delta, \mu}(\sigma(t)) = f(\text{I}_{\Delta, \mu}(\sigma(t_1)), \dots, \text{I}_{\Delta, \mu}(\sigma(t_n)))$ then $\text{I}_{\Delta, \mu}(\sigma(t)) \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}}^* f(\sigma_{\text{I}_{\Delta, \mu}}(t_1), \dots, \sigma_{\text{I}_{\Delta, \mu}}(t_n)) = \sigma_{\text{I}_{\Delta, \mu}}(t)$.
And, if there are no Δ -symbols in t , then, $\text{I}_{\Delta, \mu}(\sigma(t_i)) = \sigma_{\text{I}_{\Delta, \mu}}(t_i)$ for all i , $1 \leq i \leq n$, hence $\text{I}_{\Delta, \mu}(\sigma(t)) = \sigma_{\text{I}_{\Delta, \mu}}(t)$.
- If $f \in \Delta$ and t is μ -terminating, then $\text{I}_{\Delta, \mu}(\sigma(t)) = \mathbf{g}(t', t'') \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}} t'$ for some term t'' , with $t' = f(\text{I}_{\Delta, \mu}(\sigma(t_1)), \dots, \text{I}_{\Delta, \mu}(\sigma(t_n)))$. Hence, $\text{I}_{\Delta, \mu}(t) \hookrightarrow_{\mathcal{C}_{\varepsilon, \mu}}^* \sigma_{\text{I}_{\Delta, \mu}}(t)$.

■

Lemma 90 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, $\Delta \subseteq \mathcal{F} \setminus \mathcal{H}$ and $C[\]$ a context with n μ -replacing holes. If $t = C[t_1, \dots, t_n]$ and the context $C[\]$ contains no Δ -symbols then $I_{\Delta, \mu}(C[t_1, \dots, t_n]) = C[I_{\Delta, \mu}(t_1), \dots, I_{\Delta, \mu}(t_n)]$.*

Proof. By structural induction. Similar to the proof of Lemma 76. ■

Lemma 91 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS, $\mu \in M_{\mathcal{F}}$, and $l \rightarrow r \in R$. Let $\Delta \subseteq \mathcal{F} \setminus \mathcal{H}$ such that if $\text{root}(l) \notin \Delta$ then $\mathcal{DFun}^{\#}(l) \notin \Delta$ and $\mathcal{DFun}(r) \notin \Delta$. If terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ are μ -terminating with respect to (\mathcal{R}, μ) and $s \hookrightarrow_{\{l \rightarrow r\}, \mu} t$ then $I_{\Delta, \mu}(s) \hookrightarrow_{\{l \rightarrow r\} \cup \mathcal{C}_{\varepsilon}, \mu}^+ I_{\Delta, \mu}(t)$ if $\text{root}(l) \notin \Delta$ or $I_{\Delta, \mu}(s) \hookrightarrow_{\mathcal{C}_{\varepsilon}, \mu}^+ I_{\Delta, \mu}(t)$ otherwise.*

Proof. Let $s \hookrightarrow_{\mathcal{R}, \mu} t$ occur at $p \in \mathcal{Pos}^{\mu}(s)$. There are two cases:

- If there is a function symbol from Δ at a position above or in p , we can write $s = C[s_1, \dots, s_i, \dots, s_n]$ and $t = C[s_1, \dots, t_i, \dots, s_n]$, with $s_i \hookrightarrow_{\mathcal{R}, \mu} t_i$, where s_i and t_i are at a μ -replacing hole, $\text{root}(s_i) \in \Delta$ and the context contains no Δ -symbol. We have $I_{\Delta, \mu}(s_i) \hookrightarrow_{\mathcal{C}_{\varepsilon}, \mu} \text{order}(\bigcup_{s_i \hookrightarrow_{\mathcal{R}, \mu} u} I_{\Delta, \mu}(u))$. Since $s_i \hookrightarrow_{\mathcal{R}, \mu} t_i$, appropriate $\mathcal{C}_{\varepsilon}$ -steps extract $I_{\Delta, \mu}(t_i)$ from the term $\text{order}(\bigcup_{s_i \hookrightarrow_{\mathcal{R}, \mu} u} I_{\Delta, \mu}(u))$, so $I_{\Delta, \mu}(s_i) \hookrightarrow_{\mathcal{C}_{\varepsilon}, \mu}^+ I_{\Delta, \mu}(t_i)$. By Lemma 90 we get $I_{\Delta, \mu}(s) \hookrightarrow_{\mathcal{C}_{\varepsilon}, \mu}^+ I_{\Delta, \mu}(t)$.
- Otherwise, we may write $s = C[s_1, \dots, s_i, \dots, s_n]$ and $t = C[s_1, \dots, t_i, \dots, s_n]$ with $s_i \hookrightarrow_{\mathcal{R}, \mu} t_i$, where the context $C[\]$ only contains non- Δ -symbols. Then, $\text{root}(s_i) \notin \Delta$. Let σ be a substitution with $\text{Dom}(\sigma) \subseteq \text{Var}(l)$ s.t. $s_i = \sigma(l)$ and $t_i = \sigma(r)$. By definition, symbols in l at frozen positions are not in Δ . Then by Lemma 89: $I_{\Delta, \mu}(s_i) = I_{\Delta, \mu}(\sigma(l)) \hookrightarrow_{\mathcal{C}_{\varepsilon}, \mu}^* \sigma_{I_{\Delta, \mu}}(l)$. Since right-hand side of rule do not contain Δ -symbols, Lemma 89 yields $I_{\Delta, \mu}(t_i) = I_{\Delta, \mu}(\sigma(r)) = \sigma_{I_{\Delta, \mu}}(r)$, $\sigma_{I_{\Delta, \mu}}(l) \hookrightarrow_{\{l \rightarrow r\}, \mu} \sigma_{I_{\Delta, \mu}}(r)$, thus $I_{\Delta, \mu}(s_i) \hookrightarrow_{\{l \rightarrow r\} \cup \mathcal{C}_{\varepsilon}, \mu}^* I_{\Delta, \mu}(t_i)$. Then, $I_{\Delta, \mu}(s) \hookrightarrow_{\{l \rightarrow r\} \cup \mathcal{C}_{\varepsilon}, \mu}^* I_{\Delta, \mu}(t)$ using Lemma 90.

■

Example 92

Consider the following non-conservative CS-TRS \mathcal{R} :

$$f(c(x), y) \rightarrow h(x, y) \quad (6.32) \qquad d \rightarrow c(b) \quad (6.34)$$

$$h(c(x), y) \rightarrow f(c(y), y) \quad (6.33) \qquad a \rightarrow d \quad (6.35)$$

together with $\mu(f) = \mu(h) = \{1, 2\}$ and $\mu(c) = \mu(b) = \mu(d) = \mu(a) = \emptyset$. We have the following CS-termination problem after applying the SCC processor:

$$(\{F(c(x), y) \rightarrow H(x, y), H(c(x), y) \rightarrow F(c(y), y)\}, \mathcal{R}, \mu)$$

If we want to capture the chain:

$$H(\underline{a}, a) \rightarrow H(\underline{d}, \underline{a}) \rightarrow H(\underline{d}, \underline{d}) \rightarrow \underline{H(c(b), d)} \rightarrow \underline{F(c(d), d)} \rightarrow H(\underline{d}, \underline{d}) \rightarrow \dots$$

Our interpretation will be:

$$I_{\Delta, \mu}(H(a, a)) = H(g(a, g(d, g(c(b), \perp))), g(a, g(d, g(c(b), \perp))))$$

and the infinite chain:

$$\begin{aligned} & H(g(a, g(d, g(c(b), \perp))), g(a, g(d, g(c(b), \perp)))) \\ \hookrightarrow_{\mathcal{C}_\varepsilon, \mu} & H(g(d, g(c(b), \perp)), \overline{Fg(a, g(d, g(c(b), \perp))}) \\ \hookrightarrow_{\mathcal{C}_\varepsilon, \mu} & H(g(d, g(c(b), \perp)), g(d, g(c(b), \perp))) \\ \hookrightarrow_{\mathcal{C}_\varepsilon, \mu}^* & H(c(b), g(d, g(c(b), \perp))) \\ \hookrightarrow_{\mathcal{P}, \mu} & \overline{F(c(g(d, g(c(b), \perp))), g(d, g(c(b), \perp))} \\ \hookrightarrow_{\mathcal{P}, \mu} & H(\underline{g(d, g(c(b), \perp))}, g(a, g(d, g(c(b), \perp)))) \\ \hookrightarrow_{\mathcal{C}_\varepsilon, \mu} & \dots \end{aligned}$$

Theorem 93 (Usable rules processor) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs, $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. If there exists a μ -reduction pair (\succsim, \sqsupset) such that*

(1) $\mathcal{U}(\mathcal{R}, \mu, \mathcal{P}) \cup \mathcal{C}_\varepsilon \subseteq \succsim$, $\mathcal{P} \subseteq \succsim \cup \sqsupset$, and

(2) whenever $\mathcal{NHT}_{\mathcal{P}} \neq \emptyset$ and $\mathcal{P}_X \neq \emptyset$, we have that

(a) $f(x_1, \dots, x_i, \dots, x_n) \succsim x_i$ for all $f \in \mathcal{F}$ and all position $i \in \mu_{\mathcal{H}, \mathcal{R}, \mu}(f)$,
and

(b) $t (\succsim \cup \sqsupset) t^\#$ for all $t \in \mathcal{NHT}_{\mathcal{P}}$,

Let $\mathcal{P}_\sqsupset = \{u \rightarrow v \in \mathcal{P} \mid u \sqsupset v\}$. Then, the processor $\text{Proc}_{RP_{ur}}$ given by

$$\text{Proc}_{RP_{ur}}(\mathcal{P}, \mathcal{R}, \mu) = \begin{cases} \{(\mathcal{P} \setminus \mathcal{P}_\sqsupset, \mathcal{R}, \mu)\} & \text{if (1) and (2) hold} \\ \{(\mathcal{P}, \mathcal{R}, \mu)\} & \text{otherwise} \end{cases}$$

is sound and complete.

Proof. We have to prove that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain if and only if there is an infinite minimal $(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathcal{R}, \mu)$ -chain. The *if* part is obvious. For the *only if* part, we proceed by contradiction. Assume that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain A , but that there is no infinite minimal $(\mathcal{P} \setminus \mathcal{P}_{\sqsupset}, \mathcal{R}, \mu)$ -chain. Due to the finiteness of \mathcal{P} , we can assume that there is $\mathcal{Q} \subseteq \mathcal{P}$ such that A has a tail B

$$\sigma(u_1) \hookrightarrow_{\mathcal{Q}, \mu} \circ \triangleright_{\mu}^{\sharp} t_1 \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_2) \hookrightarrow_{\mathcal{Q}, \mu} \circ \triangleright_{\mu}^{\sharp} t_2 \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_3) \hookrightarrow_{\mathcal{Q}, \mu} \circ \triangleright_{\mu}^{\sharp} \dots$$

for some substitution σ , where all pairs in \mathcal{Q} are infinitely often used, and, for all $i \geq 1$, (1) if $u_i \rightarrow v_i \in \mathcal{Q}_{\mathcal{G}}$, then $t_i = \sigma(v_i)$ and (2) if $u_i \rightarrow v_i = u_i \rightarrow x_i \in \mathcal{Q}_{\mathcal{X}}$, then $t_i = s_i^{\sharp}$ for some s_i and some context C_i with a μ -replacing hole such that $\sigma(x_i) = C_i[s_i]$ and $C_i[s_i] = \theta_i(\bar{C}_i[\bar{s}_i])$ for some $\bar{s}_i \in \mathcal{NHT}$, hiding context $\bar{C}_i[\square]$ and substitution θ_i ; actually, since $t_i = s_i^{\sharp} = \theta_i(\bar{s}_i)^{\sharp} = \theta_i(\bar{s}_i^{\sharp})$ and $t_i \hookrightarrow_{\mathcal{R}, \mu}^* \sigma(u_{i+1})$, we can further say that $\bar{s}_i \in \mathcal{NHT}_{\mathcal{Q}}$.

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

No pair $u \rightarrow v \in \mathcal{Q}$ satisfies that $u \sqsupset v$. Otherwise, we get a contradiction.

Let Δ be the set of defined symbols of $\mathcal{R} \setminus \mathcal{U}(\mathcal{R}, \mu, \mathcal{P})$. We show that after applying the μ -interpretation $I_{\Delta, \mu}$ we obtain an infinite $(\mathcal{Q}, \mathcal{U}(\mathcal{R}, \mu, \mathcal{P}) \cup \mathcal{C}_{\varepsilon}, \mu)$ -chain. All terms in the infinite $(\mathcal{Q}, \mathcal{R}, \mu)$ -chain are μ -terminating with respect to (\mathcal{R}, μ) and hence we can indeed apply the μ -interpretation $I_{\Delta, \mu}$. Let $i \geq 1$.

■ First consider the step $u_i \rightarrow_{\mathcal{Q}} \circ \triangleright_{\mu}^{\sharp} t_i$. We have two possibilities:

1. There is $u_i \rightarrow v_i \in \mathcal{Q}_{\mathcal{F}}$ and a substitution σ such that $t_i = \sigma(r)$. By definition of minimality, $\sigma(x)$ is μ -terminating for every $x \in \mathcal{Var}^{\mu}(u_i)$. The right-hand sides of dependency pairs in $\mathcal{Q}_{\mathcal{F}}$ have no Δ -symbols. Thus, we have $I_{\Delta, \mu}(\sigma(v_i)) = \sigma_{I_{\Delta, \mu}}(v_i)$ by Lemma 89. The same Lemma yields $I_{\Delta, \mu}(\sigma(u_i)) = \sigma_{I_{\Delta, \mu}}(u_i)$ because the definition of \mathcal{U} implies that all symbol at a frozen position in u_i are usable. Hence:

$$I_{\Delta, \mu}(\sigma(u_i)) \hookrightarrow_{\mathcal{C}_{\varepsilon}, \mu}^* \sigma_{I_{\Delta, \mu}}(u_i) \rightarrow_{\mathcal{Q}_{\mathcal{F}}} \sigma_{I_{\Delta, \mu}}(v_i) = I_{\Delta, \mu}(\sigma(v_i)) = I_{\Delta, \mu}(t_i)$$

2. There is an $u_i \rightarrow v_i = u_i \rightarrow x \in \mathcal{Q}_{\mathcal{X}}$ and a substitution σ such that $\sigma(x) = C_i[t_i^{\sharp}]_p$. All terms $\sigma(x)|_q$ for a position $\Lambda \leq q \leq p$ are non- μ -terminating, and by Definition 84, we have $\sigma_{I_{\Delta, \mu}}(x) = I_{\Delta, \mu}(\sigma(x))$. Hence, $\sigma_{I_{\Delta, \mu}}(x) = I_{\Delta, \mu}(\sigma(x)) = C_i[I_{\Delta, \mu}(t_i^{\sharp})]_p$. We know that $root(t_i^{\sharp}) \in \mathcal{H}$. Lemma 89 also

yields $I_{\Delta,\mu}(\sigma(u_i)) = \sigma_{I_{\Delta,\mu}}(u_i)$ because the definition of \mathcal{U} implies that all symbol at a frozen position in u_i is usable. Hence:

$$\begin{aligned} I_{\Delta,\mu}(\sigma(u_i)) &\xrightarrow{\mathcal{C}_{\varepsilon,\mu}^*} \sigma_{I_{\Delta,\mu}}(u_i) \xrightarrow{\mathcal{Q}_{\mathcal{X}}} \sigma_{I_{\Delta,\mu}}(x) = \sigma_{I_{\Delta,\mu}}(\bar{C}_i[\bar{s}_i]) \\ &\text{and} \\ \sigma_{I_{\Delta,\mu}}(\bar{s}_i^\sharp) &= I_{\Delta,\mu}(\sigma(\bar{s}_i^\sharp)) = I_{\Delta,\mu}(t_i) \end{aligned}$$

- Next consider the rewrite sequence $t_i \xrightarrow{\mathcal{R},\mu}^* \sigma(u_{i+1})$. By repeated applications of Lemma 91 we obtain $I_{\Delta,\mu}(t_i) \xrightarrow{\mathcal{U}(\mathcal{R},\mu,\mathcal{P}) \cup \mathcal{C}_{\varepsilon,\mu}^*} \sigma_{I_{\Delta,\mu}}(u_{i+1}) = I_{\Delta,\mu}(\sigma(u_{i+1}))$.

So we obtain the infinite rewrite sequence:

$$I_{\Delta,\mu}(\sigma(u_1)) \xrightarrow{\mathcal{Q},\mu} \circ \triangleright_{\mu}^\sharp I_{\Delta,\mu}(t_1) \xrightarrow{\mathcal{U}(\mathcal{R},\mu,\mathcal{P}) \cup \mathcal{C}_{\varepsilon,\mu}^*} I_{\Delta,\mu}(\sigma(u_2)) \xrightarrow{\mathcal{Q},\mu} \circ \triangleright_{\mu}^\sharp \dots$$

Using the premises of the theorem, by monotonicity and stability of \succsim , we would have that $I_{\Delta,\mu}(t_i) \succsim I_{\Delta,\mu}(\sigma(u_{i+1}))$ for all $i \geq 1$. By stability of \sqsupset (and of \succsim), we have that $I_{\Delta,\mu}(\sigma(u_i)) (\succsim \cup \sqsupset) I_{\Delta,\mu}(t_i)$ for all $i \geq 1$ and $I_{\Delta,\mu}(\sigma(u_i)) \sqsupset I_{\Delta,\mu}(t_i)$ for all $j \in J$ for an infinite set $J = \{j_1, \dots, j_n, \dots\}$ of natural numbers $j_1 < j_2 < \dots < j_n < \dots$. We would obtain an infinite sequence consisting of infinitely many \sqsupset -steps, a contradiction to the well-foundedness of \sqsupset . Therefore, $\mathcal{Q} \subseteq (\mathcal{P} \setminus \mathcal{P}_{\sqsupset})$, thus leading a contradiction. ■

Theorem 94 (Usable rules processor for collapsing pairs) *Let $\mathcal{R} = (\mathcal{F}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs, $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. If there exists a μ -reduction pair (\succsim, \sqsupset) such that*

- (1) $\mathcal{U}(\mathcal{R}, \mu, \mathcal{P}) \cup \mathcal{C}_{\varepsilon} \subseteq \succsim$, $\mathcal{P} \subseteq \succsim \cup \sqsupset$, and
- (2) $t \sqsupset t^\sharp$ for all $t \in \mathcal{NHT}_{\mathcal{P}}$,
- (3) $f(x_1, \dots, x_i, \dots, x_n) \succsim x_i$ for all $f \in \mathcal{F}$ and for all position $i \in \mu\mathcal{H}_{\mathcal{R},\mu}(f)$, and

Then, the processor $\text{Proc}_{RP_{cur}}$ given by

$$\text{Proc}_{RP_{cur}}(\mathcal{P}, \mathcal{R}, \mu) = \begin{cases} \{(\mathcal{P}_{\mathcal{G}}, \mathcal{R}, \mu)\} & \text{if (1), (2) and (3) hold} \\ \{(\mathcal{P}, \mathcal{R}, \mu)\} & \text{otherwise} \end{cases}$$

is sound and complete.

Proof. As in the proof of Theorem 93, the *if* part is obvious and for the *only if* part, we proceed by contradiction. We assume that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain A , but that there is no infinite minimal $(\mathcal{P}_{\mathcal{G}}, \mathcal{R}, \mu)$ -chain. Thus,

there is $\mathcal{Q} \subseteq \mathcal{P}$ such that $\mathcal{Q} \cap \mathcal{P}_{\mathcal{X}} \neq \emptyset$ and A has a tail B as in the proof of Theorem 93. Now, we assume the notation as in the first paragraph of such a proof.

We have $\sigma(u_i) (\succ \cup \sqsupset) t_i$ and $t_i \succ \sigma(u_{i+1})$ for all pairs $u_i \rightarrow v_i \in \mathcal{P}_{\mathcal{G}}$. If $u_i \rightarrow v_i = u_i \rightarrow x_i \in \mathcal{Q}_{\mathcal{X}}$, then by applying the considerations in the corresponding item of the proof of Theorem 93 and taking into account that $t \sqsupset t^\sharp$ for all $t \in \mathcal{NHT}_{\mathcal{P}}$, we have now that $\sigma(u_i) (\succ \cup \sqsupset) \sigma(x_i) \sqsupset t_i \succ \sigma(u_{i+1})$. Since pairs $u_i \rightarrow v_i \in \mathcal{Q}_{\mathcal{X}}$ occur infinitely often in B , by using the compatibility conditions of the μ -reduction pair, we obtain an infinite decreasing \sqsupset -sequence which contradicts well-foundedness of \sqsupset . \blacksquare

6.4. Subterm criterion

In [HM04, HM07], Hirokawa and Middeldorp introduce a very interesting *subterm criterion* which permits to ignore certain cycles of the dependency graph *without paying attention to the rules of the TRS*. Hirokawa and Middeldorp's result applies to *cycles* in the *dependency graph*. Recently, Thiemann has adapted it to the DP-framework [Thi07, Section 4.6]. The adaptation to CSR is made in [AGL08]. Here, we improve the two specific subterm processors which only apply to CSR.

Definition 95 (Root symbols of a TRS [AGL08]) Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS. The set of root symbols associated to \mathcal{R} is:

$$\text{Root}(\mathcal{R}) = \{\text{root}(l) \mid l \rightarrow r \in R\} \cup \{\text{root}(r) \mid l \rightarrow r \in R, r \notin \mathcal{X}\}$$

Definition 96 (Simple projection [AGL08]) Let \mathcal{R} be a TRS. A simple projection for \mathcal{R} is a mapping π that assigns to every k -ary symbol $f \in \text{Root}(\mathcal{R})$ an argument position $i \in \{1, \dots, k\}$. The mapping that assigns to every term $t = f(t_1, \dots, t_k)$ with $f \in \text{Root}(\mathcal{R})$ its subterm $\pi(t) = t|_{\pi(f)}$ is also denoted by π ; we also let $\pi(x) = x$ if $x \in \mathcal{X}$.

The following result provides a kind of generalization of the subterm criterion to simple projections which only take *non- μ -replacing* arguments.

Theorem 97 (Non- μ -replacing projection processor) Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs such that $\mathcal{P}_{\mathcal{G}}$ contains no collapsing rule, $\text{Root}(\mathcal{P}) \cap \mathcal{D} = \emptyset$, and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. Let \succsim be a stable quasi-ordering on terms whose strict and stable part $>$ is well-founded and π be a simple projection for \mathcal{P} such that

1. for all $f \in \text{Root}(\mathcal{P})$, $\pi(f) \notin \mu(f)$,

2. $\pi(\mathcal{P}) \subseteq \succsim$, and,
3. whenever $\mathcal{NHT}_{\mathcal{P}} \neq \emptyset$ and $\mathcal{P}_{\mathcal{X}} \neq \emptyset$, we have that
 - $f(x_1, \dots, x_i, \dots, x_n) \succsim x_i$ for all $f \in \mathcal{F}$ and for all position $i \in \mu_{\mathcal{H}, \mathcal{R}, \mu}(f)$, and
 - $t \succsim t|_{\pi(\text{root}(t)^\sharp)}$ for all $t \in \mathcal{NHT}_{\mathcal{P}}$.

Let $\mathcal{P}_{>} = \{u \rightarrow v \in \mathcal{P} \mid \pi(u) > \pi(v)\}$. Then, the processor Proc_{NRP} given by

$$\text{Proc}_{\text{NRP}}(\mathcal{P}, \mathcal{R}, \mu) = \begin{cases} \{(\mathcal{P} \setminus \mathcal{P}_{>}, \mathcal{R}, \mu)\} & \text{if (1), (2), and (3) hold} \\ \{(\mathcal{P}, \mathcal{R}, \mu)\} & \text{otherwise} \end{cases}$$

is sound and complete.

Proof. We have to prove that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain if and only if there is an infinite minimal $(\mathcal{P} \setminus \mathcal{P}_{>}, \mathcal{R}, \mu)$ -chain. The *if* part is obvious. For the *only if* part, we proceed by contradiction. Assume that there is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mu)$ -chain A but there is no infinite minimal $(\mathcal{P} \setminus \mathcal{P}_{>}, \mathcal{R}, \mu)$ -chain. Since \mathcal{P} is finite, we can assume that there is $\mathcal{Q} \subseteq \mathcal{P}$ such that A has a tail B

$$\sigma(u_1) \xrightarrow{\Lambda}_{\mathcal{Q}, \mu} \circ \triangleright_{\mu}^{\sharp} t_1 \xrightarrow{\ast}_{\mathcal{R}, \mu} \sigma(u_2) \xrightarrow{\Lambda}_{\mathcal{Q}, \mu} \circ \triangleright_{\mu}^{\sharp} t_2 \xrightarrow{\ast}_{\mathcal{R}, \mu} \dots$$

for some substitution σ and pairs $u_i \rightarrow v_i \in \mathcal{Q}$, and

1. if $v_i \notin \mathcal{X}$, then $t_i = \sigma(v_i)$, and
2. if $v_i = x_i \in \mathcal{X}$, then $x_i \notin \text{Var}^{\mu}(u_i)$ and $t_i = s_i^{\sharp}$ for some s_i and some context $C_i[\square]$ with a μ -replacing hole such that $\sigma(x_i) = C_i[s_i]$ and $C_i[s_i] = \theta_i(\bar{C}_i[\theta(\bar{s}_i)])$ for some $\bar{s}_i \in \mathcal{NHT}_{\mathcal{P}}$, some hiding context $\bar{C}_i[\square]$ and substitution θ_i .

Furthermore, *all pairs* in \mathcal{Q} are used infinitely often in B . For all $i \geq 1$, $\text{root}(t_i) \in \text{Root}(\mathcal{P})$, $\pi(t_i) \xrightarrow{\ast}_{\mathcal{R}, \mu} \pi(\sigma(u_{i+1}))$ and also $\text{root}(t_i) = \text{root}(u_{i+1})$ for all $i \geq 1$.

No pair $u \rightarrow v \in \mathcal{Q}$ satisfies that $\pi(u) > \pi(v)$. Otherwise, by applying the simple projection π to the sequence B , for all $i \geq 1$ we get a contradiction as follows:

1. Since $\pi(f) \notin \mu(f)$ for all $f \in \text{Root}(\mathcal{Q})$, for all $i \geq 1$, $\pi(t_i) = \pi(\sigma(u_{i+1})) = \sigma(\pi(u_{i+1}))$, because no μ -rewritings are possible on the $\pi(\text{root}(t_i))$ -th immediate subterm $\pi(t_i)$ of t_i , and
2. Due to $\pi(u_i) \succsim \pi(v_i)$ and by stability of \succsim , we have that $\pi(\sigma(u_i)) = \sigma(\pi(u_i)) \succsim \sigma(\pi(v_i))$. Now, we distinguish two cases:

- a) If $u_i \rightarrow v_i \in \mathcal{Q}_G$, then $\pi(t_i) = \pi(\sigma(v_i)) = \sigma(\pi(v_i))$. Thus, $\pi(\sigma(u_i)) \gtrsim \pi(t_i)$.
- b) If $u_i \rightarrow v_i \in \mathcal{Q}_X$, then $\sigma(\pi(v_i)) = \sigma(x_i)$. Since $\sigma(x_i) = C_i[s_i]$, we have that $\sigma(x_i) \gtrsim s_i$ (because $f(x_1, \dots, x_i, \dots, x_n) \gtrsim x_i$ for all symbols $f \in \mathcal{F}$ and position $i \in \mu_{\mathcal{H}, \mathcal{R}, \mu}(f)$ and $C_i[\square]$ is an instance of a hiding context $\bar{C}_i[\square]$). Let $f = \text{root}(u_{i+1}) = \text{root}(t_i) = \text{root}(\bar{s}_i^\#)$. Since $t \gtrsim t|_{\pi(\text{root}(t)^\#)}$ for all $t \in \mathcal{NHT}_{\mathcal{P}}$, by stability, we have $s_i = \theta_i(\bar{s}_i) \gtrsim \theta_i(\bar{s}_i|_{\pi(f)}) = \theta_i(\bar{s}_i)|_{\pi(f)} = s_i|_{\pi(f)}$. Since $s_i|_{\pi(f)} = t_i|_{\pi(f)} = \pi(t_i)$, we have $s_i \gtrsim \pi(t_i)$. Hence, $\pi(\sigma(u_i)) \gtrsim \pi(t_i)$.

Thus, we always have $\pi(\sigma(u_i)) \gtrsim \pi(t_i)$. Therefore, we obtain an infinite \gtrsim sequence

$$\pi(\sigma(u_1)) \gtrsim \pi(t_1) = \pi(\sigma(u_2)) \gtrsim \pi(t_2) \cdots$$

Since the dependency pairs in \mathcal{Q} occur infinitely many, this sequence contains infinitely many $>$ steps starting from $\pi(\sigma(u_1))$. This contradicts the well-foundedness of $>$.

Therefore, $\mathcal{Q} \subseteq \mathcal{P} \setminus \mathcal{P}_{>}$, which means that B is an infinite minimal $(\mathcal{P} \setminus \mathcal{P}_{>}, \mathcal{R}, \mu)$ -chain, thus leading to a contradiction with our initial assumption. \blacksquare

Example 98

Consider the CS-TRS (\mathcal{R}, μ) in Example 32. $\text{DP}(\mathcal{R}, \mu)$ is:

$$\text{G}(x) \rightarrow \text{H}(x) \qquad \qquad \qquad \text{H}(d) \rightarrow \text{G}(c)$$

where $\mu^\#(\text{G}) = \mu^\#(\text{H}) = \emptyset$. The dependency graph contains a single cycle including both of them. The only simple projection is $\pi(\text{G}) = \pi(\text{H}) = 1$. Since $\pi(\text{G}(x)) = \pi(\text{H}(x))$, we only need to guarantee that $\pi(\text{H}(d)) = d > c = \pi(\text{G}(c))$ holds for a stable and well-founded ordering $>$. This is easily fulfilled by, e.g., a polynomial ordering.

Theorem 99 (Non- μ -replacing projection processor II) *Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ and $\mathcal{P} = (\mathcal{G}, P)$ be TRSs such that $\mathcal{P}_{\mathcal{G}}$ contains no collapsing rule, $\text{Root}(\mathcal{P}) \cap \mathcal{D} = \emptyset$, and $\mu \in M_{\mathcal{F} \cup \mathcal{G}}$. Let \gtrsim be a stable quasi-ordering on terms whose strict and stable part $>$ is well-founded and π be a simple projection for \mathcal{P} such that*

1. for all $f \in \text{Root}(\mathcal{P})$, $\pi(f) \notin \mu(f)$,

2. $\pi(\mathcal{P}) \subseteq \succsim$, and,

3. whenever $\mathcal{NHT}_{\mathcal{P}} \neq \emptyset$ and $\mathcal{P}_{\mathcal{X}} \neq \emptyset$, we have that

- $f(x_1, \dots, x_i, \dots, x_n) \succsim x_i$ for all $f \in \mathcal{F}$ and for all position $i \in \mu_{\mathcal{H}, \mathcal{R}, \mu}(f)$,
and
- $t > t|_{\pi(\text{root}(t)^\sharp)}$ for all $t \in \mathcal{NHT}_{\mathcal{P}}$.

Then, the processor $\text{Proc}_{\text{NRP2}}$ given by

$$\text{Proc}_{\text{NRP2}}(\mathcal{P}, \mathcal{R}, \mu) = \begin{cases} \{(\mathcal{P} \setminus \mathcal{P}_{\mathcal{X}}, \mathcal{R}, \mu)\} & \text{if (1), (2), and (3) hold} \\ \{(\mathcal{P}, \mathcal{R}, \mu)\} & \text{otherwise} \end{cases}$$

is sound and complete.

7

Experiments

The processors described in the previous sections have been implemented as part of the tool MU-TERM [AGIL07, Luc04a]. We have tested the impact of the CSDP-framework in practice on the 90 examples in the Context-Sensitive Rewriting subcategory of the 2007 Termination Competition:

<http://www.lri.fr/~marche/termination-competition/2007>

which are part of the Termination Problem Data Base (TPDB, version 4.0):

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

We have addressed this task in three different ways:

1. We have compared CSDPs with previously existing techniques for proving termination of CSR: transformations, CSRPO, and polynomial orderings.
2. We have compared the improvements introduced by the different CS-processors which have been defined in this thesis.
3. We have participated in the CSR subcategory of the 2007 International Termination Competition.

Tool Version	Proved	Total Time	Average Time
CSDPs	68/90	0.43s	0.01s
CSRPO	37/90	0.21s	0.01s
Polynomial Orderings	27/90	0.06s	0.01s
Transformations	56/90	5.59s	0.10s

Table 7.1: Comparison among CSR Termination Techniques

7.1. CSDPs vs. other techniques for proving termination of CSR

Several methods have been developed to prove termination of CSR for a given CS-TRS (\mathcal{R}, μ) . Two main approaches have been investigated so far:

1. *Direct proofs*, which are based on using μ -reduction orderings (see [Zan97]) such as the (context-sensitive) recursive path orderings [BLR02] and polynomial orderings [GL02, Luc04b, Luc05]. These are orderings $>$ on terms which can be used to directly compare the left- and right-hand sides of the rules in order to conclude the μ -termination of the TRS.
2. *Indirect proofs* which obtain a proof of the μ -termination of \mathcal{R} as a proof of termination of a transformed TRS \mathcal{R}_Θ^μ (where Θ represents the transformation). If we are able to prove *termination* of \mathcal{R}_Θ^μ (using the standard methods), then the μ -termination of \mathcal{R} is ensured.

We have used MU-TERM to compare all these techniques with respect to the aforementioned benchmark examples. The results of this comparison are summarized in Table 7.1. From the benchmarks summarized in Table 7.1, we clearly conclude that the CSDP-framework is the most powerful and fastest technique for proving termination of CSR. Actually, all examples which were solved by using CSRPO or polynomial orderings were also solved using CSDPs. Regarding transformations, there is only one example (namely, `Ex9_Luc06`, which can be solved by using transformation GM) that could not be solved with our current implementation of the CS-processors but, as it is shown in [AEF⁺08], this CS-problem can be solved using the instantiation processor (that it is not implemented in the current version of MU-TERM).

7.2. Contribution of the different CS-processors

Our implementation of the CSDP-framework implements the CS-processors described in this thesis and the processors described in the technical report [AGL08]. We have considered the 32 versions of MU-TERM which are obtained by using all possible combinations (see Table 7.2). This CS-processors are:

- The CS-processor that transforms collapsing pairs into noncollapsing pairs: `ProceColl` (first column).

Tool Version	U symbols	Usable rules	Narrowing	Non- μ -replacing subterm criterion	Subterm criterion
CSDPs					
1.	NO	NO	NO	NO	NO
2.	NO	NO	NO	NO	YES
3.	NO	NO	NO	YES	NO
4.	NO	NO	NO	YES	YES
5.	NO	NO	YES	NO	NO
6.	NO	NO	YES	NO	YES
7.	NO	NO	YES	YES	NO
8.	NO	NO	YES	YES	YES
9.	NO	YES	NO	NO	NO
10.	NO	YES	NO	NO	YES
11.	NO	YES	NO	YES	NO
12.	NO	YES	NO	YES	YES
13.	NO	YES	YES	NO	NO
14.	NO	YES	YES	NO	YES
15.	NO	YES	YES	YES	NO
16.	NO	YES	YES	YES	YES
17.	YES	NO	NO	NO	NO
18.	YES	NO	NO	NO	YES
19.	YES	NO	NO	YES	NO
20.	YES	NO	NO	YES	YES
21.	YES	NO	YES	NO	NO
22.	YES	NO	YES	NO	YES
23.	YES	NO	YES	YES	NO
24.	YES	NO	YES	YES	YES
25.	YES	YES	NO	NO	NO
26.	YES	YES	NO	NO	YES
27.	YES	YES	NO	YES	NO
28.	YES	YES	NO	YES	YES
29.	YES	YES	YES	NO	NO
30.	YES	YES	YES	NO	YES
31.	YES	YES	YES	YES	NO
32.	YES	YES	YES	YES	YES
CSRPO					
33.					
Transformations					
34.		NO	NO		NO
35.		NO	NO		YES
36.		NO	YES		NO
37.		NO	YES		YES
Polynomial Orderings					
38.					

Table 7.2: Comparison among CSR Termination Techniques and CS-processors (Versions)

- The CS-processors involving μ -reduction pairs with usable rules: Proc_{RPur} , Proc_{RPcur} , Proc_{RPbur} and Proc_{RPebur} (second column).
- The CS-processor Proc_{narr} that use the narrowing transformation (third column).
- The CS-processors that use the subterm criterion over non- μ -replacing positions: Proc_{NRP} and Proc_{NRP2} (fourth column).

Tool Version	Proved	Total Time	Average Time
CSDPs			
1.	54/90	3.00s	0.05s
2.	62/90	0.55s	0.01s
3.	57/90	0.82s	0.01s
4.	65/90	0.49s	0.01s
5.	54/90	4.22s	0.08s
6.	62/90	2.64s	0.04s
7.	57/90	1.27s	0.02s
8.	65/90	0.31s	0.00s
9.	67/90	1.54s	0.02s
10.	67/90	0.80s	0.01s
11.	67/90	0.72s	0.01s
12.	67/90	0.51s	0.01s
13.	67/90	1.06s	0.01s
14.	67/90	0.67s	0.01s
15.	67/90	0.89s	0.01s
16.	67/90	0.53s	0.01s
17.	54/90	0.83s	0.01s
18.	62/90	0.29s	0.00s
19.	57/90	0.70s	0.01s
20.	66/90	0.71s	0.01s
21.	54/90	0.96s	0.02s
22.	62/90	0.27s	0.00s
23.	57/90	0.73s	0.01s
24.	66/90	0.32s	0.00s
25.	67/90	0.77s	0.01s
26.	68/90	0.44s	0.01s
27.	67/90	0.58s	0.01s
28.	68/90	0.59s	0.01s
29.	67/90	1.13s	0.02s
30.	68/90	0.59s	0.01s
31.	67/90	0.69s	0.01s
32.	68/90	0.43s	0.01s
CSRPO			
33.	37/90	0.21s	0.01s
Transformations			
34.	48/90	3.52s	0.07s
35.	56/90	3.74s	0.04s
36.	48/90	2.82s	0.06s
37.	56/90	5.59s	0.10s
Polynomial Orderings			
38.	27/90	0.06s	0.01s

Table 7.3: Comparison among CSR Termination Techniques and CS-processors

- The CS-processors that use the classical subterm criterion: $\text{Proc}_{subColl}$ and $\text{Proc}_{subNColl}$ (fifth column).

The performance of these implementations have been tested and summarized in Table 7.3. Our benchmarks show that the CS-processors described in this thesis

Termination Tool	Total	CSDPs	CSRPO	Transf.	Average time
MU-TERM	68	67	0	1	2.87s
AProVE	64	0	0	64	6.90s

Table 7.4: Results of the 2007 termination competition

play an important role in our proofs. The subterm processors $\text{Proc}_{\text{sub}N\text{Coll}}$ and $\text{Proc}_{\text{sub}Coll}$ are quite efficient, but the ones which are based on simple projections for non- μ -replacing arguments (Proc_{NRP} and Proc_{NRP2}) also increase the power and the speed of the CSDPs technique. Furthermore, these two groups of CS-processors are complementary: the extra problems which are specifically solved by them are different. Narrowing is useful to simplify the graph, but it doesn't play an important role in the benchmarks, because it only applies to solve two examples (which can be solved without narrowing as well). Furthermore, we have to carefully use it because recomputing the graph can be expensive in that case. The implementation of usable rules CS-processors is shown in the experiments as a very powerful processor, subsuming in these experiments other powerful techniques as the subterm processors. The CS-processor which transforms collapsing pairs has been shown very useful in many cases, obtaining simpler problems.

7.3. CSDPs at the 2007 International Termination Competition

Nowadays, AProVE [GSKT06] is the only tool (besides MU-TERM) which implements specific methods for proving termination of CSR.

Both AProVE and MU-TERM participated in the CSR subcategory of the 2007 International Termination Competition. AProVE participated with a termination expert for CSR which, given a CS-TRS (\mathcal{R}, μ) , successively tries different transformations Θ for proving termination of CSR (i.e., $\Theta \in \{C, FR, GM, L, sGM, Z\}$, see [Luc06] for a complete description of these transformations) and then uses (on the obtained TRS $\mathcal{R}_{\Theta}^{\mu}$) a huge variety of different and complementary techniques for proving termination of rewriting (according to the DP-framework). Actually, AProVE is currently the most powerful tool for proving termination of TRSs and implements most existing results and techniques regarding DPs and related techniques.

However, MU-TERM's implementation of CSDPs was able to beat AProVE in the CSR category, as shown in Table 7.3, thus witnessing that CSDPs are actually a very powerful technique for proving termination of CSR.

8

Conclusions

We have investigated the structure of infinite context-sensitive rewrite sequences starting from minimal non- μ -terminating terms (Theorem 29). This knowledge is used to provide an appropriate definition of context-sensitive dependency pair (Definition 30), and the related notion of chain (Definition 35). In sharp contrast to the standard dependency pairs approach, where all dependency pairs have tuple symbols $f^\#$ both in the left- and right-hand sides, we have *collapsing* dependency pairs having a single *variable* in the right-hand side. These variables reflect the effect of the *migrating* variables into the termination behavior of CSR. At the level of *minimal chains*, though, this contrast is somehow recovered by a nice symmetry arising from the central notion of *hidden term* (Definition 17) and *hiding context* (Definition 18): a noncollapsing pair $u \rightarrow v$ is followed by a pair $u' \rightarrow v'$ if $\sigma(v)$ μ -rewrites into $\sigma(u')$ for some substitution σ ; a collapsing pair $u \rightarrow v$ is followed by a pair $u' \rightarrow v'$ if there is a *hidden term* t with a *hiding context* C such that $\sigma(v) = \sigma(C)[\sigma(t)]$ and $\sigma(t^\#)$ μ -rewrites into $\sigma(u')$ for some substitution σ . We have shown how to use the context-sensitive dependency pairs in proofs of termination of CSR. As in Arts and Giesl's approach, the presence or absence of infinite chains of dependency pairs from $\text{DP}(\mathcal{R}, \mu)$ characterizes the μ -terminaton of \mathcal{R} (Theorems 37 and 38).

We have provided a suitable adaptation of Giesl et al.'s *dependency pair framework* to CSR by defining appropriate notions of *CS-termination problem* (Definition 40) and *CS-processor* (Definition 42). 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 CSDP-framework. In particular, we have also described some CS-processors for using μ -reduction pairs (Definition 55) and argument filterings to ensure the absence of infinite chains of pairs (Theorems 58 and 61). We have introduced a suitable notion of usable rules which is helpful to find μ -reduction pairs (Theorems 79, 83, 93 and 94). We have defined a CS-processor that transforms col-

lapsing pairs into noncollapsing ones. We have also introduced two new processors which work in a similar way to the subterm processor, but use a very basic kind of orderings instead of the subterm relation (Theorems 97 and 99). We have implemented these ideas as part of the termination tool MU-TERM [AGIL07, Luc04a]. 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, CSDPs were an essential ingredient for MU-TERM in winning the context-sensitive subcategory of the 2007 competition of termination tools.

Bibliography

- [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 XV International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR'08*, volume to appear of *Lecture Notes in Computer Science*, page to appear. 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*, volume 4337 of *Lecture Notes in Computer Science*, pages 297–308. Springer-Verlag, 2006.
- [AGL08] B. Alarcón, R. Gutiérrez, and S. Lucas. Context-sensitive dependency pairs. Technical report, Universidad Politécnica de Valencia, 2008. Available as Technical Report DSIC-II/10/08.
- [AL07] B. Alarcón and S. Lucas. Termination of Innermost Context-Sensitive Rewriting Using Dependency Pairs. In F. Wolter, editor, *Proc. of VI International Symposium on Frontiers of Combining Systems, FroCoS'07*, volume 4720 of *Lecture Notes in Computer Science*, pages 73–87, Liverpool, United Kingdom, 2007. Springer-Verlag.
- [BLR02] C. Borralleras, S. Lucas, and A. Rubio. Recursive Path Orderings can be Context-Sensitive. In A. Voronkov, editor, *Proc. of XVIII Conference on Automated Deduction, CADE'02*, volume 2392 of *Lecture Notes in Artificial Intelligence*, pages 314–331, Copenhagen, Denmark, 2002. Springer-Verlag.

- [BM06] R. Bruni and J. Meseguer. Semantic Foundations for Generalized Rewrite Theories. *Theoretical Computer Science*, 360(1):386–414, 2006.
- [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, Barcelona, Spain, 2003.
- [CDE⁺07] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. Talcott. *All About Maude – A High-Performance Logical Framework*, volume 4350 of *Lecture Notes in Computer Science*. Springer-Verlag, 2007.
- [DLM⁺04] F. Durán, S. Lucas, J. Meseguer, C. Marché, 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, 2004. ACM Press.
- [DLM⁺08] F. Durán, S. Lucas, C. Marché, J. Meseguer, and X. Urbain. Proving Operational Termination of Membership Equational Programs. *Higher Order Symbolic Computation*, 21(1-2):59–88, 2008.
- [EWZ08] J. Endrullis, J. Waldmann, and H. Zantema. Matrix Interpretations for Proving Termination of Term Rewriting. *Journal of Automated Reasoning*, 40(2-3):195–220, 2008.
- [FGJM85] K. Futatsugi, J. A. Goguen, J.-P. Jouannaud, and J. Meseguer. Principles of OBJ2. In *Proceedings of the 12th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, POPL'85*, pages 52–66, New York, NY, USA, 1985. ACM.
- [FN97] K. Futatsugi and A. Nakagawa. An overview of CAFE Specification Environment—An algebraic approach for creating, verifying, and maintaining formal specifications over networks. In *Proceedings of the 1st International Conference on Formal Engineering Methods, ICFEM '97*, page 170, Washington, DC, USA, 1997. IEEE Computer Society.

- [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.
- [GL02] 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, New York, United States of America, 2002. ACM Press.
- [GLU08] R. Gutiérrez, S. Lucas, and X. Urbain. Usable Rules for Context-Sensitive Rewrite Systems. In A. Voronkov, editor, *Proc. of XIX International Conference on Rewriting Techniques and Applications, RTA'08*, volume 5117 of *Lecture Notes in Computer Science*, pages 126–141, Hagenberg, Austria, 2008. Springer-Verlag.
- [GM99] J. Giesl and A. Middeldorp. Transforming Context-Sensitive Rewrite Systems. In P. Narendran and M. Rusinowitch, editors, *Proc. of X International Conference on Rewriting Techniques and Applications, RTA'99*, volume 1631 of *Lecture Notes in Computer Science*, pages 271–285, Trento, Italy, 1999. Springer-Verlag.
- [GM04] J. Giesl and A. Middeldorp. Transformation Techniques for Context-Sensitive Rewrite Systems. *Journal of Functional Programming*, 14(4):379–427, 2004.
- [Gra94] B. Gramlich. Generalized Sufficient Conditions for Modular Termination of Rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 5:131–151, 1994.
- [GSKT06] 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 III International Joint Conference on Automated Reasoning, IJCAR'06*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 281–286, Seattle, United States of America, 2006. Springer-Verlag. Available at <http://www-i2.informatik.rwth-aachen.de/AProVE>.
- [GTSK04] 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*, volume 3452 of *Lecture Notes in Artificial Intelligence*, pages 301–331, Montevideo, Uruguay, 2004. Springer-Verlag.
- [GTSKF06] 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. *Software Engineering with OBJ: Algebraic Specification in Action*, chapter Introducing OBJ. 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*, volume 3091 of *Lecture Notes in Computer Science*, pages 249–268, Aachen, Germany, 2004. Springer-Verlag.
- [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(4):474–511, 2007.
- [HPJW92] P. Hudak, S. L. 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.
- [Hul80] J.-M. Hullot. Canonical Forms and Unification. In *Proceedings of the 5th Conference on Automated Deduction, CADE'80*, pages 318–334, London, UK, 1980. Springer-Verlag.
- [KNT99] K. Kusakari, M. Nakamura, and Y. Toyama. Argument Filtering Transformation. In G. Nadathur, editor, *Proc. of International Conference on Principles and Practice of Declarative Programming, PPDP'99*, volume 1702 of *Lecture Notes in Computer Science*, pages 47–61, Pittsburgh, Pennsylvania, United States of America, 1999. ACM Press.
- [LM08] 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, volume to appear of *Electronic Notes in Theoretical Computer Science*, pages 189–205. Elsevier, 2008.

- [Luc96] S. Lucas. Termination of context-sensitive rewriting by rewriting. In F. Meyer auf der Heide and B. Monien, editors, *Proc. of the 23rd. International Colloquium on Automata, Languages and Programming, ICALP'96*, volume 1099 of *Lecture Notes in Computer Science*, pages 122–133, Berlin, Germany, 1996. Springer-Verlag.
- [Luc98] S. Lucas. Context-Sensitive Computations in Functional and Functional Logic Programs. *Journal of Functional and Logic Programming*, 1998(1):1–61, 1998.
- [Luc01] S. Lucas. Termination of on-demand rewriting and termination of OBJ programs. In *Proc. of III Internatinal Conference on Principles and Practice of Declarative Programming, PPDP'01*, pages 82–93, Florence, Italy, 2001. ACM Press.
- [Luc02] S. Lucas. Context-Sensitive Rewriting Strategies. *Information and Computation*, 178(1):293–343, 2002.
- [Luc04a] 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*, volume 3091 of *Lecture Notes in Computer Science*, pages 200–209, Aachen, Germany, 2004. Springer-Verlag. Available at <http://www.dsic.upv.es/~slucas/csr/termination/muterm>.
- [Luc04b] S. Lucas. Polynomials for Proving Termination of Context-Sensitive Rewriting. In I. Walukiewicz, editor, *Proc. of VII International Conference on Foundations of Software Science and Computation Structures, FOSSACS'04*, volume 2987 of *Lecture Notes in Computer Science*, pages 318–332, Berlin, Germany, 2004. Springer-Verlag.
- [Luc05] S. Lucas. Polynomials over the Reals in Proofs of Termination: from Theory to Practice. *RAIRO Theoretical Informatics and Applications*, 39(3):547–586, 2005.

- [Luc06] S. Lucas. Proving Termination of Context-Sensitive Rewriting by Transformation. *Information and Computation*, 204(12):1782–1846, 2006.
- [McC60] J. McCarthy. Recursive Functions of Symbolic Expressions and their Computation by Machine, Part I. *Communications of the ACM*, 3(4):184–195, 1960.
- [NSEP91] E. G. J. M. H. Nöcker, J. E. W. Smesters, M. C. J. D. van Eekelen, and M. J. Plasmeijer. Concurrent Clean. In E.H.L. Aarts, J. Leeuwen, and M. Rem, editors, *Proc. on Parallel Architectures and Languages Europe, PARLE'91*, pages 202–219, New York, NY, USA, 1991. Springer-Verlag.
- [Ohl02] E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer-Verlag, 2002.
- [TeR03] TeReSe. *Term Rewriting Systems*. Cambridge University Press, 2003.
- [TGSK04] R. Thiemann, J. Giesl, and P. Schneider-Kamp. Improved Modular Termination Proofs Using Dependency Pairs. In D. Basin and M. Rusinowitch, editors, *Proc. of II International Joint Conference on Automated Reasoning, IJCAR'04*, volume 3097 of *Lecture Notes in Artificial Intelligence*, pages 75–90, Cork, County Cork, Ireland, 2004. Springer-Verlag.
- [Thi07] R. Thiemann. *The DP Framework for Proving Termination of Term Rewriting*. PhD thesis, RWTH Aachen University, Aachen, Germany, 2007. Available as Technical Report AIB-2007-17.
- [Toy87] Y. Toyama. Counterexamples to Termination for the Direct Sum of Term Rewriting Systems. *Information Processing Letters*, 25:141–143, 1987.
- [Urb04] X. Urbain. Modular & Incremental Automated Termination Proofs. *Journal of Automated Reasoning*, 32(4):315–355, 2004.
- [Zan97] H. Zantema. Termination of Context-Sensitive Rewriting. In H. Comon, editor, *Proc. of VII International Conference on Rewriting Techniques and Applications, RTA'97*, volume 1232 of *Lecture Notes in Computer Science*, pages 172–186, Sitges, Spain, 1997. Springer-Verlag.