



ELSEVIER

Available online at www.sciencedirect.com ScienceDirect

**Electronic Notes in
Theoretical Computer
Science**

Electronic Notes in Theoretical Computer Science 188 (2007) 91–103

www.elsevier.com/locate/entcs

Improving the Context-sensitive Dependency Graph

Beatriz Alarcón, Raúl Gutiérrez, and Salvador Lucas^{1,2}*Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Valencia, Spain*

Abstract

The dependency pairs method is one of the most powerful technique for proving termination of rewriting and it is currently central in most automatic termination provers. Recently, it has been adapted to be used in proofs of termination of context-sensitive rewriting. The use of *collapsing* dependency pairs i.e., *having a single variable in the right-hand side* is a novel and essential feature to obtain a correct framework in this setting. Unfortunately, dependency pairs behave as a kind of *glue* in the context-sensitive dependency graph which makes the cycles bigger, thus making some proofs of termination harder. In this paper we show that this effect can be safely mitigated by removing some arcs from the graph, thus leading to faster and easier proofs. Narrowing dependency pairs is also introduced and used here to eventually simplify the treatment of the context-sensitive dependency graph. We show the practicality of the new techniques with some benchmarks.

Keywords: Dependency pairs, term rewriting, program analysis, termination.

1 Introduction

Termination is one of the most interesting problems when dealing with context-sensitive rewrite systems. With *context-sensitive rewriting* (*CSR* [10,11]) we can *achieve* a terminating behavior with non-terminating Term Rewriting Systems (TRSs [14,15]), by pruning (all) infinite rewrite sequences. In *CSR* we only rewrite μ -replacing subterms. Here, μ is a *replacement map*, i.e., a mapping $\mu : \mathcal{F} \rightarrow \mathcal{P}(\mathbb{N})$ satisfying $\mu(f) \subseteq \{1, \dots, k\}$, for each k -ary symbol f of the signature \mathcal{F} [10]. We use them to discriminate the argument positions on which the rewriting steps are allowed. Then, t_i is a μ -replacing subterm of $f(t_1, \dots, t_k)$ if $i \in \mu(f)$; every term t (as a whole) is μ -replacing by definition. For other subterms we proceed inductively

¹ This work has been partially supported by the EU (FEDER) and the Spanish MEC, under grants TIN 2004-7943-C04-02 and HA 2006-0007, the Generalitat Valenciana under grant GV06/285, and the ICT for EU-India Cross-Cultural Dissemination ALA/95/23/2003/077-054 project. Beatriz Alarcón was partially supported by the Spanish MEC under FPU grant AP2005-3399.

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

in this way. Then, for a given TRS, we obtain a restriction of rewriting which we call *context-sensitive rewriting*. Proving termination of *CSR* is an interesting problem with several applications in the fields of term rewriting and programming languages (see [5,6,8,11,13] for further motivation).

The *dependency pairs method* [1] 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 $\text{DP}(\mathcal{R})$ which (together with \mathcal{R}) determines the so-called *dependency chains* whose finiteness characterizes termination of \mathcal{R} . The dependency pairs can be presented as a *dependency graph*, where the absence of infinite chains can be analyzed by considering the *cycles* in the graph. In [3], the dependency pairs method has been adapted to be used in proofs of termination of *CSR*. The technique has been implemented in the tool MU-TERM [2,12]. Basically, the non-variable subterms in the right-hand sides of the rules which are considered to build the CS-dependency pairs must be μ -replacing terms. Nevertheless such ‘standard’ dependency pairs do not suffice to obtain a correct method for proving termination of *CSR*.

Example 1.1 [3, Example 2] Consider the following TRS \mathcal{R} :

$$\begin{array}{l} \mathbf{a} \rightarrow \mathbf{c}(\mathbf{f}(\mathbf{a})) \\ \mathbf{f}(\mathbf{c}(\mathbf{X})) \rightarrow \mathbf{X} \end{array}$$

together with $\mu(\mathbf{c}) = \emptyset$ and $\mu(\mathbf{f}) = \{1\}$. There is no μ -replacing subterm s in the right-hand sides of the rules which is rooted by a defined symbol. Thus, there is no ‘standard’ dependency pair. We could wrongly conclude that \mathcal{R} is μ -terminating, which is not true:

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

Indeed, as shown in [3], we must add the following dependency pair

$$\mathbf{F}(\mathbf{c}(\mathbf{X})) \rightarrow \mathbf{X}$$

which would not be allowed in Arts and Giesl’s approach [1] because the right-hand side is a variable. In this paper, we call *collapsing* to such kind of dependency pairs.

As in Arts and Giesl’s approach, the analysis of infinite sequences of context-sensitive dependency pairs can be made by looking at (the cycles \mathfrak{C} of) the *context-sensitive dependency graph* associated to the CS-TRS \mathcal{R} . The nodes of the dependency graph are the dependency pairs in $\text{DP}(\mathcal{R}, \mu)$. A disappointing aspect of *collapsing* context-sensitive dependency pairs (as $\mathbf{F}(\mathbf{c}(\mathbf{X})) \rightarrow \mathbf{X}$ above) is that they are connected to *every other* dependency pair in the context-sensitive dependency graph [3]. Intuitively, this is because the variable \mathbf{X} in the right-hand side of the dependency pair could be instantiated to anything, thus being potentially able to ‘connect’ to every other dependency pair.

In this paper, we show that we can restrict the number of outgoing links of collapsing dependency pairs to dependency pairs headed by the so-called *hidden* symbols which are defined symbols that occur in non-replacing positions in the right-hand sides of some rule in the TRS. This leads to a new definition of the context-sensitive dependency graph which greatly improves the performance of the original method.

Example 1.2 Consider the following non-terminating TRS \mathcal{R} which can be used to compute the list of prime numbers [7] :

```
primes -> sieve(from(s(s(0))))      tail(cons(X,Y)) -> Y
from(X) -> cons(X,from(s(X)))        if(true,X,Y) -> X
head(cons(X,Y)) -> X                 if(false,X,Y) -> Y
filter(s(s(X)),cons(Y,Z)) ->
  if(divides(s(s(X)),Y),filter(s(s(X)),Z),cons(Y,filter(X,sieve(Y))))
sieve(cons(X,Y)) -> cons(X,filter(X,sieve(Y)))
```

together with $\mu(\text{cons}) = \mu(\text{if}) = \{1\}$ and $\mu(f) = \{1, \dots, \text{ar}(f)\}$ for any other symbols f . No (automatic or manual) proof of termination for this CS-TRS has been reported to date. By using the dependency graph as defined in [3] we were not able to find a proof with MU-TERM 4.3 [2].

In contrast, with the new definition in this paper, we have *no cycles*! Thus, a direct (and automatic) proof of μ -termination of \mathcal{R} is easy now.

Narrowing dependency pairs was also introduced by Arts and Giesl to improve the efficiency of the dependency pairs technique in proofs of termination [1]. Roughly speaking, under some conditions, a dependency pair can be replaced by a set of pairs which could simplify or restructure the dependency graph and eventually simplify the proof of termination. We also investigate this technique for dealing with the context-sensitive dependency graph.

After some preliminary definitions in Section 2, Section 3 introduces the notion of hidden symbol and investigates its properties in proofs of termination of *CSR*. Section 4 shows how to use it to improve the context-sensitive dependency graph. Section 5 adapts narrowing of dependency pairs to context-sensitive dependency pairs. Section 6 provides an experimental evaluation of our techniques. Section 7 concludes.

2 Preliminaries

Throughout the paper, \mathcal{X} denotes a countable set of variables and \mathcal{F} denotes a signature, i.e., a set of function symbols $\{\mathbf{f}, \mathbf{g}, \dots\}$, each having a fixed arity given by a mapping $\text{ar} : \mathcal{F} \rightarrow \mathbb{N}$. The set of terms built from \mathcal{F} and \mathcal{X} is $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Positions p, q, \dots are represented by chains of positive natural numbers used to address subterms of t . Given positions p, q , we denote its concatenation as $p.q$. If p is a position, and Q is a set of positions, $p.Q = \{p.q \mid q \in Q\}$. We denote the empty chain by Λ . The set of positions of a term t is $\text{Pos}(t)$. The subterm at position p of t is denoted as $t|_p$ and $t[s]_p$ is the term t with the subterm at position p replaced by s . We write $t \geq s$ if $s = t|_p$ for some $p \in \text{Pos}(t)$ and $t \triangleright s$ if $t \geq s$ and $t \neq s$. The symbol labelling 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 zero or more ‘holes’ \square (a fresh constant symbol).

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 TRS is a pair $\mathcal{R} = (\mathcal{F}, R)$ where R is a set of rewrite rules. Given $\mathcal{R} = (\mathcal{F}, R)$, we consider \mathcal{F} as the disjoint union $\mathcal{F} = \mathcal{C} \uplus \mathcal{D}$ of symbols $c \in \mathcal{C}$, called *constructors* and symbols $f \in \mathcal{D}$, called *defined functions*, where

$\mathcal{D} = \{\text{root}(l) \mid l \rightarrow r \in R\}$ and $\mathcal{C} = \mathcal{F} - \mathcal{D}$.

Context-sensitive rewriting.

A mapping $\mu : \mathcal{F} \rightarrow \mathcal{P}(\mathbb{N})$ is a *replacement map* (or \mathcal{F} -map) if $\forall f \in \mathcal{F}, \mu(f) \subseteq \{1, \dots, \text{ar}(f)\}$ [10]. Let $M_{\mathcal{F}}$ be the set of all \mathcal{F} -maps (or $M_{\mathcal{R}}$ for the \mathcal{F} -maps of a TRS (\mathcal{F}, R)). A binary relation R on terms is μ -monotonic if $t R s$ implies $f(t_1, \dots, t_{i-1}, t, \dots, t_k) R f(t_1, \dots, t_{i-1}, s, \dots, t_k)$ for every $t, s, t_1, \dots, t_k \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. The set of μ -replacing positions $\text{Pos}^\mu(t)$ of $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ is: $\text{Pos}^\mu(t) = \{\Lambda\}$, if $t \in \mathcal{X}$ and $\text{Pos}^\mu(t) = \{\Lambda\} \cup \bigcup_{i \in \mu(\text{root}(t))} i. \text{Pos}^\mu(t|_i)$, if $t \notin \mathcal{X}$. The set of replacing variables of t is $\text{Var}^\mu(t) = \{x \in \text{Var}(t) \mid \exists p \in \text{Pos}^\mu(t), t|_p = x\}$. The μ -replacing subterm relation \triangleright_μ is given by $t \triangleright_\mu s$ if there is $p \in \text{Pos}^\mu(t)$ such that $s = t|_p$. We write $t \triangleright_\mu s$ if $t \triangleright_\mu s$ and $t \neq s$. In *context-sensitive rewriting* (CSR [10]), we (only) contract replacing redexes: t μ -rewrites to s , written $t \hookrightarrow_\mu s$ (or $t \hookrightarrow_{\mathcal{R}, \mu} s$), if $t \xrightarrow{p}_{\mathcal{R}} s$ and $p \in \text{Pos}^\mu(t)$. A TRS \mathcal{R} is μ -terminating if \hookrightarrow_μ is terminating. A term t is μ -terminating if there is no infinite μ -rewrite sequence $t = t_1 \hookrightarrow_\mu t_2 \hookrightarrow_\mu \dots \hookrightarrow_\mu t_n \hookrightarrow_\mu \dots$ starting from t . A pair (\mathcal{R}, μ) where \mathcal{R} is a TRS and $\mu \in M_{\mathcal{R}}$ is often called a CS-TRS.

Dependency pairs.

Given a TRS $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ a new TRS $\text{DP}(\mathcal{R}) = (\mathcal{F}^\sharp, D(R))$ of *dependency pairs* for \mathcal{R} is given as follows: if $f(t_1, \dots, t_m) \rightarrow r \in R$ and $r = C[g(s_1, \dots, s_n)]$ for some defined symbol $g \in \mathcal{D}$ and $s_1, \dots, s_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, then $f^\sharp(t_1, \dots, t_m) \rightarrow g^\sharp(s_1, \dots, s_n) \in D(R)$, where f^\sharp and g^\sharp are new fresh symbols (called *tuple symbols*) associated to defined symbols f and g respectively [1]. Let \mathcal{D}^\sharp be the set of tuple symbols associated to symbols in \mathcal{D} and $\mathcal{F}^\sharp = \mathcal{F} \cup \mathcal{D}^\sharp$. As usual, for $t = f(t_1, \dots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$, we write t^\sharp to denote the *marked term* $f^\sharp(t_1, \dots, t_k)$. 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 $f(t_1, \dots, t_k) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. Given $T \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})$, let T^\sharp be the set $\{t^\sharp \mid t \in T\}$.

3 Structure of infinite μ -rewrite sequences

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. Obviously, if $t \in \mathcal{M}_{\infty, \mu}$, then $\text{root}(t)$ is a defined symbol. Furthermore, since μ -terminating terms are preserved under μ -rewriting, it follows that $\mathcal{M}_{\infty, \mu}$ is also preserved under *inner* μ -rewritings.

Lemma 3.1 *Let \mathcal{R} be a TRS and $\mu \in M_{\mathcal{R}}$. Let $t \in \mathcal{M}_{\infty, \mu}$. If $t \xrightarrow{>^e}^* s$, then $s \in \mathcal{M}_{\infty, \mu}$.*

The following proposition establishes that, given $t \in \mathcal{M}_{\infty, \mu}$, there are 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 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 which are activated by *migrating* variables in a rule $l \rightarrow r$, i.e., variables $x \in \text{Var}^\mu(r) - \text{Var}^\mu(l)$ which are *not* μ -replacing in the left-hand side l but become μ -replacing in the right-hand side r .

Proposition 3.2 [3] *Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and $\mu \in M_{\mathcal{R}}$. 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 $t \xrightarrow{\epsilon}^* \sigma(l) \xrightarrow{\epsilon} \sigma(r) \triangleright_\mu u$ and either (1) there is a μ -replacing subterm s of r such that $u = \sigma(s)$, or (2) there is $x \in \text{Var}^\mu(r) - \text{Var}^\mu(l)$ such that $\sigma(x) \triangleright_\mu u$.*

Now we investigate the structure of such sequences in more detail. In the following, we write $t \triangleright_\mu s$ to denote that s is a non-replacing (hence strict!) subterm of t : $t \triangleright_\mu s$ if there is $p \in \text{Pos}(t) - \text{Pos}^\mu(t)$ such that $s = t|_p$.

Definition 3.3 [Hidden symbol] Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{R}}$. We say that $f \in \mathcal{F}$ is a *hidden symbol* if there is a rule $l \rightarrow r \in R$ and $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $r \triangleright_\mu t$ and $\text{root}(t) = f$. Let $\mathcal{H}(\mathcal{R}, \mu)$ (or just \mathcal{H} , if \mathcal{R} and μ are clear for the context) be the set of all hidden symbols in (\mathcal{R}, μ) .

Lemma 3.4 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ and σ be a substitution. If there is a rule $l \rightarrow r \in R$ such that $\sigma(l) \not\triangleright t$ and $\sigma(r) \triangleright_\mu t$, then there is no $x \in \text{Var}(r)$ such that $\sigma(x) \triangleright t$. Furthermore, there is a term $t' \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $r \triangleright_\mu t'$, $\sigma(t') = t$ and $\text{root}(t) = \text{root}(t') \in \mathcal{H}$.*

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 $\sigma(r) \triangleright_\mu t$, then there is a non-variable and non-replacing position $p \in \text{Pos}_{\mathcal{F}}(r) - \text{Pos}^\mu(r)$, such that $\text{root}(r|_p) = \text{root}(t) \in \mathcal{H}(\mathcal{R}, \mu)$ and $\sigma(r|_p) = t$. Then, we let $t' = r|_p$. \square

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 rooted by a hidden symbol.

Lemma 3.5 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let \mathcal{A} be a finite 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 $t_1 \not\triangleright t$ and $t_n \triangleright_\mu t$, then $\text{root}(t) \in \mathcal{H}$.*

Proof. By induction on n :

- (i) If $n = 1$, then it is vacuously true.
- (ii) If $n > 1$, then we assume that $t_1 \not\triangleright t$ and $t_n \triangleright_\mu t$. Let $l \rightarrow r \in R$ be such that $t_{n-1} = C[\sigma(l)]$ and $t_n = C[\sigma(r)]$ for some context $C[\]$. We consider two cases: either $t_{n-1} \triangleright_\mu t$ holds or not.
 - (a) If $t_{n-1} \triangleright_\mu t$, then by the induction hypothesis we have that $\text{root}(t) \in \mathcal{H}$.
 - (b) If $t_{n-1} \triangleright_\mu t$ does not hold, then one of the following cases holds:
 - (1) $t_{n-1} \triangleright_\mu t$; then $t_{n-1} \in \mathcal{M}_{\infty, \mu}$ implies that $t \notin \mathcal{M}_{\infty, \mu}$, leading to a contradiction.

- (2) $t_{n-1} \not\triangleright_\mu t$ (in particular, $\sigma(l) \not\triangleright_\mu t$); then, since $t_n \triangleright_\mu t$ there must be $\sigma(r) \triangleright_\mu t$. Thus, by Lemma 3.4 we conclude that $\text{root}(t) \in \mathcal{H}$. \square

Now we use the previous lemmas to investigate infinite sequences that mix 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 rules.

Proposition 3.6 *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS and $\mu \in M_{\mathcal{R}}$. Let \mathcal{A} be an infinite sequence of the form $t_1 \xrightarrow{\epsilon} s_1 \supseteq_\mu t'_2 \xrightarrow{\epsilon}^* t_2 \xrightarrow{\epsilon} s_2 \supseteq_\mu t'_3 \xrightarrow{\epsilon}^* t_3 \cdots$ 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 $t_i \triangleright_\mu t$ for some $i \geq 1$, then $\text{root}(t) \in \mathcal{H} \cap \mathcal{D}$ or $t_1 \triangleright_\mu t$.*

Proof. By induction on i :

- (i) If $i = 1$, it is trivial.
- (ii) If $i > 1$ and $t_i \triangleright_\mu t$, then we consider two cases: either $t_{i-1} \triangleright_\mu t$ holds or not.
 - (a) If $t_{i-1} \triangleright_\mu t$, then by the induction hypothesis we get $t_1 \triangleright_\mu t$ or $\text{root}(t) \in \mathcal{H} \cap \mathcal{D}$, as desired.
 - (b) If $t_{i-1} \triangleright_\mu t$ does not hold, then let $l \rightarrow r \in R$ be such that $t_{i-1} = \sigma(l)$ and $s_{i-1} = \sigma(r) \supseteq_\mu t'_i$. We consider two cases:
 - (1) if $t_{i-1} \triangleright_\mu t$ then being $t_{i-1} \in \mathcal{M}_{\infty, \mu}$ it would imply that $t \notin \mathcal{M}_{\infty, \mu}$, thus leading to a contradiction.
 - (2) If $t_{i-1} \not\triangleright_\mu t$, then we consider two cases: either $t'_i \triangleright_\mu t$ or $t'_i \not\triangleright_\mu t$.
 - (A) If $t'_i \triangleright_\mu t$, since $t'_i, t \in \mathcal{M}_{\infty, \mu}$ the case $t'_i \triangleright_\mu t$ is excluded and the only possibility is that $t'_i \triangleright_\mu t$. Then, since $\sigma(l) = t_{i-1} \not\triangleright_\mu t$ and $\sigma(r) \supseteq_\mu t'_i \triangleright_\mu t$, i.e. $\sigma(r) \triangleright_\mu t$, by Lemma 3.4 we conclude that $\text{root}(t) \in \mathcal{H}$. Since $t \in \mathcal{M}_{\infty, \mu}$, we have $\text{root}(t) \in \mathcal{H} \cap \mathcal{D}$.
 - (B) If $t'_i \not\triangleright_\mu t$, then, by applying Lemma 3.1 and Lemma 3.5 to the μ -rewrite sequence $t'_i \xrightarrow{\epsilon}^* t_i$ we conclude $\text{root}(t) \in \mathcal{H} \cap \mathcal{D}$.

\square

As an immediate consequence of Proposition 3.6, we have the following result which we will use later.

Corollary 3.7 *Let (\mathcal{R}, μ) be a CS-TRS, \mathcal{A} be an infinite sequence of the form $t_1 \xrightarrow{\epsilon} s_1 \supseteq_\mu t'_2 \xrightarrow{\epsilon}^* t_2 \xrightarrow{\epsilon} s_2 \supseteq_\mu t'_3 \xrightarrow{\epsilon}^* t_3 \cdots$ 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 $t_i \triangleright_\mu t$ for some $i \geq 1$ and $\text{root}(t) \in \mathcal{D} - \mathcal{H}$, then $t_1 \triangleright_\mu t$.*

4 Revised context-sensitive dependency graph

Proposition 3.2 motivates the definition of context-sensitive dependency pair(s) and chain of context-sensitive dependency pairs.

Definition 4.1 [CS-dependency pairs [3]] Let $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS and $\mu \in M_{\mathcal{R}}$. 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 (CS-DPs) where:

$$\text{DP}_{\mathcal{F}}(\mathcal{R}, \mu) = \{l^\sharp \rightarrow s^\sharp \mid l \rightarrow r \in R, r \succeq_\mu s, \text{root}(s) \in \mathcal{D}, l \not\preceq_\mu s\}$$

and $\text{DP}_{\mathcal{X}}(\mathcal{R}, \mu) = \{l^\sharp \rightarrow x \mid l \rightarrow r \in R, x \in \text{Var}^\mu(r) - \text{Var}^\mu(l)\}$. We extend $\mu \in M_{\mathcal{F}}$ into $\mu^\sharp \in M_{\mathcal{F}^\sharp}$ by $\mu^\sharp(f) = \mu(f)$ if $f \in \mathcal{F}$, and $\mu^\sharp(f^\sharp) = \mu(f)$ if $f \in \mathcal{D}$.

Example 4.2 Consider the CS-TRS (\mathcal{R}, μ) in Example 1.2. There are six context-sensitive dependency pairs:

```

1: PRIMES -> SIEVE(from(s(s(0))))
2: PRIMES -> FROM(s(s(0)))
3: TAIL(cons(X,Y)) -> Y
4: IF(true,X,Y) -> X
5: IF(false,X,Y) -> Y
6: FILTER(s(s(X)),cons(Y,Z)) ->
   IF(divides(s(s(X)),Y),filter(s(s(X)),Z),cons(Y,filter(X,sieve(Y))))

```

Note the three collapsing dependency pairs: (3), (4), and (5).

Definition 4.3 [Chain of CS-DPs [3]] Let (\mathcal{R}, μ) be a CS-TRS. Given $\mathcal{P} \subseteq \text{DP}(\mathcal{R}, \mu)$, an $(\mathcal{R}, \mathcal{P}, \mu^\sharp)$ -chain is a finite or infinite sequence of pairs $u_i \rightarrow v_i \in \mathcal{P}$, for $i \geq 1$ such that there is a substitution σ satisfying both:

- (i) $\sigma(v_i) \hookrightarrow_{\mathcal{R}, \mu^\sharp}^* \sigma(u_{i+1})$, if $u_i \rightarrow v_i \in \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$, and
- (ii) if $u_i \rightarrow v_i = u_i \rightarrow x_i \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$, then there is $s_i \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $\sigma(x_i) \succeq_\mu s_i$ and $s_i^\sharp \hookrightarrow_{\mathcal{R}, \mu^\sharp}^* \sigma(u_{i+1})$.

Here, as usual we assume that different occurrences of dependency pairs do not share any variable (renamings are used if necessary). An $(\mathcal{R}, \mathcal{P}, \mu^\sharp)$ -chain is called *minimal* if for all $i \geq 1$ $\sigma(u_i)^\sharp \in \mathcal{M}_{\infty, \mu}$, $s_i \in \mathcal{M}_{\infty, \mu}$ (whenever they occur in the chain) and all dependency pairs in \mathcal{P} occur infinitely often.

Remark 4.4 When an $(\mathcal{R}, \text{DP}(\mathcal{R}, \mu), \mu^\sharp)$ -chain is written for a given substitution σ , we write $\sigma(u) \hookrightarrow_{\text{DP}(\mathcal{R}, \mu), \mu^\sharp} \sigma(v)$ for steps which use a dependency pair $u \rightarrow v \in \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$ but we rather write $\sigma(u) \hookrightarrow_{\text{DP}(\mathcal{R}, \mu), \mu^\sharp} s^\sharp$ for steps which use a dependency pair $u \rightarrow x \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$, where s is as in Definition 4.3.

Theorem 4.5 (Correctness and completeness [3]) Let \mathcal{R} be a TRS and $\mu \in M_{\mathcal{R}}$. \mathcal{R} is μ -terminating if and only if there is no infinite $(\mathcal{R}, \text{DP}(\mathcal{R}, \mu), \mu^\sharp)$ -chain.

An essential aspect of the mechanization of the dependency pairs approach is the analysis of infinite sequences of dependency pairs by looking at (the cycles \mathfrak{C} of) the *dependency graph* associated to the TRS \mathcal{R} . In [3], the *context-sensitive dependency graph*, is defined as follows:

- (i) There is an arc from a dependency pair $u \rightarrow v \in \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$ to a dependency pair $u' \rightarrow v' \in \text{DP}(\mathcal{R}, \mu)$ if there is a substitutions σ such that $\sigma(v) \hookrightarrow_{\mathcal{R}, \mu^\sharp}^* \sigma(u')$.
- (ii) There is an arc from a dependency pair $u \rightarrow v \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$ to *each* dependency pair $u' \rightarrow v' \in \text{DP}(\mathcal{R}, \mu)$.

Connecting each collapsing dependency pair with *every* other dependency pair

makes the cycles bigger, thus making some proofs of termination harder. Thanks to the results in the previous section, we can prove the following.

Theorem 4.6 *There is no infinite minimal $(\mathcal{R}, \mathcal{P}, \mu^\sharp)$ -chain involving an infinite number of dependency pairs $u_i \rightarrow v_i \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$ such that $\text{root}(u_{i+1})^\sharp \notin \mathcal{H}$.*

Proof. By contradiction. Let A be an infinite $(\mathcal{R}, \mathcal{P}, \mu^\sharp)$ -minimal chain of CS-DPs characterized by the CS-DPs $u_i \rightarrow v_i$ for $i \geq 1$:

$$\sigma(u_1) \xrightarrow{\epsilon} \mathcal{P} s_1^\sharp \hookrightarrow_{\mathcal{R}}^* \sigma(u_2) \xrightarrow{\epsilon} \mathcal{P} s_2^\sharp \hookrightarrow_{\mathcal{R}}^* \dots$$

where, $s_i^\sharp = \sigma(v_i)$ if $u_i \rightarrow v_i \in \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$ and $\sigma(x_i) \triangleright_\mu s_i$ if $u_i \rightarrow v_i = u_i \rightarrow x_i \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$. Let I be the infinite set of indices satisfying that for all $i \in I$, $u_i \rightarrow v_i \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$ and $\text{root}(u_{i+1})^\sharp \notin \mathcal{H}$. Given $i \in I$, let $\eta(i)$ be the ‘next’ positive integer in I : $\eta(i) = \min(\{j \in I \mid j > i\})$. Obviously, for all $i \in I$, $\eta(i) \in I$. Now consider the following sequence A^\sharp which is obtained from A by ‘unsharpening’ the tuple symbols and using the rules $l_i \rightarrow r_i$ which originate the dependency pairs $u_i \rightarrow v_i$ which are used in A :

$$\sigma(u_1)^\sharp \xrightarrow{\epsilon} \mathcal{R} \sigma(r_1) \triangleright_\mu s_1 \hookrightarrow_{\mathcal{R}}^* \sigma(u_2)^\sharp \xrightarrow{\epsilon} \mathcal{R} \sigma(r_2) \triangleright_\mu s_2 \hookrightarrow_{\mathcal{R}}^* \dots$$

which corresponds to A above: by minimality of A (see Definition 4.3), we have that $\sigma(u_i)^\sharp, s_i \in \mathcal{M}_{\infty, \mu}$ for all $i \geq 1$. By definition of I , for all $i \in I$, $v_i = x_i \in \mathcal{X}$. By definition of collapsing dependency pair, $x_i \in \text{Pos}(u_i) - \text{Pos}^\mu(u_i)$ and $\sigma(x_i) \triangleright_\mu s_i$. Thus, $\sigma(u_i) \triangleright_\mu s_i$ for all $i \in I$. By repeatedly applying Corollary 3.7, we have that $\sigma(u_i) \triangleright_\mu \sigma(u_{\eta(i)})$, i.e., $\sigma(u_i) \triangleright \sigma(u_{\eta(i)})$ for all $i \in I$. Thus, we obtain an infinite \triangleright -sequence which contradicts well-foundedness of \triangleright . \square

As a consequence of this result, we can dismiss the arcs of the dependency graph which connect collapsing dependency pairs $u \rightarrow v$ and dependency pairs $u' \rightarrow v'$ such that $\text{root}(u')^\sharp \notin \mathcal{H}$. This leads to a new definition of the context-sensitive dependency graph:

Definition 4.7 [Context-Sensitive Dependency Graph] Let \mathcal{R} be a TRS and $\mu \in \mathcal{M}_{\mathcal{R}}$. The context-sensitive dependency graph consists of the set $\text{DP}(\mathcal{R}, \mu)$ of context-sensitive dependency pairs together with arcs which connect them as follows:

- (i) There is an arc from a dependency pair $u \rightarrow v \in \text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$ to a dependency pair $u' \rightarrow v' \in \text{DP}(\mathcal{R}, \mu)$ if there is a substitutions σ such that $\sigma(v) \hookrightarrow_{\mathcal{R}, \mu^\sharp}^* \sigma(u')$.
- (ii) There is an arc from a dependency pair $u \rightarrow v \in \text{DP}_{\mathcal{X}}(\mathcal{R}, \mu)$ to a dependency pair $u' \rightarrow v' \in \text{DP}(\mathcal{R}, \mu)$ if $\text{root}(u')^\sharp \in \mathcal{H}(\mathcal{R}, \mu)$.

Example 4.8 Consider again the TRS \mathcal{R} in Example 1.2. The hidden defined symbols are **filter**, **from** and **sieve**. The dependency graph which corresponds to this example is shown in Figure 1 (right). Note that, in contrast to the situation

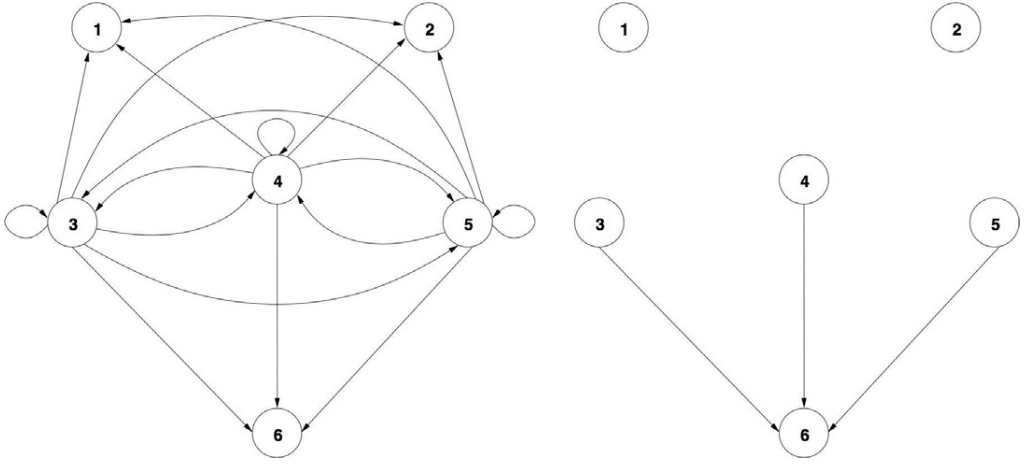


Fig. 1. Dependency graphs for Example 1.2: according to [3] (left) and according to Definition 4.7 (right)

with the old dependency graph (Figure 1, left) the new dependency graph has *no* cycle!

As noticed by Arts and Giesl, the presence of an infinite chain of dependency pairs corresponds to a cycle in the dependency graph (but not vice-versa). In the dependency graph this is true in the following sense: for each infinite chain of dependency pairs there is a suffix of the chain which corresponds to a cycle in the new dependency graph.

On the other hand, the treatment of cycles of the context-sensitive dependency graph for concluding termination by means of orderings remains as described in [3], but using the dependency graph in Definition 4.7.

Example 4.9 Consider the following TRS \mathcal{R} [16, Example 4]

```

f(X) -> cons(X, f(g(X)))
g(0) -> s(0)
g(s(X)) -> s(s(g(X)))
sel(0, cons(X, Y)) -> X
sel(s(X), cons(Y, Z)) -> sel(X, Z)

```

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

```

G(s(X)) -> G(X)
SEL(s(X), cons(Y, Z)) -> SEL(X, Z)
SEL(s(X), cons(Y, Z)) -> Z

```

The set of hidden symbols is $\mathcal{H} = \{\mathbf{f}, \mathbf{g}\}$ and there are two cycles:

- (i) $G(s(X)) \rightarrow G(X)$
- (ii) $\text{SEL}(s(X), \text{cons}(Y, Z)) \rightarrow \text{SEL}(X, Z)$

By using the subterm criterion [3, Section 5] we can easily prove that the system is μ -terminating.

5 Narrowing context-sensitive dependency pairs

There are examples where the automation of the CS-DP method can fail or be more difficult due to the *estimation* of the arcs that connect two CS-dependency pairs (by means of functions CAP^μ and REN^μ , see [3]).

Example 5.1 Consider the following example [13, Proposition 7]

$$\begin{aligned} f(0) &\rightarrow \text{cons}(0, f(s(0))) \\ f(s(0)) &\rightarrow f(p(s(0))) \\ p(s(X)) &\rightarrow X \end{aligned}$$

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

$$\begin{aligned} F(s(0)) &\rightarrow F(p(s(0))) \\ F(s(0)) &\rightarrow P(s(0)) \end{aligned}$$

The *estimated* CS-dependency graph contains one cycle consisting of the CS-dependency pair

$$F(s(0)) \rightarrow F(p(s(0)))$$

However, this cycle does *not* belong to the CS-dependency graph because there is no way to μ -rewrite $F(p(s(0)))$ into $F(s(0))$!

The problem is that with the estimated CS-dependency graph, we connect more dependency pairs than needed. The over-estimation eventually comes when a CS-dependency pair $u \rightarrow v$ is connected to $u' \rightarrow v'$ in the estimated dependency graph and v and u' do not unify, i.e. at least a rewriting step with some rule of \mathcal{R} is needed to reduce (some instance of) v to (the corresponding instance of) u' . It is then possible that, after performing such a necessary μ -rewriting step, the connection between them gets clearly lost, i.e. the nodes were not really connected in the graph. This is missed in the estimated dependency graph due to the use of CAP^μ and REN^μ . We can use *context-sensitive* narrowing to avoid this problem.

Definition 5.2 [Context-sensitive narrowing [10]] Let (\mathcal{R}, μ) be a CS-TRS. A term t μ -narrows to a term s (written $t \rightsquigarrow_\mu s$), if there exists a non-variable position $p \in \text{Pos}^\mu(t)$, θ is the most general unifier of $t|_p$ and l for a rewrite rule $l \rightarrow r$ in \mathcal{R} (sharing no variable with t), and $s = \theta(t[r]_p)$.

To achieve more precision when connecting two CS-DPs in a $(\mathcal{R}, DP(\mathcal{R}, \mu), \mu^\sharp)$ -chain, we may perform all possible μ -narrowings steps on v in order to develop the reductions from (instances of) v to (instances of) u' . Then, we obtain new terms v_1, \dots, v_n which are μ -narrowings of v with unifier θ_i for $i \in \{1, \dots, n\}$ and can be used instead of v . Not only the right-hand sides of the CS-dependency pairs are μ -narrowed: the unifier which used in the narrowing step should also be applied on the left-hand sides of the μ -narrowed pairs. Therefore, we can replace a CS-dependency pair $u \rightarrow v$ by all new μ -narrowed pairs $\theta_1(u) \rightarrow v_1, \dots, \theta_n(u) \rightarrow v_n$. The next result shows that under those conditions, the set of CS-dependency pairs can be replaced by their narrowings without losing correctness or completeness.

Theorem 5.3 (Narrowing refinement for CS-termination) *Let \mathcal{R} be a TRS and let \mathcal{P} be a set of CS-dependency pairs. Let $u \rightarrow v \in \mathcal{P}$ such that v is linear and for all $u' \rightarrow v' \in \mathcal{P}$ (with renamed variables) the terms v and u' are not unifiable. Let*

$$\mathcal{P}' = (\mathcal{P} - \{u \rightarrow v\}) \cup \{u' \rightarrow v' \mid u' \rightarrow v' \text{ is a narrowing of } u \rightarrow v\}.$$

There exists an infinite $(\mathcal{R}, \mathcal{P}, \mu^\sharp)$ -chain iff there exists an infinite $(\mathcal{R}, \mathcal{P}', \mu^\sharp)$ -chain.

Proof. The proof of this theorem corresponds to the proof of Theorem 25 in [1]. Note that only dependency pairs in $\text{DP}_{\mathcal{F}}(\mathcal{R}, \mu)$ can be narrowed. As in Arts and Giesl's proof, requiring the no-unification between the CS-dependency pair to narrow and the rest of the set; the linearity of v ; and the renaming of the variables of the different (occurrences of) dependency pairs is still necessary to guarantee that narrowing CS-dependency pairs do not miss any chain from \mathcal{P} . The main difference is that the reductions between dependency pairs are μ -reductions, but since we are using μ -narrowing, the whole proof is adapted without loss of generality. \square

Thus, after narrowing the dependency pairs in $\text{DP}(\mathcal{R}, \mu)$ we can build a *narrowed* dependency graph. Afterwards, we can use it to check termination as usual.

Example 5.4 (Continuing Example 5.1) Since the right-hand side of the CS-dependency pair in Example 5.1 does not unify with any left-hand side of a dependency pair, (including itself) and it can be μ -narrowed at position 1 (notice that $\mu(\mathbf{f}) = \{1\}$) by using the rule

$$p(s(X)) \rightarrow X$$

we can replace it by its μ -narrowed CS-dependency pair:

$$F(s(0)) \rightarrow F(0)$$

The *narrowed* pair does not form any cycle in the estimated narrowed graph and termination is easily proved now.

6 Experiments

The techniques described in the previous sections have been implemented as part of the tool MU-TERM [2,12]. We have used our new implementation to compare with the last version of the tool: MU-TERM 4.3. The benchmarks were executed in a completely automatic way (see [2] for a description of MU-TERM's termination expert) and with a timeout of 1 minute on the 90 examples in the Context-Sensitive Rewriting subcategory of the 2006 Termination Competition, available through the URL:

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

As remarked above, our termination expert works as explained in [2] for version 4.3 of MU-TERM. For the new version 4.4 of MU-TERM, we have just used the new definition of the (eventually narrowed) dependency graph. We have compared our new implementation with the previous version of MU-TERM (corresponding to [3]).

Termination Tool	Total	CS-DPs	CSRPO	Transf.	Average time
MU-TERM (PROLE'06)	66	65	0	1	1.68s
MU-TERM (FST&TCS'06)	56	45	7	4	4.55s
AProVE	56	0	0	56	4.74s

Table 1

We have also used AProVE for proving termination of the examples. AProVE [9] is currently the most powerful tool for proving termination of TRSs and implements most existing results and techniques regarding DPs and related techniques. AProVE is also able to prove termination of Context-Sensitive Rewriting by using *transformations*. Such transformations obtain a proof of the μ -termination of a TRS \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 (see [13] for a recent survey).

A complete account of our experiments can be found here:

<http://www.dsic.upv.es/~rgutierrez/muterm/prole/benchmarks.html>

Table 1 summarizes our benchmarks. As shown in Table 1, the results make clear the advantages of the new refinement: we are able to prove 10 additional examples and the proofs are almost *three* times faster (in the average).

Furthermore, we can say that the new refinement developed for the CS-DP approach greatly improves on the use of other techniques: the use of transformations and other (also powerful) techniques like CSRPO [4] becomes now anecdotic or null.

7 Conclusions

We have introduced a simplification of the context-sensitive dependency graph by restricting the outgoing links of collapsing dependency pairs to dependency pairs headed by the so-called *hidden* symbols. Hidden symbols are defined symbols that occur in non-replacing positions in the right-hand sides of some rule in the TRS. This greatly improves the performance of termination proofs based on the dependency graph proposed in [3]. Narrowing context-sensitive dependency pairs has also been investigated. It can also be helpful to simplify or restructure the dependency graph and eventually simplify the proof of termination. Regarding the practical use of the (refinements on the) new CS-dependency graph in proofs of termination of *CSR*, we have implemented these ideas as part of the termination tool MU-TERM and we have obtained quite good results in terms of new examples which could be proved, and also regarding the time for achieving the proofs.

Since the state-of-the-art of DP-based techniques for proving termination of *CSR* which has been introduced in this paper corresponds to the development of DPs in the late nineties, we can conclude that further improvements of CS-DPs will evolve in such a way that the CS-dependency pairs approach can play for *CSR* the (practical and theoretical) role than dependency pairs play in rewriting.

Many other aspects of the dependency pairs approach are also worth to be considered and extended to *CSR* (modularity issues, innermost computations, usable rules, . . .). They provide an interesting subject for future work.

References

- [1] T. Arts and J. Giesl. Termination of Term Rewriting Using Dependency Pairs *Theoretical Computer Science*, 236:133-178, 2000.
- [2] B. Alarcón, R. Gutiérrez, J. Iborra, and S. Lucas. Proving Termination of Context-Sensitive Rewriting with MU-TERM. *Electronic Notes in Theoretical Computer Science*, to appear, 2007.
- [3] B. Alarcón, R. Gutiérrez, and S. Lucas. Context-Sensitive Dependency Pairs. In N. Garg and S. Arun-Kumar, editors *Proc. of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS'06*, LNCS 4337:297-308, Springer-Verlag, Berlin, 2006.
- [4] C. Borralleras, S. Lucas, and A. Rubio. Recursive Path Orderings can be Context-Sensitive. In A. Voronkov, editor *Proc. of 18th International Conference on Automated Deduction, CADE'02*, LNAI 2392:314-331, Springer-Verlag, Berlin, 2002.
- [5] F. Durán, S. Lucas, J. Meseguer, C. Marché, and X. Urbain. Proving Termination of Membership Equational Programs. In P. Sestoft and N. Heintze, editors, *Proc. of PEPM'04*, pages 147-158, ACM Press, New York, 2004.
- [6] F. Durán, S. Lucas, J. Meseguer, C. Marché, and X. Urbain. Proving Operational Termination of Membership Equational Programs. *Higher-Order and Symbolic Computation*, to appear, 2006.
- [7] J. Giesl and A. Middeldorp. Transforming Context-Sensitive Rewrite Systems. In P. Narendran and M. Rusinowitch, editors, *Proc. of 10th International Conference on Rewriting Techniques and Applications, RTA'99*, LNCS 1631:271-285, Springer-Verlag, Berlin, 1999.
- [8] J. Giesl and A. Middeldorp. Transformation techniques for context-sensitive rewrite systems. *Journal of Functional Programming*, 14(4): 379-427, 2004.
- [9] J. Giesl, P. Schneider-Kamp, and R. Thiemann. AProVE 1.2: Automatic Termination Proofs in the Dependency Pair Framework. In U. Furbach and N. Shankar, editors, *Proc. of Third International Joint Conference on Automated Reasoning, IJCAR'06*, LNAI 4130:281-286, Springer-Verlag, Berlin, 2006. Available at <http://www-i2.informatik.rwth-aachen.de/AProVE>.
- [10] S. Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1):1-61, January 1998.
- [11] S. Lucas. Context-sensitive rewriting strategies. *Information and Computation*, 178(1):293-343, 2002.
- [12] S. Lucas. MU-TERM: A Tool for Proving Termination of Context-Sensitive Rewriting. In V. van Oostrom, editor, *Proc. of RTA'04*, LNCS 3091:200-209, Springer-Verlag, Berlin, 2004. Available at <http://www.dsic.upv.es/~slucas/csr/termination/muterm>.
- [13] S. Lucas. Proving termination of context-sensitive rewriting by transformation. *Information and Computation*, 204(12):1782-1846, 2006.
- [14] E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer-Verlag, Berlin, 2002.
- [15] TeReSe, editor, *Term Rewriting Systems*, Cambridge University Press, 2003.
- [16] H. Zantema. Termination of Context-Sensitive Rewriting. In H. Comon, editor, *Proc. of RTA'97*, LNCS 1232:172-186, Springer-Verlag, Berlin, 1997.