RWTH Aachen

Department of Computer Science
Technical Report

Improving Context-Sensitive Dependency Pairs

Beatriz Alarcón, Fabian Emmes, Carsten Fuhs, Jürgen Giesl, Raúl Gutiérrez, Salvador Lucas, Peter Schneider-Kamp, René Thiemann

ISSN 0935–3232 · Aachener Informatik Berichte · AIB-2008-13

RWTH Aachen · Department of Computer Science · Sept. 2008 (revised version)

The publications of the Department of Computer Science of *RWTH Aachen University* are in general accessible through the World Wide Web.

`http://aib.informatik.rwth-aachen.de/`

Improving Context-Sensitive Dependency Pairs*

Beatriz Alarcón[1], Fabian Emmes[2], Carsten Fuhs[2], Jürgen Giesl[2], Raúl Gutiérrez[1], Salvador Lucas[1], Peter Schneider-Kamp[2], and René Thiemann[3]

[1] DSIC, Universidad Politécnica de Valencia, Spain
[2] LuFG Informatik 2, RWTH Aachen University, Germany
[3] Institute of Computer Science, University of Innsbruck, Austria

Abstract. Context-sensitive dependency pairs (CS-DPs) are currently the most powerful method for automated termination analysis of context-sensitive rewriting. However, compared to DPs for ordinary rewriting, CS-DPs suffer from two main drawbacks: (a) CS-DPs can be *collapsing*. This complicates the handling of CS-DPs and makes them less powerful in practice. (b) There does not exist a "*DP framework*" for CS-DPs which would allow one to apply them in a flexible and modular way. This paper solves drawback (a) by introducing a new definition of CS-DPs. With our definition, CS-DPs are always non-collapsing and thus, they can be handled like ordinary DPs. This allows us to solve drawback (b) as well, i.e., we extend the existing DP framework for ordinary DPs to context-sensitive rewriting. We implemented our results in the tool AProVE and successfully evaluated them on a large collection of examples.

1 Introduction

Context-sensitive rewriting [22, 23] models evaluations in programming languages. It uses a *replacement map* μ with $\mu(f) \subseteq \{1, ..., \text{arity}(f)\}$ for every function symbol f to specify the argument positions of f where rewriting may take place.

Example 1. Consider this context-sensitive term rewrite system (CS-TRS)

$$\begin{array}{rlrl} \mathsf{gt}(\mathsf{0}, y) &\to \mathsf{false} & \mathsf{p}(\mathsf{0}) &\to \mathsf{0} \\ \mathsf{gt}(\mathsf{s}(x), \mathsf{0}) &\to \mathsf{true} & \mathsf{p}(\mathsf{s}(x)) &\to x \\ \mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y)) &\to \mathsf{gt}(x, y) & \mathsf{minus}(x, y) &\to \mathsf{if}(\mathsf{gt}(y, \mathsf{0}), \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y)), x) \\ \mathsf{if}(\mathsf{true}, x, y) &\to x & \mathsf{div}(\mathsf{0}, \mathsf{s}(y)) &\to \mathsf{0} \\ \mathsf{if}(\mathsf{false}, x, y) &\to y & \mathsf{div}(\mathsf{s}(x), \mathsf{s}(y)) &\to \mathsf{s}(\mathsf{div}(\mathsf{minus}(x, y), \mathsf{s}(y))) \end{array} \quad (1)$$

with $\mu(\mathsf{if}) = \{1\}$ and $\mu(f) = \{1, \ldots, \text{arity}(f)\}$ for all other symbols f to model the usual behavior of if*: in $\mathsf{if}(t_1, t_2, t_3)$, one may evaluate t_1, but not t_2 or t_3. It will turn out that due to μ, this CS-TRS is indeed terminating. In contrast, if one allows arbitrary reductions, then the TRS would be non-terminating:*

* Authors from Valencia were partially supported by the EU (FEDER) and the Spanish MEC/MICINN, under grants TIN 2007-68093-C02-02 and HA 2006-0007. B. Alarcón was partially supported by the Spanish MEC/MICINN under FPU grant AP2005-3399. R. Gutiérrez was partially supported by the Spanish MEC/MICINN, under grant TIN 2004-7943-C04-02. Authors from Aachen were supported by the DAAD under grant D/06/12785 and by the DFG under grant GI 274/5-2.

$$\mathsf{minus}(0,0) \to^+ \mathsf{if}(\mathsf{gt}(0,0), \mathsf{minus}(0,0), 0) \to^+ \mathsf{if}(..., \mathsf{if}(\mathsf{gt}(0,0), \mathsf{minus}(0,0), 0), ...) \to^+ ...$$

There are two approaches to prove termination of context-sensitive rewriting. The first approach transforms CS-TRSs to ordinary TRSs, cf. [12, 25]. But transformations often generate complicated TRSs where all termination tools fail.

Therefore, it is more promising to adapt existing termination techniques from ordinary term rewriting to the context-sensitive setting. Such adaptions were done for classical methods like RPO or polynomial orders [7, 18, 24]. However, much more powerful techniques like the *dependency pair* (DP) method [5] are implemented in almost all current termination tools for TRSs. But for a long time, it was not clear how to adapt the DP method to context-sensitive rewriting.

This was solved first in [1]. The corresponding implementation in the tool MU-TERM [3] outperformed all previous tools for termination of CS rewriting.

Nevertheless, the existing results on CS-DPs [1, 2, 4, 19] still have major disadvantages compared to the DP method for ordinary rewriting, since CS-DPs can be *collapsing*. To handle such DPs, one has to impose strong requirements which make the CS-DP method quite weak and which make it difficult to extend refined termination techniques based on DPs to the CS case. In particular, the *DP framework* [13, 16, 20], which is the most powerful formulation of the DP method for ordinary TRSs, has not yet been adapted to the CS setting.

In this paper, we solve these problems. After presenting preliminaries in Sect. 2, we introduce a new notion of *non-collapsing* CS-DPs in Sect. 3. This new notion makes it much easier to adapt termination techniques based on DPs to context-sensitive rewriting. Therefore, Sect. 4 extends the *DP framework* to the context-sensitive setting and shows that existing methods from this framework only need minor changes to apply them to context-sensitive rewriting.

All our results are implemented in the termination prover AProVE [15]. As shown by the empirical evaluation in Sect. 5, our contributions improve the power of automated termination analysis for context-sensitive rewriting substantially.

2 Context-Sensitive Rewriting and CS-Dependency Pairs

See [6] and [22] for basics on term rewriting and context-sensitive rewriting, respectively. Let $\mathcal{P}os(s)$ be the set of *positions* of a term s. For a replacement map μ, we define the *active positions* $\mathcal{P}os^\mu(s)$: For $x \in \mathcal{V}$ let $\mathcal{P}os^\mu(x) = \{\varepsilon\}$ where ε is the root position. Moreover, $\mathcal{P}os^\mu(f(s_1, \ldots, s_n)) = \{\varepsilon\} \cup \{i\,p \mid i \in \mu(f),\ p \in \mathcal{P}os^\mu(s_i)\}$. We say that $s \trianglerighteq_\mu t$ holds if $t = s|_p$ for some $p \in \mathcal{P}os^\mu(s)$ and $s \rhd_\mu t$ if $s \trianglerighteq_\mu t$ and $s \neq t$. Moreover, $s \rhd_{\not\mu} t$ if $t = s|_p$ for some $p \in \mathcal{P}os(s) \setminus \mathcal{P}os^\mu(s)$. We denote the ordinary subterm relations by $\trianglerighteq$ and $\rhd$.

A *CS-TRS* $(\mathcal{R}, \mu)$ consists of a finite TRS $\mathcal{R}$ and a replacement map μ. We have $s \hookrightarrow_{\mathcal{R},\mu} t$ iff there are $\ell \to r \in \mathcal{R}$, $p \in \mathcal{P}os^\mu(s)$, and a substitution σ with $s|_p = \sigma(\ell)$ and $t = s[\sigma(r)]_p$. This reduction is an *innermost* step (denoted $\stackrel{\mathrm{i}}{\hookrightarrow}_{\mathcal{R},\mu}$) if all t with $s|_p \rhd_\mu t$ are in normal form w.r.t. $(\mathcal{R}, \mu)$. A term s is in *normal form* w.r.t. $(\mathcal{R}, \mu)$ if there is no term t with $s \hookrightarrow_{\mathcal{R},\mu} t$. A CS-TRS $(\mathcal{R}, \mu)$ is *terminating* if $\hookrightarrow_{\mathcal{R},\mu}$ is well founded and *innermost terminating* if $\stackrel{\mathrm{i}}{\hookrightarrow}_{\mathcal{R},\mu}$ is well founded.

Let $\mathcal{D} = \{root(\ell) \mid \ell \to r \in \mathcal{R}\}$ be the set of *defined symbols*. For every $f \in \mathcal{D}$, let $f^\sharp$ be a fresh *tuple symbol* of same arity, where we often write "F" instead of "$f^\sharp$". For $t = f(t_1, \ldots, t_n)$ with $f \in \mathcal{D}$, let $t^\sharp = f^\sharp(t_1, \ldots, t_n)$.

Definition 2 (CS-DPs [1]). *Let $(\mathcal{R}, \mu)$ be a CS-TRS. If $\ell \to r \in \mathcal{R}$, $r \trianglerighteq_\mu t$, and* $\mathrm{root}(t) \in \mathcal{D}$, *then $\ell^\sharp \to t^\sharp$ is an* ordinary dependency pair.[4] *If $\ell \to r \in \mathcal{R}$, $r \trianglerighteq_\mu x$ for a variable x, and $\ell \not\trianglerighteq_\mu x$, then $\ell^\sharp \to x$ is a* collapsing DP. *Let $\mathsf{DP}_o(\mathcal{R}, \mu)$ and $\mathsf{DP}_c(\mathcal{R}, \mu)$ be the sets of all ordinary resp. all collapsing DPs.*

Example 3. For the TRS of Ex. 1, we obtain the following CS-DPs.

$$\mathsf{GT}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{GT}(x, y) \quad (2) \qquad \mathsf{M}(x, y) \to \mathsf{IF}(\mathsf{gt}(y, \mathsf{0}), \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y)), x) \quad (5)$$
$$\mathsf{IF}(\mathsf{true}, x, y) \to x \quad (3) \qquad \mathsf{M}(x, y) \to \mathsf{GT}(y, \mathsf{0}) \quad (6)$$
$$\mathsf{IF}(\mathsf{false}, x, y) \to y \quad (4) \qquad \mathsf{D}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{D}(\mathsf{minus}(x, y), \mathsf{s}(y)) \quad (7)$$
$$\mathsf{D}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{M}(x, y) \quad (8)$$

To prove termination, one has to show that there is no infinite *chain* of DPs. For ordinary rewriting, a sequence $s_1 \to t_1, s_2 \to t_2, \ldots$ of DPs is a *chain* if there is a substitution σ such that $t_i\sigma$ reduces to $s_{i+1}\sigma$.[5] If all $t_i\sigma$ are terminating, then the chain is *minimal* [13, 16, 21]. But due to the collapsing DPs, the notion of "chains" has to be adapted when it is used with CS-DPs [1]. If $s_i \to t_i$ is a collapsing DP (i.e., if $t_i \in \mathcal{V}$), then instead of $t_i\sigma \hookrightarrow^*_{\mathcal{R},\mu} s_{i+1}\sigma$ (and termination of $t_i\sigma$ for minimality), one requires that there is a term w_i with $t_i\sigma \trianglerighteq_\mu w_i$ and $w_i^\sharp \hookrightarrow^*_{\mathcal{R},\mu} s_{i+1}\sigma$. For minimal chains, $w_i^\sharp$ must be terminating.

Example 4. Ex. 1 has the chain (5), (3), (5) as $\mathsf{IF}(\mathsf{gt}(\mathsf{s}(y), \mathsf{0}), \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(\mathsf{s}(y))), x)$ $\hookrightarrow^*_{\mathcal{R},\mu}$ $\mathsf{IF}(\mathsf{true}, \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(\mathsf{s}(y))), x)$ $\hookrightarrow_{(3),\mu}$ $\mathsf{minus}(\mathsf{p}(x), \mathsf{p}(\mathsf{s}(y)))$ *and* $(\mathsf{minus}(\mathsf{p}(x), \mathsf{p}(\mathsf{s}(y))))^\sharp = \mathsf{M}(\mathsf{p}(x), \mathsf{p}(\mathsf{s}(y)))$ *is an instance of the left-hand side of (5).*

A CS-TRS is terminating iff there is no infinite chain [1]. As in the non-CS case, the above notion of chains can also be adapted to *innermost* rewriting. Then a CS-TRS is innermost terminating iff there is no infinite innermost chain [4].

Due to the collapsing CS-DPs (and the corresponding definition of "chains"), it is not easy to extend existing techniques for proving absence of infinite chains to CS-DPs. Therefore, we now introduce a new improved definition of CS-DPs.

3 Non-Collapsing CS-Dependency Pairs

Ordinary DPs only consider active subterms of right-hand sides. So Rule (1) of Ex. 1 only leads to the DP (5), but not to $\mathsf{M}(x, y) \to \mathsf{M}(\mathsf{p}(x), \mathsf{p}(y))$. However, the inactive subterm $\mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y))$ of the right-hand side of (1) may become active again when applying the rule $\mathsf{if}(\mathsf{true}, x, y) \to x$. Therefore, Def. 2 creates a collapsing DP like (3) whenever a rule $\ell \to r$ has a *migrating variable* x with $r \trianglerighteq_\mu x$, but $\ell \not\trianglerighteq_\mu x$. Indeed, when instantiating the *collapse-variable* x in (3) with an instance of the "hidden term" $\mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y))$, one obtains a chain which simulates the rewrite sequence from $\mathsf{minus}(t_1, t_2)$ over $\mathsf{if}(..., \mathsf{minus}(\mathsf{p}(t_1), \mathsf{p}(t_2)), ...)$

[4] A refinement is to eliminate DPs where $\ell \triangleright_\mu t$, cf. [1, 8].

[5] We always assume that different occurrences of DPs are variable-disjoint and consider substitutions whose domains may be infinite.

to $\mathsf{minus}(\mathsf{p}(t_1), \mathsf{p}(t_2))$, cf. Ex. 4. Our main observation is that collapsing DPs are only needed for certain instantiations of the variables. One might be tempted to allow only instantiations of collapse-variables by *hidden terms*.[6]

Definition 5 (Hidden Term). *Let $(\mathcal{R}, \mu)$ be a CS-TRS. We say that t is a* hidden term *if $root(t) \in \mathcal{D}$ and if there exists a rule $\ell \to r \in \mathcal{R}$ with $r \rhd_{\not\mu} t$.*

In Ex. 1, the only hidden term is $\mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y))$. But unfortunately, only allowing instantiations of collapse-variables with hidden terms would be unsound.

Example 6. Consider $\mu(\mathsf{g}) = \{1\}$, $\mu(\mathsf{a}) = \mu(\mathsf{b}) = \mu(\mathsf{f}) = \mu(\mathsf{h}) = \varnothing$ and the rules

$$\begin{array}{rlcrl} \mathsf{a} &\to \mathsf{f}(\mathsf{g}(\mathsf{b})) & (9) & \mathsf{h}(x) &\to x \\ \mathsf{f}(x) &\to \mathsf{h}(x) & & \mathsf{b} &\to \mathsf{a} \end{array}$$

The CS-TRS has the following infinite rewrite sequence:

$$\mathsf{a} \hookrightarrow_{\mathcal{R},\mu} \mathsf{f}(\mathsf{g}(\mathsf{b})) \hookrightarrow_{\mathcal{R},\mu} \underline{\mathsf{h}(\mathsf{g}(\mathsf{b})) \hookrightarrow_{\mathcal{R},\mu} \mathsf{g}(\mathsf{b})} \hookrightarrow_{\mathcal{R},\mu} \mathsf{g}(\mathsf{a}) \hookrightarrow_{\mathcal{R},\mu} \dots$$

We obtain the following CS-DPs according to Def. 2:

$$\begin{array}{rlrlc} \mathsf{A} &\to \mathsf{F}(\mathsf{g}(\mathsf{b})) & \mathsf{H}(x) &\to x & (10) \\ \mathsf{F}(x) &\to \mathsf{H}(x) & \mathsf{B} &\to \mathsf{A} & \end{array}$$

The only hidden term is b, *obtained from Rule (9). There is also an infinite chain that corresponds to the infinite reduction above. However, here the collapse-variable x in the DP (10) must be instantiated by* $\mathsf{g}(\mathsf{b})$ *and not by the hidden term* b, *cf. the underlined part above. So if one replaced (10) by* $\mathsf{H}(\mathsf{b}) \to \mathsf{b}$, *there would be no infinite chain anymore and one would falsely conclude termination.*

The problem in Ex. 6 is that rewrite rules may add additional symbols like g *above* hidden terms. This can happen if a term $\mathsf{g}(t)$ occurs at an inactive position in a right-hand side and if an instantiation of t could possibly reduce to a term containing a hidden term (i.e., if t has a defined symbol or a variable at an active position). Then we call $\mathsf{g}(\Box)$ a *hiding context*, since it can "hide" a hidden term. Moreover, the composition of hiding contexts is again a hiding context.

Definition 7 (Hiding Context). *Let $(\mathcal{R}, \mu)$ be a CS-TRS. The function symbol f* hides position i *if there is a rule $\ell \to r \in \mathcal{R}$ with $r \rhd_{\not\mu} f(r_1, \dots, r_i, \dots, r_n)$, $i \in \mu(f)$, and r_i contains a defined symbol or a variable at an active position. A context C is* hiding *iff $C = \Box$ or C has the form $f(t_1, \dots, t_{i-1}, C', t_{i+1}, \dots, t_n)$ where f hides position i and C' is a hiding context.*

Example 8. In Ex. 6, g *hides position 1 due to Rule (9). So the hiding contexts are $\Box, \mathsf{g}(\Box), \mathsf{g}(\mathsf{g}(\Box)), \dots$ In the TRS of Ex. 1,* minus *hides both positions 1 and 2 and* p *hides position 1 due to Rule (1). So the hiding contexts are* $\Box, \mathsf{p}(\Box), \mathsf{minus}(\Box, \Box), \mathsf{p}(\mathsf{p}(\Box)), \mathsf{minus}(\Box, \mathsf{p}(\Box)), \dots$

To remove collapsing DPs $s \to x$, we now restrict ourselves to instantiations of x with terms of the form $C[t]$ where C is a hiding context and t is a hidden term. So in Ex. 6, the variable x in the DP (10) should only be instantiated by

[6] A similar notion of *hidden symbols* was presented in [2, 4], but there one only used these symbols to improve one special termination technique (the *dependency graph*).

b, g(b), g(g(b)), etc. To represent these infinitely many instantiations in a finite way, we replace $s \to x$ by new *unhiding* DPs (which "unhide" hidden terms).

Definition 9 (Improved CS-DPs). *For a CS-TRS* $(\mathcal{R},\mu)$*, if* $\mathsf{DP}_c(\mathcal{R},\mu) \neq \varnothing$*, we introduce a fresh*[7] unhiding tuple symbol U *and the following* unhiding DPs*:*

- $s \to \mathsf{U}(x)$ *for every* $s \to x \in \mathsf{DP}_c(\mathcal{R},\mu)$*,*
- $\mathsf{U}(f(x_1,\ldots,x_i,\ldots,x_n)) \to \mathsf{U}(x_i)$ *for every function symbol* f *of any arity* n *and every* $1 \leq i \leq n$ *where* f *hides position* i*, and*
- $\mathsf{U}(t) \to t^\sharp$ *for every hidden term* t*.*

Let $\mathsf{DP}_u(\mathcal{R},\mu)$ *be the set of all unhiding DPs (where* $\mathsf{DP}_u(\mathcal{R},\mu) = \varnothing$*, if* $\mathsf{DP}_c(\mathcal{R},\mu) = \varnothing$*). Then the set of* improved CS-DPs *is* $\mathsf{DP}(\mathcal{R},\mu) = \mathsf{DP}_o(\mathcal{R},\mu) \cup \mathsf{DP}_u(\mathcal{R},\mu)$*.*

Example 10. In Ex. 6, instead of (10) we get the unhiding DPs

$$\mathsf{H}(x) \to \mathsf{U}(x), \qquad \mathsf{U}(\mathsf{g}(x)) \to \mathsf{U}(x), \qquad \mathsf{U}(\mathsf{b}) \to \mathsf{B}.$$

Now there is indeed an infinite chain. In Ex. 1, instead of (3) and (4), we obtain:[8]

$$\mathsf{IF}(\mathsf{true},x,y) \to \mathsf{U}(x) \quad (11) \qquad \mathsf{U}(\mathsf{p}(x)) \to \mathsf{U}(x) \quad (15)$$
$$\mathsf{IF}(\mathsf{false},x,y) \to \mathsf{U}(y) \quad (12) \qquad \mathsf{U}(\mathsf{minus}(x,y)) \to \mathsf{U}(x) \quad (16)$$
$$\mathsf{U}(\mathsf{minus}(\mathsf{p}(x),\mathsf{p}(y))) \to \mathsf{M}(\mathsf{p}(x),\mathsf{p}(y)) \quad (13) \qquad \mathsf{U}(\mathsf{minus}(x,y)) \to \mathsf{U}(y) \quad (17)$$
$$\mathsf{U}(\mathsf{p}(x)) \to \mathsf{P}(x) \quad (14)$$

Clearly, the improved CS-DPs are never collapsing. Thus, now the definition of (minimal)[9] *chains* is completely analogous to the one for ordinary rewriting.

Definition 11 (Chain). *Let* $\mathcal{P}$ *and* $\mathcal{R}$ *be TRSs and let* μ *be a replacement map. We extend* μ *to tuple symbols by defining* $\mu(f^\sharp) = \mu(f)$ *for all* $f \in \mathcal{D}$ *and* $\mu(\mathsf{U}) = \varnothing$*.*[10] *A sequence of pairs* $s_1 \to t_1, s_2 \to t_2, \ldots$ *from* $\mathcal{P}$ *is a* $(\mathcal{P},\mathcal{R},\mu)$-chain *iff there is a substitution* σ *with* $t_i\sigma \hookrightarrow^*_{\mathcal{R},\mu} s_{i+1}\sigma$ *and* $t_i\sigma$ *is terminating w.r.t.* $(\mathcal{R},\mu)$ *for all* i*. It is an* innermost $(\mathcal{P},\mathcal{R},\mu)$-chain *iff* $t_i\sigma \overset{\mathrm{i}}{\hookrightarrow}^*_{\mathcal{R},\mu} s_{i+1}\sigma$*,* $s_i\sigma$ *is in normal form, and* $t_i\sigma$ *is innermost terminating w.r.t.* $(\mathcal{R},\mu)$ *for all* i*.*

Our main theorem shows that improved CS-DPs are still sound and complete.

Theorem 12 (Soundness and Completeness of Improved CS-DPs). *A CS-TRS* $(\mathcal{R},\mu)$ *is terminating iff there is no infinite* $(\mathsf{DP}(\mathcal{R},\mu),\mathcal{R},\mu)$*-chain and innermost terminating iff there is no infinite innermost* $(\mathsf{DP}(\mathcal{R},\mu),\mathcal{R},\mu)$*-chain.*

Proof. We only prove the theorem for "full" termination. The proof for innermost termination is very similar and can be found in Appendix A.

<u>**Soundness**</u>

$\mathcal{M}_{\infty,\mu}$ contains all *minimal non-terminating terms*: $t \in \mathcal{M}_{\infty,\mu}$ iff t is non-termi-

[7] Alternatively, one could also use different U-symbols for different collapsing DPs.

[8] We omitted the DP $\mathsf{U}(\mathsf{p}(y)) \to \mathsf{P}(y)$ that is "identical" to (14).

[9] Since we only regard *minimal* chains in the following, we included the "minimality requirement" in Def. 11, i.e., we require that all $t_i\sigma$ are (innermost) terminating. As in the DP framework for ordinary rewriting, this restriction to minimal chains is needed for several DP processors (e.g., for the reduction pair processor of Thm. 21).

[10] We define $\mu(\mathsf{U}) = \varnothing$, since the purpose of U is only to remove context around hidden terms. But during this removal, U's argument should not be evaluated.

nating and every r with $t \rhd_\mu r$ terminates. A term u has the *hiding property* iff

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

We first prove the following claim:

$$\text{Let } u \text{ be a term with the hiding property and let } u \hookrightarrow_{\mathcal{R},\mu} v \trianglerighteq_\mu w \text{ with } w \in \mathcal{M}_{\infty,\mu}. \text{ Then } w \text{ also has the hiding property.} \tag{18}$$

Let $w \rhd_{\not\mu} s \trianglerighteq_\mu t'$ for some terms s and t' with $t' \in \mathcal{M}_{\infty,\mu}$. Clearly, this also implies $v \rhd_{\not\mu} s$. If already $u \rhd s$, then we must have $u \rhd_{\not\mu} s$ due to the minimality of u. Thus, t' is an instance of a hidden term and $s = C[t']$ for a hiding context C, since u has the hiding property. Otherwise, $u \not\rhd s$. There must be a rule $\ell \to r \in \mathcal{R}$, an active context D (i.e., a context where the hole is at an active position), and a substitution δ such that $u = D[\delta(\ell)]$ and $v = D[\delta(r)]$. Clearly, $u \not\rhd s$ implies $\delta(\ell) \not\rhd s$ and $D \not\rhd s$. Hence, $v \rhd_{\not\mu} s$ means $\delta(r) \rhd_{\not\mu} s$. (The root of s cannot be above $\Box$ in D since those positions would be active.) Note that s cannot be at or below a variable position of r, because this would imply $\delta(\ell) \rhd s$. Thus, s is an instance of a non-variable subterm of r that is at an inactive position. So there is a $r' \notin \mathcal{V}$ with $r \rhd_{\not\mu} r'$ and $s = \delta(r')$. Recall that $s \trianglerighteq_\mu t'$, i.e., there is a $p \in \mathcal{P}os^\mu(s)$ with $s|_p = t'$. If p is a non-variable position of r', then $\delta(r'|_p) = t'$ and $r'|_p$ is a subterm with defined root at an active position (since $t' \in \mathcal{M}_{\infty,\mu}$ implies $\mathrm{root}(t') \in \mathcal{D}$). Hence, $r'|_p$ is a hidden term and thus, t' is an instance of a hidden term. Moreover, any instance of the context $C' = r'[\Box]_p$ is hiding. So if we define C to be $\delta(C')$, then $s = \delta(r') = \delta(r')[t']_p = \delta(C')[t'] = C[t']$ for the hiding context C. On the contrary, if p is not a non-variable position of r', then $p = p_1\, p_2$ where $r'|_{p_1}$ is a variable x. Now t' is an active subterm of $\delta(x)$ (more precisely, $\delta(x)|_{p_2} = t'$). Since x also occurs in ℓ, we have $\delta(\ell) \rhd \delta(x)$ and thus $u \rhd \delta(x)$. Due to the minimality of u this implies $u \rhd_{\not\mu} \delta(x)$. Since $u \rhd_{\not\mu} \delta(x) \trianglerighteq_\mu t'$, the hiding property of u implies that t' is an instance of a hidden term and that $\delta(x) = \overline{C}[t']$ for a hiding context $\overline{C}$. Note that since $r'|_{p_1}$ is a variable, the context C' around this variable is also hiding (i.e., $C' = r'[\Box]_{p_1}$). Thus, the context $C = \delta(C')[\overline{C}]$ is hiding as well and $s = \delta(r') = \delta(r')[\delta(x)[t']_{p_2}]_{p_1} = \delta(C')[\overline{C}[t']] = C[t']$.

Proof of Thm. 12 using Claim (18)

If $\mathcal{R}$ is not terminating, then there is a $t \in \mathcal{M}_{\infty,\mu}$ that is minimal w.r.t. $\trianglerighteq$. So there are t, t_i, s_i, t'_{i+1} such that

$$t \xhookrightarrow{>\varepsilon}{}^*_{\mathcal{R},\mu}\; t_1 \xrightarrow{\varepsilon}_{\mathcal{R}} s_1 \trianglerighteq_\mu t'_2 \xhookrightarrow{>\varepsilon}{}^*_{\mathcal{R},\mu}\; t_2 \xrightarrow{\varepsilon}_{\mathcal{R}} s_2 \trianglerighteq_\mu t'_3 \xhookrightarrow{>\varepsilon}{}^*_{\mathcal{R},\mu}\; t_3 \ldots \tag{19}$$

where $t_i, t'_i \in \mathcal{M}_{\infty,\mu}$ and all proper subterms of t (also at *inactive* positions) terminate. Here, "ε" (resp. "$> \varepsilon$") denotes reductions at (resp. strictly below) the root.

Note that (18) implies that all t_i have the hiding property. To see this, we use induction on i. Since t trivially has the hiding property (as it has no non-terminating proper subterms) and all terms in the reduction $t \xhookrightarrow{>\varepsilon}{}^*_{\mathcal{R},\mu}\; t_1$ are from $\mathcal{M}_{\infty,\mu}$ (as both $t, t_1 \in \mathcal{M}_{\infty,\mu}$), we conclude that t_1 also has the hiding property by applying (18) repeatedly. In the induction step, if t_{i-1} has the hiding property, then one application of (18) shows that t'_i also has the hiding property. By applying (18) repeatedly, one then also shows that t_i has the hiding property.

Now we show that $t_i^\sharp \to^+_{\mathsf{DP}(\mathcal{R},\mu)} {t'_{i+1}}^\sharp$ and that all terms in the reduction $t_i^\sharp \to^+_{\mathsf{DP}(\mathcal{R},\mu)} {t'_{i+1}}^\sharp$ terminate w.r.t. $(\mathcal{R},\mu)$. As ${t'_{i+1}}^\sharp \overset{>\varepsilon}{\hookrightarrow}{}^*_{\mathcal{R},\mu} t_{i+1}^\sharp$, we get an infinite $(\mathsf{DP}(\mathcal{R},\mu),\mathcal{R},\mu)$-chain.

From (19) we know that there are $\ell_i \to r_i \in \mathcal{R}$ and $p_i \in \mathcal{P}os^\mu(s_i)$ with $t_i = \ell_i\sigma$, $s_i = r_i\sigma$, and $s_i|_{p_i} = r_i\sigma|_{p_i} = t'_{i+1}$ for all i. First let $p_i \in \mathcal{P}os(r_i)$ with $r_i|_{p_i} \notin \mathcal{V}$. Then $\ell_i^\sharp \to (r_i|_{p_i})^\sharp \in \mathsf{DP}_o(\mathcal{R},\mu)$ and $t_i^\sharp = \ell_i^\sharp\sigma \to_{\mathsf{DP}_o(\mathcal{R},\mu)} (r_i|_{p_i})^\sharp\sigma = {t'_{i+1}}^\sharp$. Moreover, as $t_i, t'_{i+1} \in \mathcal{M}_{\infty,\mu}$, the terms $t_i^\sharp$ and ${t'_{i+1}}^\sharp$ are terminating.

Now let p_i be at or below the position of a variable x_i in r_i. By minimality of t_i, x_i only occurs at inactive positions of ℓ_i. Thus, $\ell_i^\sharp \to \mathsf{U}(x_i) \in \mathsf{DP}_u(\mathcal{R},\mu)$ and $r_i = C_i[x_i]$ where C_i is an active context. Recall that $t_i = \ell_i\sigma$ has the hiding property and that $t_i \rhd_{\not\mu} \sigma(x_i) \unrhd_\mu t'_{i+1}$. Thus, we have $\sigma(x_i) = C[t'_{i+1}]$ for a hiding context C and moreover, t'_{i+1} is an instance of a hidden term. Hence we obtain:

$$\begin{array}{lll}
t_i^\sharp = & \sigma(\ell_i^\sharp) & \\
\to_{\mathsf{DP}_u(\mathcal{R},\mu)} & \mathsf{U}(\sigma(x_i)) & \text{since } \ell_i^\sharp \to \mathsf{U}(x_i) \in \mathsf{DP}_u(\mathcal{R},\mu) \\
= & \mathsf{U}(C[t'_{i+1}]) & \text{for a hiding context } C \\
\to^*_{\mathsf{DP}_u(\mathcal{R},\mu)} & \mathsf{U}(t'_{i+1}) & \text{since } \mathsf{U}(C[x]) \to^*_{\mathsf{DP}_u(\mathcal{R},\mu)} \mathsf{U}(x) \text{ for any hiding context } C \\
\to_{\mathsf{DP}_u(\mathcal{R},\mu)} & {t'_{i+1}}^\sharp & \text{since } t'_{i+1} \text{ is an instance of a hidden term and} \\
 & & \mathsf{U}(t) \to_{\mathsf{DP}_u(\mathcal{R},\mu)} t^\sharp \text{ for any instance } t \text{ of a hidden term}
\end{array}$$

All terms in the reduction above are terminating. The reason is that again $t_i, t'_{i+1} \in \mathcal{M}_{\infty,\mu}$ implies that $t_i^\sharp$ and ${t'_{i+1}}^\sharp$ are terminating. Moreover, all terms $\mathsf{U}(\ldots)$ are normal forms since $\mu(\mathsf{U}) = \varnothing$ and since U does not occur in $\mathcal{R}$.

Completeness

Let there be an infinite chain $v_1 \to w_1, v_2 \to w_2, ...$ of improved CS-DPs. First, let the chain have an infinite tail consisting only of DPs of the form $\mathsf{U}(f(x_1,...,x_i,...,x_n)) \to \mathsf{U}(x_i)$. Since $\mu(\mathsf{U}) = \varnothing$, there are terms t_i with $\mathsf{U}(t_1) \xrightarrow{\varepsilon}_{\mathsf{DP}(R,\mu)} \mathsf{U}(t_2) \xrightarrow{\varepsilon}_{\mathsf{DP}(R,\mu)} ...$ Hence, $t_1 \rhd_\mu t_2 \rhd_\mu ..$ which contradicts the well-foundedness of $\rhd_\mu$.

Now we regard the remaining case. Here the chain has infinitely many DPs $v \to w$ with $v = \ell^\sharp$ for a rule $\ell \to r \in \mathcal{R}$. Let $v_i \to w_i$ be such a DP and let $v_j \to w_j$ with $j > i$ be the *next* such DP in the chain. Let σ be the substitution used for the chain. We show that then $v_i^\flat\sigma \hookrightarrow^*_{\mathcal{R},\mu} C[v_j^\flat\sigma]$ for an active context C. Here, $(f^\sharp(t_1,\ldots,t_n))^\flat = f(t_1,\ldots,t_n)$ for all $f \in \mathcal{D}$. Doing this for all such DPs implies that there is an infinite reduction w.r.t. $(\mathcal{R},\mu)$.

If $v_i \to w_i \in \mathsf{DP}_o(R,\mu)$ then the claim is trivial, because then $j = i+1$ and $v_i^\flat\sigma \hookrightarrow_{\mathcal{R},\mu} C[w_i^\flat\sigma] \hookrightarrow^*_{\mathcal{R},\mu} C[v_{i+1}^\flat\sigma]$ for some active context C.

Otherwise, $v_i \to w_i$ has the form $v_i \to \mathsf{U}(x)$. Then $v_i^\flat\sigma \hookrightarrow_{\mathcal{R},\mu} C_1[\sigma(x)]$ for an active context C_1. Moreover, $\mathsf{U}(\sigma(x))$ reduces to $\mathsf{U}(\delta(t))$ for a hidden term t and a δ by removing hiding contexts. Since hiding contexts are active, $\sigma(x) = C_2[\delta(t)]$ for an active context C_2. Finally, $t^\sharp\delta \overset{>\varepsilon}{\hookrightarrow}{}^*_{\mathcal{R},\mu} v_j\sigma$ and thus, $t\delta \overset{>\varepsilon}{\hookrightarrow}{}^*_{\mathcal{R},\mu} v_j^\flat\sigma$. By defining $C = C_1[C_2]$, we get $v_i^\flat\sigma \hookrightarrow^+_{\mathcal{R},\mu} C[v_j^\flat\sigma]$. □

4 CS Dependency Pair Framework

By Thm. 12, (innermost) termination of a CS-TRS is equivalent to absence of infinite (innermost) chains. For ordinary rewriting, the *DP framework* is the most recent and powerful collection of methods to prove absence of infinite chains automatically. Due to our new notion of (non-collapsing) CS-DPs, adapting the DP framework to the context-sensitive case now becomes much easier.[11]

In the DP framework, termination techniques operate on *DP problems* instead of TRSs. Def. 13 adapts this notion to context-sensitive rewriting.

Definition 13 (CS-DP Problem and Processor). *A* CS-DP problem *is a tuple* $(\mathcal{P}, \mathcal{R}, \mu, e)$*, where* $\mathcal{P}$ *and* $\mathcal{R}$ *are TRSs,* μ *is a replacement map, and* $e \in \{\mathbf{t}, \mathbf{i}\}$ *is a flag that stands for* **t**ermination *or* **i**nnermost termination*. We also call* $(\mathcal{P}, \mathcal{R}, \mu)$*-chains "*$(\mathcal{P}, \mathcal{R}, \mu, \mathbf{t})$*-chains" and we call innermost* $(\mathcal{P}, \mathcal{R}, \mu)$*-chains "*$(\mathcal{P}, \mathcal{R}, \mu, \mathbf{i})$*-chains". A* CS-DP problem $(\mathcal{P}, \mathcal{R}, \mu, e)$ *is* finite *if there is no infinite* $(\mathcal{P}, \mathcal{R}, \mu, e)$*-chain.*

A CS-DP processor *is a function Proc that takes a CS-DP problem as input and returns a possibly empty set of CS-DP problems. The processor Proc is* sound *if a CS-DP problem d is finite whenever all problems in Proc(d) are finite.*

For a CS-TRS $(\mathcal{R}, \mu)$, the termination proof starts with the *initial DP problem* $(\mathsf{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu, e)$ where e depends on whether one wants to prove termination or innermost termination. Then sound DP processors are applied repeatedly. If the final processors return empty sets, then (innermost) termination is proved. Since innermost termination is usually easier to show than full termination, one should use $e = \mathbf{i}$ whenever possible. As shown in [11], termination and innermost termination coincide for CS-TRSs $(\mathcal{R}, \mu)$ where $\mathcal{R}$ is *orthogonal* (i.e., left-linear and without critical pairs). So $(DP(\mathcal{R}, \mu), \mathcal{R}, \mu, \mathbf{i})$ would be the initial DP problem for Ex. 1, even when proving full termination. In Sect. 4.1 - 4.3, we recapitulate 3 important DP processors and extend them to context-sensitive rewriting.

4.1 Dependency Graph Processor

The first processor decomposes a DP problem into several sub-problems. To this end, one determines which pairs can follow each other in chains by constructing a *dependency graph*. In contrast to related definitions for collapsing CS-DPs in [1, 4], Def. 14 is analogous to the corresponding definition for non-CS rewriting.

Definition 14 (CS-Dependency Graph). *For a CS-DP problem* $(\mathcal{P}, \mathcal{R}, \mu, e)$*, the nodes of the* $(\mathcal{P}, \mathcal{R}, \mu, e)$-dependency graph *are the pairs of* $\mathcal{P}$*, and there is an arc from* $v \to w$ *to* $s \to t$ *iff* $v \to w, s \to t$ *is a* $(\mathcal{P}, \mathcal{R}, \mu, e)$*-chain.*

Example 15. Fig. 1 shows the dependency graph for Ex. 1, for both $e \in \{\mathbf{t}, \mathbf{i}\}$.[12]

[11] The proofs for this section can be found in Appendices B - D.

[12] To improve readability, we omitted nodes (6) and (14) from the graph. There are arcs from the nodes (8) and (13) to (6) and from all nodes (11), (12), (15), (16), (17) to (14). But (6) and (14) have no outgoing arcs and thus, they are not on any cycle.

A set $\mathcal{P}' \neq \varnothing$ of DPs is a *cycle* if for every $v \to w, s \to t \in \mathcal{P}'$, there is a non-empty path from $v \to w$ to $s \to t$ traversing only pairs of $\mathcal{P}'$. A cycle $\mathcal{P}'$ is a *strongly connected component* ("SCC") if $\mathcal{P}'$ is not a proper subset of another cycle.

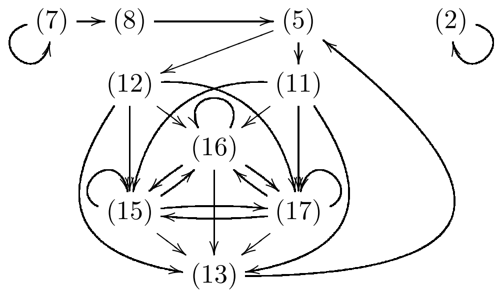

Fig. 1. Dependency graph for Ex. 1

One can prove termination separately for each SCC. Thus, the following processor (whose soundness is obvious and completely analogous to the non-context-sensitive case) modularizes termination proofs.

Theorem 16 (CS-Dependency Graph Processor). *For $d = (\mathcal{P}, \mathcal{R}, \mu, e)$, let $Proc(d) = \{(\mathcal{P}_1, \mathcal{R}, \mu, e), \ldots, (\mathcal{P}_n, \mathcal{R}, \mu, e)\}$, where $\mathcal{P}_1, \ldots, \mathcal{P}_n$ are the SCCs of the $(\mathcal{P}, \mathcal{R}, \mu, e)$-dependency graph. Then Proc is sound.*

Example 17. The graph in Fig. 1 has the three SCCs $\mathcal{P}_1 = \{(2)\}$, $\mathcal{P}_2 = \{(7)\}$, $\mathcal{P}_3 = \{(5), (11)\text{-}(13), (15)\text{-}(17)\}$. Thus, the initial DP problem $(\mathsf{DP}(\mathcal{R}, \mu), \mathcal{R}, \mu, \mathbf{i})$ is transformed into the new problems $(\mathcal{P}_1, \mathcal{R}, \mu, \mathbf{i})$, $(\mathcal{P}_2, \mathcal{R}, \mu, \mathbf{i})$, $(\mathcal{P}_3, \mathcal{R}, \mu, \mathbf{i})$.

As in the non-context-sensitive setting, the CS-dependency graph is not computable and thus, one has to use estimations to over-approximate the graph. For example, [1, 4] adapted the estimation of [5] that was originally developed for ordinary rewriting: $\textsc{Cap}^\mu(t)$ replaces all active subterms of t with defined root symbol by different fresh variables. Multiple occurrences of the same such subterm are also replaced by pairwise different variables. $\textsc{Ren}^\mu(t)$ replaces all active occurrences of variables in t by different fresh variables (i.e., no variable occurs at several active positions in $\textsc{Ren}^\mu(t)$). So $\textsc{Ren}^\mu(\textsc{Cap}^\mu(\mathsf{IF}(\mathsf{gt}(y, \mathsf{0}), \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y)), x))) = \textsc{Ren}^\mu(\mathsf{IF}(z, \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y)), x)) = \mathsf{IF}(z', \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y)), x)$.

To estimate the CS-dependency graph in the case $e = \mathbf{t}$, one draws an arc from $v \to w$ to $s \to t$ whenever $\textsc{Ren}^\mu(\textsc{Cap}^\mu(w))$ and s unify.[13] If $e = \mathbf{i}$, then one can modify $\textsc{Cap}^\mu$ and $\textsc{Ren}^\mu$ by taking into account that instantiated subterms at active positions of the left-hand side must be in normal form, cf. [4]. $\textsc{Cap}^\mu_v(w)$ is like $\textsc{Cap}^\mu(w)$, but the replacement of subterms of w by fresh variables is not done if the subterms also occur at active positions of v. Similarly, $\textsc{Ren}^\mu_v(w)$ is like $\textsc{Ren}^\mu(w)$, but the renaming of variables in w is not done if the variables also occur active in v. Now we draw an arc from $v \to w$ to $s \to t$ whenever $\textsc{Ren}^\mu_v(\textsc{Cap}^\mu_v(w))$ and s unify by an mgu θ where $v\theta$ and $s\theta$ are in normal form.[14]

It turns out that for the TRS of Ex. 1, the resulting estimated dependency graph is identical to the "real" graph in Fig. 1.

[13] Here (and also later in the instantiation processor of Sect. 4.3), we always assume that $v \to w$ and $s \to t$ are renamed apart to be variable-disjoint.

[14] These estimations can be improved further by adapting existing refinements to the context-sensitive case. However, different to the non-context-sensitive case, for $e = \mathbf{i}$ it is not sufficient to check only for unification of $\textsc{Cap}^\mu_v(w)$ and s (i.e., renaming variables with $\textsc{Ren}^\mu_v$ is also needed). This can be seen from the non-innermost terminating CS-TRS $(\mathcal{R}, \mu)$ from [4, Ex. 8] with $\mathcal{R} = \{\mathsf{f}(\mathsf{s}(x), x) \to \mathsf{f}(x, x), \mathsf{a} \to \mathsf{s}(\mathsf{a})\}$

4.2 Reduction Pair Processor

There are several processors to simplify DP problems by applying suitable *well-founded orders* (e.g., the *reduction pair processor* [16, 20], the *subterm criterion processor* [21], etc.). Due to the absence of collapsing DPs, most of these processors are now straightforward to adapt to the context-sensitive setting. In the following, we present the reduction pair processor with *usable rules*, because it is the only processor whose adaption is more challenging. (The adaption is similar to the one in [4, 19] for the CS-DPs of Def. 2.)

To prove that a DP problem is finite, the reduction pair processor generates constraints which should be satisfied by a *μ-reduction pair* $(\succsim, \succ)$ [1]. Here, $\succsim$ is a stable μ-monotonic quasi-order, $\succ$ is a stable well-founded order, and $\succsim$ and $\succ$ are compatible (i.e., $\succ \circ \succsim \subseteq \succ$ or $\succsim \circ \succ \subseteq \succ$). Here, *$\mu$-monotonicity* means that $s_i \succsim t_i$ implies $f(s_1, ..., s_i, ..., s_n) \succsim f(s_1, ..., t_i, ..., s_n)$ whenever $i \in \mu(f)$.

For a DP problem $(\mathcal{P}, \mathcal{R}, \mu, e)$, the generated constraints ensure that some rules in $\mathcal{P}$ are strictly decreasing (w.r.t. $\succ$) and all remaining rules in $\mathcal{P}$ and $\mathcal{R}$ are weakly decreasing (w.r.t. $\succsim$). Requiring $\ell \succsim r$ for all $\ell \to r \in \mathcal{R}$ ensures that in a chain $s_1 \to t_1, s_2 \to t_2, ...$ with $t_i\sigma \hookrightarrow^*_{\mathcal{R},\mu} s_{i+1}\sigma$, we have $t_i\sigma \succsim s_{i+1}\sigma$ for all i. Hence, if a reduction pair satisfies the constraints, then one can delete the strictly decreasing pairs from $\mathcal{P}$ as they cannot occur infinitely often in chains.

To improve this idea, it is desirable to require only a weak decrease of *certain* instead of *all* rules. In the non-context-sensitive setting, when proving innermost termination, it is sufficient if just the *usable rules* are weakly decreasing [5]. The same is true when proving full termination, provided that $\succsim$ is *$\mathcal{C}_\varepsilon$-compatible*, i.e., $\mathsf{c}(x, y) \succsim x$ and $\mathsf{c}(x, y) \succsim y$ holds for a fresh function symbol c [16, 21].

For a term containing a symbol f, all f-rules are *usable*. Moreover, if the f-rules are usable and f *depends on* h (denoted $f \blacktriangleright_{\mathcal{R}} h$) then the h-rules are usable as well. Here, $f \blacktriangleright_{\mathcal{R}} h$ if $f = h$ or if there is a symbol g with $g \blacktriangleright_{\mathcal{R}} h$ and g occurs in the right-hand side of an f-rule. The usable rules of a DP problem are defined to be the usable rules of the right-hand sides of the DPs.

As in [4, 19], Def. 18 adapts[15] the concept of usable rules to the CS setting, resulting in $\mathcal{U}^{\blacktriangleright}(\mathcal{P}, \mathcal{R}, \mu)$. But as shown in [19], for CS rewriting it is also helpful to consider an alternative definition of "dependence" $\rhd_{\mathcal{R},\mu}$ where f also depends on symbols from *left-hand sides* of f-rules. Let $\mathcal{F}^\mu(t)$ (resp. $\mathcal{F}^{\not\mu}(t)$) contain all function symbols occurring at active (resp. inactive) positions of a term t.

Definition 18 (CS-Usable Rules). *Let $Rls(f) = \{\ell \to r \in \mathcal{R} \mid \mathrm{root}(\ell) = f\}$. For any symbols f, h and CS-TRS $(\mathcal{R}, \mu)$, let $f \blacktriangleright_{\mathcal{R},\mu} h$ if $f = h$ or if there is a symbol g with $g \blacktriangleright_{\mathcal{R},\mu} h$ and a rule $\ell \to r \in Rls(f)$ with $g \in \mathcal{F}^\mu(r)$. Let $f \rhd_{\mathcal{R},\mu} h$ if $f = h$ or if there is a symbol g with $g \rhd_{\mathcal{R},\mu} h$ and a rule $\ell \to r \in Rls(f)$ with*

and $\mu(\mathsf{f}) = \{1\}$, $\mu(\mathsf{s}) = \varnothing$. Clearly, $\textsc{Cap}^\mu_{\mathsf{F}(\mathsf{s}(x),x)}(\mathsf{F}(x, x)) = \mathsf{F}(x, x)$ does not unify with $\mathsf{F}(\mathsf{s}(y), y)$. In contrast, $\textsc{Ren}^\mu_{\mathsf{F}(\mathsf{s}(x),x)}(\textsc{Cap}^\mu_{\mathsf{F}(\mathsf{s}(x),x)}(\mathsf{F}(x, x))) = \mathsf{F}(x', x)$ unifies with $\mathsf{F}(\mathsf{s}(y), y)$. Thus, without using $\textsc{Ren}^\mu_{\mathsf{F}(\mathsf{s}(x),x)}$ one would conclude that the dependency graph has no cycle and wrongly prove (innermost) termination.

[15] The adaptions can also be extended to refined definitions of usable rules [14, 16].

$g \in \mathcal{F}^{\not\mu}(\ell) \cup \mathcal{F}(r)$. *We define two forms of* usable rules*:*

$$\mathcal{U}^{\blacktriangleright}(\mathcal{P},\mathcal{R},\mu) = \bigcup_{s\to t\in\mathcal{P}, f\in\mathcal{F}^{\mu}(t), f\blacktriangleright_{\mathcal{R},\mu} g} Rls(g)$$
$$\mathcal{U}^{\rhd}(\mathcal{P},\mathcal{R},\mu) = \bigcup_{s\to t\in\mathcal{P}, f\in\mathcal{F}^{\not\mu}(s)\cup\mathcal{F}(t), f\rhd_{\mathcal{R},\mu} g} Rls(g) \cup \bigcup_{\ell\to r\in\mathcal{R}, f\in\mathcal{F}^{\not\mu}(r), f\rhd_{\mathcal{R},\mu} g} Rls(g)$$

Example 19. We continue Ex. 17. $\mathcal{U}^{\blacktriangleright}(\mathcal{P}_1,\mathcal{R},\mu) = \varnothing$ *for* $\mathcal{P}_1 = \{(2)\}$*, since there is no defined symbol at an active position in the right-hand side* $\mathsf{GT}(x,y)$ *of (2). For* $\mathcal{P}_2 = \{(7)\}$*,* $\mathcal{U}^{\blacktriangleright}(\mathcal{P}_2,\mathcal{R},\mu)$ *are the* minus-*,* if-*, and* gt*-rules, since* minus *occurs at an active position in* $\mathsf{D}(\mathsf{minus}(x,y),\mathsf{s}(y))$ *and* minus *depends on* if *and* gt*. For* $\mathcal{P}_3 = \{(5),(11)\text{-}(13),(15)\text{-}(17)\}$*,* $\mathcal{U}^{\blacktriangleright}(\mathcal{P}_3,\mathcal{R},\mu)$ *are the* gt- *and* p*-rules, as* gt *and* p *are the only defined symbols at active positions of right-hand sides in* $\mathcal{P}_3$*.*

In contrast, all $\mathcal{U}^{\rhd}(\mathcal{P}_i,\mathcal{R},\mu)$ *contain all rules except the* div*-rules, as* minus *and* p *are root symbols of hidden terms and* minus *depends on* if *and* gt*.*

As shown in [4, 19], the direct adaption of the usable rules to the context-sensitive case (i.e., $\mathcal{U}^{\blacktriangleright}(\mathcal{P},\mathcal{R},\mu)$) can only be used for *conservative* CS-TRSs (if $e = \mathbf{i}$) resp. for *strongly conservative* CS-TRSs (if $e = \mathbf{t}$).[16] Let $\mathcal{V}^{\mu}(t)$ (resp. $\mathcal{V}^{\not\mu}(t)$) be all variables occurring at active (resp. inactive) positions of a term t.

Definition 20 (Conservative and Strongly Conservative). *A CS-TRS* $(\mathcal{R},\mu)$ *is* conservative *iff* $\mathcal{V}^{\mu}(r) \subseteq \mathcal{V}^{\mu}(\ell)$ *for all rules* $\ell \to r \in \mathcal{R}$*. It is* strongly conservative *iff it is conservative and moreover,* $\mathcal{V}^{\mu}(\ell) \cap \mathcal{V}^{\not\mu}(\ell) = \varnothing$ *and* $\mathcal{V}^{\mu}(r) \cap \mathcal{V}^{\not\mu}(r) = \varnothing$ *for all rules* $\ell \to r \in \mathcal{R}$*.*

Now we can define the reduction pair processor.

Theorem 21 (CS-Reduction Pair Processor). *Let* $(\succsim, \succ)$ *be a* μ*-reduction pair. For a CS-DP Problem* $d = (\mathcal{P},\mathcal{R},\mu,e)$*, the result of* $Proc(d)$ *is*

- $\{(\mathcal{P}\setminus\succ,\mathcal{R},\mu,e)\}$*, if* $\mathcal{P} \subseteq (\succ\cup\succsim)$ *and at least one of the following holds:*

  (i) $\mathcal{U}^{\blacktriangleright}(\mathcal{P},\mathcal{R},\mu) \subseteq \succsim$*,* $\mathcal{P}\cup\mathcal{U}^{\blacktriangleright}(\mathcal{P},\mathcal{R},\mu)$ *is strongly conservative,* $\succsim$ *is* $\mathcal{C}_\varepsilon$*-compatible*
  (ii) $\mathcal{U}^{\blacktriangleright}(\mathcal{P},\mathcal{R},\mu) \subseteq \succsim$*,* $\mathcal{P}\cup\mathcal{U}^{\blacktriangleright}(\mathcal{P},\mathcal{R},\mu)$ *is conservative,* $e = \mathbf{i}$
  (iii) $\mathcal{U}^{\rhd}(\mathcal{P},\mathcal{R},\mu) \subseteq \succsim$*,* $\succsim$ *is* $\mathcal{C}_\varepsilon$*-compatible*
  (iv) $\mathcal{R} \subseteq \succsim$

- $\{d\}$*, otherwise.*

Then Proc is sound.

Example 22. As $\mathcal{U}^{\blacktriangleright}(\mathcal{P}_1,\mathcal{R},\mu) = \varnothing$ and $\mathcal{P}_1 = \{(2)\}$ *is even strongly conservative, by Thm. 21 (i) or (ii) we only have to orient (2), which already works with the embedding order. So* $(\mathcal{P}_1,\mathcal{R},\mu,\mathbf{i})$ *is transformed to the empty set of DP problems.*

[16] The corresponding counterexamples in [4, 19] show that these restrictions are still necessary for our new notion of CS-DPs. In cases where one cannot use $\mathcal{U}^{\blacktriangleright}$, one can also attempt a termination proof where one drops the replacement map, i.e., where one regards the ordinary TRS $\mathcal{R}$ instead of the CS-TRS $(\mathcal{R},\mu)$. This may be helpful, since $\mathcal{U}^{\rhd}$ is not necessarily a subset of the non-context-sensitive usable rules.

For $\mathcal{P}_2 = \{(7)\}, \mathcal{U}^{\blacktriangleright}(\mathcal{P}_2, \mathcal{R}, \mu)$ contains the if*-rules which are not conservative. Hence, we use Thm. 21 (iii) with a reduction pair based on the following max-polynomial interpretation [9]:* $[\mathsf{D}(x,y)] = [\mathsf{minus}(x,y)] = [\mathsf{p}(x)] = x$, $[\mathsf{s}(x)] = x+1$, $[\mathsf{if}(x,y,z)] = \max(y,z)$, $[\mathsf{0}] = [\mathsf{gt}(x,y)] = [\mathsf{true}] = [\mathsf{false}] = 0$. *Then the DP (7) is strictly decreasing and all rules from $\mathcal{U}^{\rhd}(\mathcal{P}_2, \mathcal{R}, \mu)$ are weakly decreasing. Thus, the processor also transforms $(\mathcal{P}_2, \mathcal{R}, \mu, \mathbf{i})$ to the empty set of DP problems.*

Finally, we regard $\mathcal{P}_3 = \{(5), (11)\text{-}(13), (15)\text{-}(17)\}$ where we use Thm. 21 (iii) with the interpretation $[\mathsf{M}(x,y)] = [\mathsf{minus}(x,y)] = x+y+1$, $[\mathsf{IF}(x,y,z)] = [\mathsf{if}(x,y,z)] = \max(y,z)$, $[\mathsf{U}(x)] = [\mathsf{p}(x)] = [\mathsf{s}(x)] = x$, $[\mathsf{0}] = [\mathsf{gt}(x,y)] = [\mathsf{true}] = [\mathsf{false}] = 0$. *Then the DPs (16) and (17) are strictly decreasing, whereas all other DPs from $\mathcal{P}_3$ and all rules from $\mathcal{U}^{\rhd}(\mathcal{P}_3, \mathcal{R}, \mu)$ are weakly decreasing. So the processor results in the DP problem $(\{(5), (11)\text{-}(13), (15)\}, \mathcal{R}, \mu, \mathbf{i})$.*

Next we apply $[\mathsf{M}(x,y)] = [\mathsf{minus}(x,y)] = x+1$, $[\mathsf{IF}(x,y,z)] = \max(y, z+1)$, $[\mathsf{if}(x,y,z)] = \max(y,z)$, $[\mathsf{U}(x)] = [\mathsf{p}(x)] = [\mathsf{s}(x)] = x$, $[\mathsf{0}] = [\mathsf{gt}(x,y)] = [\mathsf{true}] = [\mathsf{false}] = 0$. *Now (12) is strictly decreasing and all other remaining DPs and usable rules are weakly decreasing. Removing (12) yields $(\{(5), (11), (13), (15)\}, \mathcal{R}, \mu, \mathbf{i})$.*

Thm. 21 (iii) and (iv) are a significant improvement over previous reduction pair processors [1, 2, 4, 19] for the CS-DPs from Def. 2. The reason is that all previous CS-reduction pair processors require that the context-sensitive subterm relation is contained in $\succsim$ (i.e., $\trianglerighteq_\mu \subseteq \succsim$) whenever there are collapsing DPs. This is a very hard requirement which destroys one of the main advantages of the DP method (i.e., the possibility to filter away arbitrary arguments).[17] With our new non-collapsing CS-DPs, this requirement is no longer needed.

Example 23. If one requires $\trianglerighteq_\mu \subseteq \succsim$, then the reduction pair processor would fail for Ex. 1, since then one cannot make the DP (7) strictly decreasing. The reason is that due to $2 \in \mu(\mathsf{minus})$, $\trianglerighteq_\mu \subseteq \succsim$ implies $\mathsf{minus}(x,y) \succsim y$. So one cannot "filter away" the second argument of minus*. But then a strict decrease of DP (7) together with μ-monotonicity of $\succsim$ implies $\mathsf{D}(\mathsf{s}(x), \mathsf{s}(\mathsf{s}(x))) \succ \mathsf{D}(\mathsf{minus}(x, \mathsf{s}(x)), \mathsf{s}(\mathsf{s}(x))) \succsim \mathsf{D}(\mathsf{s}(x), \mathsf{s}(\mathsf{s}(x)))$, in contradiction to the well-foundedness of $\succ$.*

4.3 Transforming Context-Sensitive Dependency Pairs

To increase the power of the DP method, there exist several processors to transform a DP into new pairs (e.g., *narrowing*, *rewriting*, *instantiating*, or *forward instantiating* DPs [16]). We now adapt the *instantiation* processor to the context-sensitive setting. Similar adaptions can also be done for the other processors.[18]

[17] Moreover, previous CS-reduction pair processors also require $f(x_1, \ldots, x_n) \succsim f^\sharp(x_1, \ldots, x_n)$ for all $f \in \mathcal{D}$ or $f(x_1, \ldots, x_n) \succ f^\sharp(x_1, \ldots, x_n)$ for all $f \in \mathcal{D}$. This requirement also destroys an important feature of the DP method, i.e., that tuple symbols $f^\sharp$ can be treated independently from the original corresponding symbols f. This feature often simplifies the search for suitable reduction pairs considerably.

[18] In the papers on CS-DPs up to now, the only existing adaption of such a processor was the straightforward adaption of the *narrowing* processor in the case $e = \mathbf{t}$, cf. [2]. However, this processor would not help for the TRS of Ex. 1.

The idea of this processor is the following. For a DP $s \to t$, we investigate which DPs $v \to w$ can occur before $s \to t$ in chains. To this end, we use the same estimation as for dependency graphs in Sect. 4.1, i.e., we check whether there is an mgu θ of $\textsc{Ren}^\mu(\textsc{Cap}^\mu(w))$ and s if $e = \mathbf{t}$ and analogously for $e = \mathbf{i}$.[19] Then we replace $s \to t$ by the new DPs $s\theta \to t\theta$ for all such mgu's θ. This is sound since in any chain $\ldots, v \to w, s \to t, \ldots$ where an instantiation of w reduces to an instantiation of s, one could use the new DP $s\theta \to t\theta$ instead.

Theorem 24 (CS-Instantiation Processor). *Let $\mathcal{P}' = \mathcal{P} \uplus \{s \to t\}$. For $d = (\mathcal{P}', \mathcal{R}, \mu, e)$, let the result of $Proc(d)$ be $(\mathcal{P} \cup \overline{\mathcal{P}}, \mathcal{R}, \mu, e)$ where*

- $\overline{\mathcal{P}} = \{s\theta \to t\theta \mid \theta = mgu(\textsc{Ren}^\mu(\textsc{Cap}^\mu(w)), s), v \to w \in \mathcal{P}'\}$, *if* $e = \boldsymbol{t}$
- $\overline{\mathcal{P}} = \{s\theta \to t\theta \mid \theta = mgu(\textsc{Ren}^\mu_v(\textsc{Cap}^\mu_v(w)), s), v \to w \in \mathcal{P}', s\theta, v\theta \text{ normal}\}$, *if* $e = \boldsymbol{i}$

Then Proc is sound.

Example 25. For the TRS of Ex. 1, we still had to solve the problem $(\{(5), (11), (13), (15)\}, \mathcal{R}, \mu, \mathbf{i})$*, cf. Ex. 22. DP (11) has the variable-renamed left-hand side* $\mathsf{IF}(\mathsf{true}, x', y')$*. So the only DP that can occur before (11) in chains is* (5) *with the right-hand side* $\mathsf{IF}(\mathsf{gt}(y, \mathsf{0}), \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y)), x)$*. Recall* $\textsc{Ren}^\mu(\textsc{Cap}^\mu(\mathsf{IF}(\mathsf{gt}(y, \mathsf{0}), \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y)), x))) = \mathsf{IF}(z', \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y)), x)$*, cf. Sect. 4.1. So the mgu is* $\theta = [z'/\mathsf{true},\ x'/\mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y)),\ y'/x]$*. Hence, we can replace (11) by*

$$\mathsf{IF}(\mathsf{true}, \mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y)), x) \to \mathsf{U}(\mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y))) \tag{20}$$

Here the CS variant of the instantiation processor is advantageous over the non-CS one which uses $\textsc{Cap}$ *instead of* $\textsc{Cap}^\mu$*, where* $\textsc{Cap}$ *replaces all subterms with defined root (e.g.,* $\mathsf{minus}(\mathsf{p}(x), \mathsf{p}(y))$*) by fresh variables. So the non-CS processor would not help here as it only generates a variable-renamed copy of (11).*

When re-computing the dependency graph, there is no arc from (20) to (15) as $\mu(\mathsf{U}) = \varnothing$*. So the DP problem is decomposed into* $(\{(15)\}, \mathcal{R}, \mu, \mathbf{i})$ *(which is easily solved by the reduction pair processor) and* $(\{(5), (20), (13)\}, \mathcal{R}, \mu, \mathbf{i})$*.*

Now we apply the reduction pair processor again with the following rational *polynomial interpretation [10]:* $[\mathsf{M}(x, y)] = \frac{3}{2}x + \frac{1}{2}y$*,* $[\mathsf{minus}(x, y)] = 2x + \frac{1}{2}y$*,* $[\mathsf{IF}(x, y, z)] = \frac{1}{2}x + y + \frac{1}{2}z$*,* $[\mathsf{if}(x, y, z)] = \frac{1}{2}x + y + z$*,* $[\mathsf{U}(x)] = x$*,* $[\mathsf{p}(x)] = [\mathsf{gt}(x, y)] = \frac{1}{2}x$*,* $[\mathsf{s}(x)] = 2x + 2$*,* $[\mathsf{true}] = 1$*,* $[\mathsf{false}] = [\mathsf{0}] = 0$*. Then (20) is strictly decreasing and can be removed, whereas all other remaining DPs and usable rules are weakly decreasing. A last application of the dependency graph processor then detects that there is no cycle anymore and thus, it returns the empty set of DP problems. Hence, termination of the TRS from Ex. 1 is proved. As shown in our experiments in Sect. 5, this proof can easily be performed automatically.*

5 Experiments and Conclusion

We have developed a new notion of context-sensitive dependency pairs which improves significantly over previous notions. There are two main advantages:

[19] The counterexample of [4, Ex. 8] in Footnote 14 again illustrates why $\textsc{Ren}^\mu_v$ is also needed in the innermost case (whereas this is unnecessary for non-CS rewriting).

(1) **Easier adaption of termination techniques to CS rewriting**
Now CS-DPs are very similar to DPs for ordinary rewriting and consequently, the existing powerful termination techniques from the DP framework can easily be adapted to context-sensitive rewriting. We have demonstrated this with some of the most popular DP processors in Sect. 4. Our adaptions subsume the existing earlier adaptions of the dependency graph [2], of the usable rules [19], and of the modifications for innermost rewriting [4], which were previously developed for the notion of CS-DPs from [1].

(2) **More powerful termination analysis for CS rewriting**
Due to the absence of collapsing CS-DPs, one does not have to impose extra restrictions anymore when extending the DP processors to CS rewriting, cf. Ex. 23. Hence, the power of termination proving is increased substantially.

To substantiate Claim (2), we performed extensive experiments. We implemented our new non-collapsing CS-DPs and all DP processors from this paper in the termination prover AProVE [15].[20] In contrast, the prover MU-TERM [3] uses the collapsing CS-DPs. Moreover, the processors for these CS-DPs are not formulated within the DP framework and thus, they cannot be applied in the same flexible and modular way. While MU-TERM was the most powerful tool for termination analysis of context-sensitive rewriting up to now (as demonstrated by the *International Competition of Termination Tools* 2007 [26]), due to our new notion of CS-DPs, now AProVE is substantially more powerful. For instance, AProVE easily proves termination of our leading example from Ex. 1, whereas MU-TERM fails. Moreover, we tested the tools on all 90 context-sensitive TRSs from the *Termination Problem Data Base* that was used in the competition. We used a time limit of 120 seconds for each example. Then MU-TERM can prove termination of 68 examples, whereas the new version of AProVE proves termination of 78 examples (including all 68 TRSs where MU-TERM is successful).[21] Since 4 examples are known to be non-terminating, at most 8 more of the 90 examples could potentially be detected as terminating. So due to the results of this paper, termination proving of context-sensitive rewriting has now become very powerful. To experiment with our implementation and for details, we refer to `http://aprove.informatik.rwth-aachen.de/eval/CS-DPs/`.

References

1. B. Alarcón, R. Gutiérrez, and S. Lucas. Context-sensitive dependency pairs. In *Proc. FSTTCS'06*, LNCS 4337, pages 297-308, 2006.
2. B. Alarcón, R. Gutiérrez, and S. Lucas. Improving the context-sensitive dependency graph. In *Proc. PROLE'06*, ENTCS 188, pages 91-103, 2007.
3. B. Alarcón, R. Gutiérrez, J. Iborra, S. Lucas. Proving termination of context-sensitive rewriting with MU-TERM. *Pr. PROLE'06*, ENTCS 188, p. 105-115, 2007.

[20] We also used the subterm criterion and forward instantiation processors, cf. Sect. 4.
[21] If AProVE is restricted to use exactly the same processors as MU-TERM, then it still succeeds on 74 examples. So its superiority is indeed mainly due to the new CS-DPs which enable an easy adaption of the DP framework to the CS setting.

4. B. Alarcón and S. Lucas. Termination of innermost context-sensitive rewriting using dependency pairs. In *Proc. FroCoS'07*, LNAI 4720, pages 73-87, 2007.
5. T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133-178, 2000.
6. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge, 1998.
7. C. Borralleras, S. Lucas, and A. Rubio. Recursive path orderings can be context-sensitive. In *Proc. CADE'02*, LNAI 2392, pages 314-331, 2002.
8. N. Dershowitz. Termination by abstraction. *ICLP'04*, LNCS 3132, p. 1-18, 2004.
9. C. Fuhs, J. Giesl, A. Middeldorp, P. Schneider-Kamp, R. Thiemann, and H. Zankl. Maximal termination. In *Proc. RTA'08*, LNCS 5117, pages 110-125, 2008.
10. C. Fuhs, R. Navarro-Marset, C. Otto, J. Giesl, S. Lucas, and P. Schneider-Kamp. Search techniques for rational polynomial orders. In *Proc. AISC'08*, LNAI 5144, pages 109-124, 2008.
11. J. Giesl and A. Middeldorp. Innermost termination of context-sensitive rewriting. In *Proc. DLT'02*, LNCS 2450, pages 231-244, 2003.
12. J. Giesl and A. Middeldorp. Transformation techniques for context-sensitive rewrite systems. *Journal of Functional Programming*, 14(4):379-427, 2004.
13. J. Giesl, R. Thiemann, P. Schneider-Kamp. The DP framework: combining techniques for automated termination proofs. In *LPAR'04*, LNAI 3452, 301-331, 2005.
14. J. Giesl, R. Thiemann, P. Schneider-Kamp. Proving and disproving termination of higher-order functions. In *Proc. FroCoS'05*, LNAI 3717, pages 216-231, 2005.
15. J. Giesl, P. Schneider-Kamp, R. Thiemann. AProVE 1.2: Automatic termination proofs in the DP framework. In *Proc. IJCAR'06*, LNAI 4130, pages 281-286, 2006.
16. J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Mechanizing and improving dependency pairs. *Journal of Automatic Reasoning*, 37(3):155-203, 2006.
17. B. Gramlich. Generalized sufficient conditions for modular termination of rewriting. *Appl. Algebra in Engineering, Comm. and Computing*, 5:131-151, 1994.
18. B. Gramlich and S. Lucas. Simple termination of context-sensitive rewriting. In *Proc. RULE'02*, ACM Press, pages 29-41, 2002.
19. R. Gutiérrez, S. Lucas, and X. Urbain. Usable rules for context-sensitive rewrite systems. In *Proc. RTA'08*, LNCS 5117, pages 126-141, 2008.
20. N. Hirokawa and A. Middeldorp. Automating the dependency pair method. *Information and Computation*, 199(1,2):172-199, 2005.
21. N. Hirokawa and A. Middeldorp. Tyrolean Termination Tool: techniques and features. *Information and Computation*, 205(4):474-511, 2007.
22. S. Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1):1-61, 1998.
23. S. Lucas. Context-sensitive rewriting strategies. *Inf. Comp.*, 178(1):293-343, 2002.
24. S. Lucas. Polynomials for proving termination of context-sensitive rewriting. In *Proc. FOSSACS'04*, LNCS 2987, pages 318-332, 2004.
25. S. Lucas. Proving termination of context-sensitive rewriting by transformation. *Information and Computation*, 204(12):1782-1846, 2006.
26. C. Marché and H. Zantema. The termination competition. In *Proc. RTA'07*, LNCS 4533, pages 303-313, 2007.
27. X. Urbain. Modular & incremental automated termination proofs. *Journal of Automated Reasoning*, 32(4):315-355, 2004.

A Proof of Thm. 12 for Innermost Termination

The proof is similar to the proof for full termination.

For soundness, $\mathcal{M}_{\infty,\mu}$ now denotes the set of all minimal non-*innermost*-terminating terms: $t \in \mathcal{M}_{\infty,\mu}$ iff t is not innermost terminating and every active proper subterm of t is innermost terminating. If $\mathcal{R}$ is not innermost terminating, then there must be terms t, t_i, s_i, t'_{i+1} such that

$$t \overset{\mathsf{i}\ >\varepsilon}{\hookrightarrow}{}^{*}_{\mathcal{R},\mu} \ t_1 \overset{\mathsf{i}\ \varepsilon}{\hookrightarrow}_{\mathcal{R},\mu} \ s_1 \trianglerighteq_\mu t'_2 \overset{\mathsf{i}\ >\varepsilon}{\hookrightarrow}{}^{*}_{\mathcal{R},\mu} \ t_2 \overset{\mathsf{i}\ \varepsilon}{\hookrightarrow}_{\mathcal{R},\mu} \ s_2 \trianglerighteq_\mu t'_3 \overset{\mathsf{i}\ >\varepsilon}{\hookrightarrow}{}^{*}_{\mathcal{R},\mu} \ t_3 \ldots$$

where $t_i, t'_i \in \mathcal{M}_{\infty,\mu}$ for all $i \geq 1$ and where all proper subterms of t are innermost terminating.

The proof that all t_i have the hiding property now works as before. To show that this implies Thm. 12, one additionally has to prove that all terms in the reduction $t_i^\sharp \to^+_{\mathsf{DP}(\mathcal{R},\mu)} {t'_{i+1}}^\sharp$ except ${t'_{i+1}}^\sharp$ are in normal form w.r.t. $(\mathcal{R}, \mu)$. This holds as each proper subterm at an active position of a t_i is in normal form w.r.t. $(\mathcal{R}, \mu)$. Since the replacement map for the tuple symbols is the same as for the corresponding defined symbols, and since tuple symbols do not occur in $\mathcal{R}$, we conclude that each $t_i^\sharp$ is in normal form. All other terms in the reduction except ${t'_{i+1}}^\sharp$ have the form $\mathsf{U}(\ldots)$ where $\mu(\mathsf{U}) = \varnothing$. Thus, they are in normal form as well.

For completeness, the proof is again analogous to the one for full termination. One only has to show that the innermost evaluation strategy is respected in the context-sensitive reduction sequence that is obtained from the infinite innermost chain $v_1 \to w_1, v_2 \to w_2, \ldots$ But this follows from the fact that for an innermost chain each $\sigma(v_i)$ is in normal form and thus, every proper subterm at an active position of $\sigma(v_i^\flat)$ is in normal form as well. $\square$

B Proof of Thm. 16

Every infinite chain must end in an SCC of the dependency graph. Thus, if there is an infinite $(\mathcal{P}, \mathcal{R}, \mu, e)$-chain, then there is also an infinite $(\mathcal{P}_i, \mathcal{R}, \mu, e)$-chain for some SCC $\mathcal{P}_i$. $\square$

C Proof of Thm. 21

We first need some definitions and results. To prove that we only need the usable rules instead of all the rules for proving finiteness of a CS-DP problem, we use a transformation [16, 17, 19, 21, 27] (in the innermost case this is not necessary). Let $\mathcal{C}_\varepsilon = \{\mathsf{c}(x, y) \to x, \mathsf{c}(x, y) \to y\}$ and we extend any replacement map μ to the fresh symbol c by defining $\mu(\mathsf{c}) = \{1, 2\}$. Then the idea is to transform an infinite $(\mathcal{P}, \mathcal{R}, \mu)$-chain into an infinite (not necessarily minimal) $(\mathcal{P}, \mathcal{U}^{\blacktriangleright}(\mathcal{P}, \mathcal{R}, \mu) \cup \mathcal{C}_\varepsilon, \mu)$-chain if $\mathcal{P} \cup \mathcal{U}^{\blacktriangleright}(\mathcal{P}, \mathcal{R}, \mu)$ is strongly conservative and into an infinite (not necessarily minimal) $(\mathcal{P}, \mathcal{U}^{\rhd}(\mathcal{P}, \mathcal{R}, \mu) \cup \mathcal{C}_\varepsilon, \mu)$-chain otherwise. Here, in a possibly non-minimal chain $s_1 \to t_1, s_2 \to t_2, \ldots$ with $t_i\sigma \hookrightarrow^* s_{i+1}\sigma$, one does not impose the requirement that all $t_i\sigma$ are terminating. In the following, let $\mathcal{F}$ be the set of all function symbols and let $\mathcal{T}(\mathcal{F}, \mathcal{V})$ denote the set of all terms over the symbols $\mathcal{F}$ and the variables $\mathcal{V}$. The transformation $\mathcal{I}_1$ is needed for the usable rules $\mathcal{U}^{\rhd}$ and $\mathcal{I}_2$ is needed for the usable rules $\mathcal{U}^{\blacktriangleright}$.

Definition 26 **($\mathcal{I}_1$).** *Let $\Delta \subseteq \mathcal{F}$ and let $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$. We define $\mathcal{I}_1(t)$:*

- $\mathcal{I}_1(x) = x$ *if* $x \in \mathcal{V}$
- $\mathcal{I}_1(f(t_1,\ldots,t_n)) = f(\mathcal{I}_1(t_1),\ldots,\mathcal{I}_1(t_n))$ *if* $f \in \Delta$ *or if* $f(t_1,\ldots,t_n)$ *is non-terminating w.r.t.* $(\mathcal{R},\mu)$
- $\mathcal{I}_1(f(t_1,\ldots,t_n)) = \mathcal{C}omp(\{f(\mathcal{I}_1(t_1),\ldots,\mathcal{I}_1(t_n))\} \cup \mathcal{R}ed_1(f(t_1,\ldots,t_n)))$ *if* $f \notin \Delta$ *and* $f(t_1,\ldots,t_n)$ *is terminating w.r.t.* $(\mathcal{R},\mu)$

where $\mathcal{R}ed_1(t) = \{\mathcal{I}_1(t') \mid t \hookrightarrow_{\mathcal{R},\mu} t'\}$. *Moreover,* $\mathcal{C}omp(\{t\} \uplus M) = \mathsf{c}(t, \mathcal{C}omp(M))$ *and* $\mathcal{C}omp(\varnothing) = \mathsf{d}$, *where* d *is a fresh constant. To make* $\mathcal{C}omp$ *well defined, in "*$\{t\} \uplus M$*", we assume that* t *is smaller than all terms in* M *w.r.t some total well-founded order on terms.*

Definition 27 **($\mathcal{I}_2$).** *Let $\Delta \subseteq \mathcal{F}$ and let $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$ be a terminating term w.r.t. $(\mathcal{R},\mu)$. We define $\mathcal{I}_2(t)$:*

- $\mathcal{I}_2(x) = x$ *if* $x \in \mathcal{V}$
- $\mathcal{I}_2(f(t_1,\ldots,t_n)) = f(\overline{t_1},\ldots,\overline{t_n})$ *if* $f \in \Delta$
- $\mathcal{I}_2(f(t_1,\ldots,t_n)) = \mathcal{C}omp(\{f(\overline{t_1},\ldots,\overline{t_n})\} \cup \mathcal{R}ed_2(f(t_1,\ldots,t_n)))$ *if* $f \notin \Delta$

where $\mathcal{R}ed_2(t) = \{\mathcal{I}_2(t') \mid t \hookrightarrow_{\mathcal{R},\mu} t'\}$. *Moreover,* $\overline{t_i} = \mathcal{I}_2(t_i)$ *if* $i \in \mu(f)$ *and* $\overline{t_i} = t_i$ *otherwise.*

Lemma 28 (Properties of $\mathcal{I}_1$). *Let $\Delta \subseteq \mathcal{F}$ such that $f \in \Delta$ whenever there is a rule $\ell \to r \in \mathcal{R}$ with*

- $f \in \mathcal{F}^{\not\mu}(r)$ *or*
- $\mathrm{root}(\ell) \in \Delta$ *and* $f \in \mathcal{F}(r)$ *or*
- $\mathrm{root}(\ell) \in \Delta$ *and* $f \in \mathcal{F}^{\not\mu}(\ell)$

Let $t, s \in \mathcal{T}(\mathcal{F},\mathcal{V})$ and for any substitution σ, let $\mathcal{I}_1(\sigma)$ be the substitution with $x\,\mathcal{I}_1(\sigma) = \mathcal{I}_1(x\sigma)$.

1. *$\mathcal{I}_1$ is well defined.*
2. *If $t \in \mathcal{T}(\Delta,\mathcal{V})$, then $\mathcal{I}_1(t\sigma) = t\mathcal{I}_1(\sigma)$.*
3. *If $t \in \mathcal{T}(\mathcal{F},\mathcal{V})$ and all subterms t' of t at inactive positions are from $\mathcal{T}(\Delta,\mathcal{V})$, then $\mathcal{I}_1(t\sigma) \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} t\mathcal{I}_1(\sigma)$.*
4. *If t is terminating w.r.t. $(\mathcal{R},\mu)$ and $t \hookrightarrow_{\{\ell\to r\},\mu} s$, where $\ell \to r \in \mathcal{R}$, then $\mathcal{I}_1(t) \hookrightarrow^+_{\{\ell\to r\}\cup\mathcal{C}_\varepsilon,\mu} \mathcal{I}_1(s)$ if $\mathrm{root}(\ell) \in \Delta$ and $\mathcal{I}_1(t) \hookrightarrow^+_{\mathcal{C}_\varepsilon,\mu} \mathcal{I}_1(s)$ otherwise.*

Proof. **(1)** According to Def. 26, to obtain an infinite term as result of $\mathcal{I}_1(t)$, we would have to perform an infinite number of applications of $\mathcal{R}ed_1$. This means that t is terminating and that there exists an infinite sequence of the form $t = u_1 \trianglerighteq t_1 \hookrightarrow_{\mathcal{R},\mu} u_2 \trianglerighteq t_2 \hookrightarrow_{\mathcal{R},\mu} u_3 \ldots$ where $\mathrm{root}(t_i) \notin \Delta$ and t_i is terminating for all $i \geq 1$. W.l.o.g. we can assume that $t_i \not\trianglerighteq t_{i+1}$ (otherwise, we simply consider the modified sequence $\ldots \trianglerighteq t_{i-1} \hookrightarrow_{\mathcal{R},\mu} u_i \trianglerighteq t_{i+1} \hookrightarrow_{\mathcal{R},\mu} u_{i+1} \trianglerighteq \ldots$). For $i \geq 1$, there is a rule $\ell \to r$, an active position p of t_i, and a substitution σ such that $t_i = C[\sigma(\ell)]_p \hookrightarrow_{\mathcal{R},\mu} C[\sigma(r)]_p = u_{i+1} \trianglerighteq t_{i+1}$. We have the following possibilities:

- t_{i+1} is a subterm of u_{i+1} above position p, i.e., $t_{i+1} \trianglerighteq \sigma(r)$. Then $u_{i+1} \trianglerighteq_\mu t_{i+1}$ because p is an active position.
- t_{i+1} is a subterm of u_{i+1} neither above nor below p. Then we already have $t_i \rhd t_{i+1}$ in contradiction to the prerequisite.
- t_{i+1} is a subterm of u_{i+1} strictly below position p, i.e., $\sigma(r) \rhd t_{i+1}$. Note that there is no variable x in r such that $\sigma(x) \trianglerighteq t_{i+1}$, because that would already imply $t_i \rhd t_{i+1}$ in contradiction to the prerequisite. Hence, there is a subterm s with $s \notin \mathcal{V}$ and $r \rhd s$ such that $\sigma(s) = t_{i+1}$. Since $\mathrm{root}(s) \notin \Delta$ and $\mathcal{F}^{\not\mu}(r) \subseteq \Delta$, we have $r \trianglerighteq_\mu s$ and hence, $u_{i+1} \trianglerighteq_\mu t_{i+1}$.

The resulting sequence is: $t = u_1 \trianglerighteq_\mu t_1 \hookrightarrow_{\mathcal{R},\mu} u_2 \trianglerighteq_\mu t_2 \hookrightarrow_{\mathcal{R},\mu} u_3 \ldots$, contradicting the termination of t.

(2) We use induction on t. If t is a variable then $\mathcal{I}_1(t\sigma) = t\,\mathcal{I}_1(\sigma)$. If $t = f(t_1, \ldots, t_n)$, since $f \in \Delta$ we have $\mathcal{I}_1(t\sigma) = \mathcal{I}_1(f(t_1\sigma, \ldots, t_n\sigma)) = f(\mathcal{I}_1(t_1\sigma), \ldots, \mathcal{I}_1(t_n\sigma))$. By the induction hypothesis we get $f(\mathcal{I}_1(t_1\sigma), \ldots, \mathcal{I}_1(t_n\sigma)) = f(t_1\,\mathcal{I}_1(\sigma), \ldots, t_n\,\mathcal{I}_1(\sigma)) = t\,\mathcal{I}_1(\sigma)$.

(3) We use induction on t. If t is a variable then $\mathcal{I}_1(t\sigma) = t\,\mathcal{I}_1(\sigma)$. Now let $t = f(t_1, \ldots, t_n)$. By the prerequisite, if $i \notin \mu(f)$ then t_i only contains Δ-symbols and hence, $\mathcal{I}_1(t_i\sigma) = t_i\mathcal{I}_1(\sigma)$ by (2). If $i \in \mu(f)$ then by the induction hypothesis, $\mathcal{I}_1(t_i\sigma) \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} t_i\mathcal{I}_1(\sigma)$.

- If $f \in \Delta$ or t is non-terminating, then $\mathcal{I}_1(t\sigma) = \mathcal{I}_1(f(t_1\sigma, \ldots, t_n\sigma)) = f(\mathcal{I}_1(t_1\sigma), \ldots, \mathcal{I}_1(t_n\sigma)) \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} f(t_1\,\mathcal{I}_1(\sigma), \ldots, t_n\,\mathcal{I}_1(\sigma)) = t\,\mathcal{I}_1(\sigma)$.
- If $f \notin \Delta$ and t is terminating, then $\mathcal{I}_1(t\sigma) \hookrightarrow^+_{\mathcal{C}_\varepsilon,\mu} f(\mathcal{I}_1(t_1\sigma), \ldots, \mathcal{I}_1(t_n\sigma))$. Hence, $\mathcal{I}_1(t\sigma) \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} t\,\mathcal{I}_1(\sigma)$ as before.

(4) We use induction on the position p of the redex in the reduction $t \hookrightarrow_{\{\ell\to r\},\mu} s$. First let $\mathrm{root}(t) \in \Delta$ and $p = \varepsilon$. So we have $t = \ell\sigma \xrightarrow{\varepsilon}_{\mathcal{R}} r\sigma = s$ for some substitution σ. Moreover, $r \in \mathcal{T}(\Delta, \mathcal{V})$ and all subterms ℓ' of ℓ at inactive positions are from $\mathcal{T}(\Delta, \mathcal{V})$ by the conditions on Δ. By (3) and (2) we get $\mathcal{I}_1(\ell\sigma) \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} \ell\,\mathcal{I}_1(\sigma) \hookrightarrow_{\{\ell\to r\},\mu} r\,\mathcal{I}_1(\sigma) = \mathcal{I}_1(r\sigma)$.
Now let $\mathrm{root}(t) \in \Delta$ and $p \neq \varepsilon$. Hence, $t = f(t_1, \ldots, t_i, \ldots, t_n)$, $s = f(t_1, \ldots, s_i, \ldots, t_n)$, $i \in \mu(f)$, and $t_i \hookrightarrow_{\{\ell\to r\},\mu} s_i$. The induction hypothesis implies $\mathcal{I}_1(t_i) \hookrightarrow^+_{\{\ell\to r\}\cup\mathcal{C}_\varepsilon,\mu} \mathcal{I}_1(s_i)$ and hence, $\mathcal{I}_1(t) \hookrightarrow^+_{\{\ell\to r\}\cup\mathcal{C}_\varepsilon,\mu} \mathcal{I}_1(s)$.
Finally, we consider the case $\mathrm{root}(t) \notin \Delta$. Due to termination of t, we have $\mathcal{I}_1(t) = \mathcal{C}omp(\{\ldots\} \cup \mathcal{R}ed_1(t))$. Since $\mathcal{I}_1(s) \in \mathcal{R}ed_1(t)$, we obtain $\mathcal{C}omp(\{\ldots\} \cup \mathcal{R}ed_1(t)) \hookrightarrow^+_{\mathcal{C}_\varepsilon,\mu} \mathcal{I}_1(s)$. □

Lemma 29 (Properties of $\mathcal{I}_2$). *Let $\Delta \subseteq \mathcal{F}$ such that $f \in \Delta$ whenever there is a rule $\ell \to r \in \mathcal{R}$ with $\mathrm{root}(\ell) \in \Delta$ and $f \in \mathcal{F}^\mu(r)$. Let $t, s, t\sigma \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ be terminating w.r.t. $(\mathcal{R}, \mu)$, and let σ be a terminating substitution w.r.t. $(\mathcal{R}, \mu)$ (i.e., $\sigma(x)$ is terminating for all $x \in \mathcal{V}$). Let $[t, \sigma]$ be the term that results from t by replacing occurrences of $x \in \mathcal{V}(t)$ at position p in t by $\mathcal{I}_2(\sigma(x))$ if $p \in \mathcal{P}os^\mu(t)$ and by $\sigma(x)$ if $p \notin \mathcal{P}os^\mu(t)$.*

1. *$\mathcal{I}_2$ is well defined.*
2. *If $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and for all non-variable subterms t' at active positions of t we have $\mathrm{root}(t') \in \Delta$, then $\mathcal{I}_2(t\sigma) = [t, \sigma]$.*
3. *$\mathcal{I}_2(t\sigma) \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} [t, \sigma]$.*

4. *If $t \hookrightarrow_{\{\ell\to r\},\mu} s$ then $\mathcal{I}_2(t) \hookrightarrow^+_{\{\ell\to r\}\cup\mathcal{C}_\varepsilon,\mu} \mathcal{I}_2(s)$ under the condition that whenever* $\mathrm{root}(\ell) \in \Delta$ *then $\ell \to r \in \mathcal{R}$ is strongly conservative (i.e., $\mathcal{V}^\mu(r) \subseteq \mathcal{V}^\mu(\ell)$, $\mathcal{V}^\mu(\ell) \cap \mathcal{V}^{\not\mu}(\ell) = \varnothing$, and $\mathcal{V}^\mu(r) \cap \mathcal{V}^{\not\mu}(r) = \varnothing$).*

Proof. **(1)** According to Def. 27, to obtain an infinite term as result of $\mathcal{I}_2(t)$, we would have to perform an infinite number of applications of $\mathcal{R}ed_2$. Since t is terminating, all subterms at active positions are terminating as well. Hence, this is not possible.

(2) We use induction on t. If t is a variable then $\mathcal{I}_2(t\sigma) = [t,\sigma]$. Now let $t = f(t_1,\dots,t_n)$. Since $f \in \Delta$ we have $\mathcal{I}_2(t\sigma) = \mathcal{I}_2(f(t_1\sigma,\dots,t_n\sigma)) = f(\overline{t_1\sigma},\dots,\overline{t_n\sigma})$. For $i \in \mu(f)$, we have $\overline{t_i\sigma} = \mathcal{I}_2(t_i\sigma) = [t_i,\sigma]$ by the induction hypothesis. For $i \notin \mu(f)$, we have $\overline{t_i\sigma} = t_i\sigma$. This implies $f(\overline{t_1\sigma},\dots,\overline{t_n\sigma}) = [t,\sigma]$.

(3) We again use induction on t. If t is a variable then again $\mathcal{I}_2(t\sigma) = [t,\sigma]$. Now let $t = f(t_1,\dots,t_n)$. If $f \in \Delta$, then $\mathcal{I}_2(t\sigma) = \mathcal{I}_2(f(t_1\sigma,\dots,t_n\sigma)) = f(\overline{t_1\sigma},\dots,\overline{t_n\sigma})$ and if $f \notin \Delta$, then $\mathcal{I}_2(t\sigma) \hookrightarrow^+_{\mathcal{C}_\varepsilon,\mu} f(\overline{t_1\sigma},\dots,\overline{t_n\sigma})$. For $i \in \mu(f)$, we have $\overline{t_i\sigma} = \mathcal{I}_2(t_i\sigma) \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} [t_i,\sigma]$ by the induction hypothesis. For $i \notin \mu(f)$, we have $\overline{t_i\sigma} = t_i\sigma$. Together, this implies $f(\overline{t_1\sigma},\dots,\overline{t_n\sigma}) \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} [t,\sigma]$.

(4) We use induction on the position p of the redex in the reduction $t \hookrightarrow_{\{\ell\to r\},\mu} s$. First let $\mathrm{root}(t) \in \Delta$ and $p = \varepsilon$. So we have $t = \ell\sigma \xrightarrow{\varepsilon}_{\mathcal{R}} r\sigma = s$ for some substitution σ. Moreover, for all subterms r' at active positions of r, $\mathrm{root}(r') \in \Delta$ by the condition on Δ. For any term t, we define the substitution σ_t as $\sigma_t(x) = \mathcal{I}_2(\sigma(x))$ if $x \in \mathcal{V}^\mu(t)$ and $\sigma_t(x) = \sigma(x)$ otherwise. As $\mathcal{V}^\mu(\ell) \cap \mathcal{V}^{\not\mu}(\ell) = \varnothing$ and $\mathcal{V}^\mu(r) \cap \mathcal{V}^{\not\mu}(r) = \varnothing$, we have $[\ell,\sigma] = \ell\sigma_\ell$ and $[r,\sigma] = r\sigma_r$. Moreover, for all variables x we have $x\sigma_\ell \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} x\sigma_r$. To see this, note that by strong conservativity, σ_ℓ and σ_r only differ on variables $x \in \mathcal{V}^\mu(\ell) \setminus \mathcal{V}^\mu(r)$. Here, we have $x\sigma_\ell = \mathcal{I}_2(x\sigma) \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} [x,\sigma] = x\sigma = x\sigma_r$ by (3). Hence,

$$\begin{array}{rcll}
\mathcal{I}_2(t) & = & \mathcal{I}_2(\ell\sigma) & \\
& \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} & [\ell,\sigma] & \text{by (3)} \\
& = & \ell\sigma_\ell & \\
& \hookrightarrow^*_{\mathcal{C}_\varepsilon,\mu} & \ell\sigma_r & \\
& \to_{\{\ell\to r\}} & r\sigma_r & \\
& = & [r,\sigma] & \\
& = & \mathcal{I}_2(r\sigma) & \text{by (2)}
\end{array}$$

Now let $\mathrm{root}(t) \in \Delta$ and $p \neq \varepsilon$. Hence, $t = f(t_1,\dots,t_i,\dots,t_n)$, $s = f(t_1,\dots,s_i,\dots,t_n)$, $i \in \mu(f)$, and $t_i \hookrightarrow_{\{\ell\to r\},\mu} s_i$. The induction hypothesis implies $\mathcal{I}_2(t_i) \hookrightarrow^+_{\{\ell\to r\}\cup\mathcal{C}_\varepsilon,\mu} \mathcal{I}_2(s_i)$ and hence, $\mathcal{I}_2(t) \hookrightarrow^+_{\{\ell\to r\}\cup\mathcal{C}_\varepsilon,\mu} \mathcal{I}_2(s)$. Finally, we consider the case $\mathrm{root}(t) \notin \Delta$. We have $\mathcal{I}_2(t) = \mathcal{C}omp(\{\dots\} \cup \mathcal{R}ed_2(t))$. Since $\mathcal{I}_2(s) \in \mathcal{R}ed_2(t)$, we obtain $\mathcal{C}omp(\{\dots\} \cup \mathcal{R}ed_2(t)) \hookrightarrow^+_{\mathcal{C}_\varepsilon,\mu} \mathcal{I}_2(s)$. ◻

Now we prove Thm. 21. First, we consider the case $e = \mathbf{t}$. If $(\mathcal{P},\mathcal{R},\mu,\mathbf{t})$ is not finite, then there is an infinite $(\mathcal{P},\mathcal{R},\mu)$-chain $s_1 \to t_1, s_2 \to t_2, \dots$ with $t_i\sigma \hookrightarrow^*_{\mathcal{R},\mu} s_{i+1}\sigma$ for all i. Moreover, all $t_i\sigma$ are terminating.

If $\mathcal{P} \cup \mathcal{U}^{\blacktriangleright}(\mathcal{P}, \mathcal{R}, \mu)$ is strongly conservative, we apply $\mathcal{I}_2$ and let Δ consist of all "usable" symbols, i.e., all symbols f where $Rls(f) \subseteq \mathcal{U}^{\blacktriangleright}(\mathcal{P}, \mathcal{R}, \mu)$. Note that the application of $\mathcal{I}_2$ is always possible since all terms $t_i\sigma$ and $s_{i+1}\sigma$ are terminating due to the minimality of the chain. If $\mathcal{P} \cup \mathcal{U}^{\blacktriangleright}(\mathcal{P}, \mathcal{R}, \mu)$ is not strongly conservative, we apply $\mathcal{I}_1$ and let Δ consist of all "usable" symbols again, i.e., all symbols f where $Rls(f) \subseteq \mathcal{U}^{\rhd}(\mathcal{P}, \mathcal{R}, \mu)$. Using Lemma 29 (4) (resp. Lemma 28 (4)), we obtain $\mathcal{I}_2(t_i\sigma) \hookrightarrow^*_{\mathcal{U}^{\blacktriangleright}(\mathcal{P},\mathcal{R},\mu) \cup \mathcal{C}_\varepsilon, \mu} \mathcal{I}_2(s_{i+1}\sigma)$ (resp. $\mathcal{I}_1(t_i\sigma) \hookrightarrow^*_{\mathcal{U}^{\rhd}(\mathcal{P},\mathcal{R},\mu) \cup \mathcal{C}_\varepsilon, \mu}$ $\mathcal{I}_1(s_{i+1}\sigma)$).

Moreover, by the definition of $\mathcal{U}^{\blacktriangleright}(\mathcal{P}, \mathcal{R}, \mu)$ (resp. $\mathcal{U}^{\rhd}(\mathcal{P}, \mathcal{R}, \mu)$), for all non-variable subterms t'_i at active positions of t_i, we have $\mathrm{root}(t'_i) \in \Delta$ (resp. all t_i are terms over the signature Δ). So by Lemma 29 (2) and (3) we get $t_i\sigma_{t_i} = [t_i, \sigma] = \mathcal{I}_2(t_i\sigma) \hookrightarrow^*_{\mathcal{U}^{\blacktriangleright}(\mathcal{P},\mathcal{R},\mu) \cup \mathcal{C}_\varepsilon} \mathcal{I}_2(s_{i+1}\sigma) \hookrightarrow^*_{\mathcal{C}_\varepsilon, \mu} [s_{i+1}, \sigma] = s_{i+1}\sigma_{s_{i+1}} \hookrightarrow^*_{\mathcal{C}_\varepsilon, \mu} s_{i+1}\sigma_{t_{i+1}}$ similar to the proof of Lemma 29 (3), where we now use the fact that $\mathcal{P}$ is strongly conservative. Correspondingly, by Lemma 28 (2) and (3) we obtain $t_i\,\mathcal{I}_1(\sigma) = \mathcal{I}_1(t_i\sigma) \hookrightarrow^*_{\mathcal{U}^{\rhd}(\mathcal{P},\mathcal{R},\mu) \cup \mathcal{C}_\varepsilon} \mathcal{I}_1(s_{i+1}\sigma) \hookrightarrow^*_{\mathcal{C}_\varepsilon, \mu} s_{i+1}\,\mathcal{I}_1(\sigma)$.

Thus, we have

- $s_i\sigma_{t_i} \succ t_i\sigma_{t_i}$ (resp. $s_i\,\mathcal{I}_1(\sigma) \succ t_i\,\mathcal{I}_1(\sigma)$) for all i where $s_i \to t_i \in \mathcal{P}_\succ$
- $s_i\sigma_{t_i} \succsim t_i\sigma_{t_i}$ (resp. $s_i\,\mathcal{I}_1(\sigma) \succsim t_i\,\mathcal{I}_1(\sigma)$) for all other i
- $t_i\sigma_{t_i} \succsim s_{i+1}\sigma_{t_{i+1}}$ (resp. $t_i\,\mathcal{I}_1(\sigma) \succsim s_{i+1}\,\mathcal{I}_1(\sigma)$) for all i

Since $\succ$ is well founded and compatible with $\succsim$, dependency pairs from $\mathcal{P}_\succ$ cannot occur infinitely often in this chain. Thus, there is an $n \geq 0$ such that all pairs $s_i \to t_i$ with $i \geq n$ are from $\mathcal{P} \setminus \mathcal{P}_\succ$. Therefore, if we omit the first $n-1$ pairs from the original chain, we obtain an infinite $(\mathcal{P} \setminus \mathcal{P}_\succ, \mathcal{R}, \mu)$-chain $s_n \to t_n, s_{n+1} \to t_{n+1}, \ldots$ Hence, $(\mathcal{P} \setminus \mathcal{P}_\succ, \mathcal{R}, \mu, \mathbf{t})$ is not finite either.

Now, we consider the case $e = \mathbf{i}$. Case (iii) of the theorem (i.e., the case where we use $\mathcal{U}^{\rhd}(\mathcal{P}, \mathcal{R}, \mu)$) is proved as for full termination. The only difference is that we now define $\mathcal{I}_2(t)$ for *innermost* terminating terms t and let $\mathcal{R}ed_2(t) = \{\mathcal{I}_2(t') \mid t \overset{\mathsf{i}}{\hookrightarrow}_{\mathcal{R},\mu} t'\}$. Then Lemma 29 can be proved as before for all *innermost* terminating terms $t, s, t\sigma$.

Thus, we now consider Case (ii) where $\mathcal{U}^{\blacktriangleright}(\mathcal{P}, \mathcal{R}, \mu)$ is conservative, but not necessarily strongly conservative. We show the following claim (similar to [16, Lemma 23]). Here, $\mathcal{U}^{\blacktriangleright}(t, \mu) = \bigcup_{f \in \mathcal{F}^\mu(t), f \blacktriangleright_{\mathcal{R},\mu} g} Rls(g)$.

> Let t, v be terms with $t\sigma \overset{\mathsf{i}}{\hookrightarrow}_{\mathcal{R},\mu} v$ and let $\sigma(x)$ be normal for all $x \in \mathcal{V}^\mu(t)$. Then $t\sigma \overset{\mathsf{i}}{\hookrightarrow}_{\mathcal{U}^{\blacktriangleright}(t,\mu)} v$. Moreover, there is a term u and a substitution σ' such that $v = u\sigma'$, $\sigma'(x)$ is normal for all $x \in \mathcal{V}^\mu(u)$, and $\mathcal{U}^{\blacktriangleright}(u, \mu) \subseteq \mathcal{U}^{\blacktriangleright}(t, \mu)$. (21)

First note that (21) is enough to prove the theorem in Case (ii). The reason is that by induction, (21) can be used to show that $t\sigma \overset{\mathsf{i}}{\hookrightarrow}^*_{\mathcal{R},\mu} v$ implies $t\sigma \overset{\mathsf{i}}{\hookrightarrow}^*_{\mathcal{U}^{\blacktriangleright}(t,\mu),\mu} v$. Hence, if $(\mathcal{P}, \mathcal{R}, \mu, \mathbf{i})$ is not finite, then there is an infinite innermost $(\mathcal{P}, \mathcal{R}, \mu)$-chain $s_1 \to t_1, s_2 \to t_2, \ldots$ with $t_i\sigma \overset{\mathsf{i}}{\hookrightarrow}^*_{\mathcal{R},\mu} s_{i+1}\sigma$ for all i and all $s_i\sigma$ are in normal form. With the claim above and conservativity of the DP $s_i \to t_i$, this means $t_i\sigma \overset{\mathsf{i}}{\hookrightarrow}^*_{\mathcal{U}^{\blacktriangleright}(t_i,\mu),\mu} s_{i+1}\sigma$ and thus, $t_i\sigma \overset{\mathsf{i}}{\hookrightarrow}^*_{\mathcal{U}^{\blacktriangleright}(\mathcal{P},\mathcal{R},\mu),\mu} s_{i+1}\sigma$. Then one continues as in the proof for full termination.

Now we prove Claim (21) by induction on the position of the reduction $t\sigma \xhookrightarrow{\mathrm{i}}_{\mathcal{R},\mu} v$. This position must be a non-variable position of t because $\sigma(x)$ is normal for all $x \in \mathcal{V}^\mu(t)$. So t has the form $f(t_1,\ldots,t_n)$.

If the reduction is at the root position, then we have $t\sigma = \ell\sigma' \xhookrightarrow{\mathrm{i}}_{\mathcal{R}} r\sigma' = v$ where $\ell \to r \in \mathcal{U}^\blacktriangleright(t,\mu)$. Note that $\sigma'(x)$ is normal for all $x \in \mathcal{V}^\mu(\ell)$ due to the innermost strategy and thus, this also holds for all $x \in \mathcal{V}^\mu(r)$ since the rule $\ell \to r$ is conservative. By defining $u = r$, we obtain $v = u\sigma'$ and $\mathcal{U}^\blacktriangleright(u,\mu) \subseteq \mathcal{U}^\blacktriangleright(t,\mu)$.

Otherwise, $t\sigma = f(t_1\sigma,\ldots,t_i\sigma,\ldots,t_n\sigma) \xhookrightarrow{\mathrm{i}}_{\mathcal{R},\mu} f(t_1\sigma,\ldots,v_i,\ldots,t_n\sigma) = v$ where $i \in \mu(f)$ and $t_i\sigma \xhookrightarrow{\mathrm{i}}_{\mathcal{R},\mu} v_i$. The induction hypothesis implies $t_i\sigma \xhookrightarrow{\mathrm{i}}_{\mathcal{U}^\blacktriangleright(t_i,\mu),\mu} v_i$ and thus, $t\sigma \xhookrightarrow{\mathrm{i}}_{\mathcal{U}^\blacktriangleright(t_i,\mu),\mu} v$. As $\mathcal{U}^\blacktriangleright(t_i,\mu) \subseteq \mathcal{U}^\blacktriangleright(t,\mu)$, we also have $t\sigma \xhookrightarrow{\mathrm{i}}_{\mathcal{U}^\blacktriangleright(t,\mu),\mu} v$.

By the induction hypothesis there is some term u_i and some substitution σ_i with $v_i = u_i\sigma_i$ and $\sigma_i(x)$ is normal for all $x \in \mathcal{V}^\mu(u_i)$. Let u_i' result from u_i by replacing its variables x by corresponding fresh variables x'. We define $\sigma'(x') = \sigma_i(x)$ for all these fresh variables and $\sigma'(x) = \sigma(x)$ for all $x \in \mathcal{V}(t)$. Then for $u = f(t_1,\ldots,u_i',\ldots,t_n)$ we obtain $v = u\sigma'$ and $\sigma'(x)$ is normal for all $x \in \mathcal{V}^\mu(u)$. Obviously, we have $\mathcal{U}^\blacktriangleright(u_i',\mu) = \mathcal{U}^\blacktriangleright(u_i,\mu)$ and the induction hypothesis implies $\mathcal{U}^\blacktriangleright(u_i,\mu) \subseteq \mathcal{U}^\blacktriangleright(t_i,\mu) \subseteq \mathcal{U}^\blacktriangleright(t,\mu)$. Since $\mathcal{U}^\blacktriangleright(u,\mu)$ differs from $\mathcal{U}^\blacktriangleright(t,\mu)$ only by containing $\mathcal{U}^\blacktriangleright(u_i',\mu)$ instead of $\mathcal{U}^\blacktriangleright(t_i,\mu)$, we also obtain $\mathcal{U}^\blacktriangleright(u,\mu) \subseteq \mathcal{U}^\blacktriangleright(t,\mu)$. □

D Proof of Thm. 24

We only prove the case $e = \mathbf{t}$. For $e = \mathbf{i}$ the proof is similar.

Let $\ldots, v \to w, s \to t, v' \to w', \ldots$ be an $(\mathcal{P}',\mathcal{R},\mu)$-chain. By Def. 11 there exist a substitution σ such that $w\sigma \hookrightarrow^*_{\mathcal{R},\mu} s\sigma$ and $t\sigma \hookrightarrow^*_{\mathcal{R},\mu} v'\sigma$ and $t\sigma$ is terminating. It is easy to show that then $s\sigma = \textsc{Ren}^\mu(\textsc{Cap}^\mu(w))\sigma\delta$ for some substitution δ whose domain only contains fresh variables introduced by $\textsc{Ren}^\mu$ resp. $\textsc{Cap}^\mu$. Thus we can assume w.l.o.g. that σ is equal to δ on all these fresh variables. Hence, we have $s\sigma = \textsc{Ren}^\mu(\textsc{Cap}^\mu(w))\sigma$, which implies that there is an mgu θ of s and $\textsc{Ren}^\mu(\textsc{Cap}^\mu(w))$ with $\sigma = \theta\tau$ for some substitution τ. This shows that $s\theta \to t\theta \in \overline{\mathcal{P}}$ and thus we obtain $w\sigma \hookrightarrow^*_{\mathcal{R},\mu} s\sigma = s\theta\tau$ and $t\theta\tau = t\sigma \hookrightarrow^*_{\mathcal{R},\mu} v'\sigma$. Note that $t\theta\tau = t\sigma$ is terminating. Thus, $\ldots, v \to w, s\theta \to t\theta, v' \to w', \ldots$ is a $(\mathcal{P} \cup \overline{\mathcal{P}},\mathcal{R},\mu)$-chain. In this way, we can replace every occurrence of $s \to t$ in a chain by a pair from $\overline{\mathcal{P}}$ except for the first pair. However, if $s \to t, v_1 \to w_1, v_2 \to w_2, \ldots$ is an infinite chain, then so is $v_1 \to w_1, v_2 \to w_2, \ldots$. □

Aachener Informatik-Berichte

This list contains all technical reports published during the past five years. A complete list of reports dating back to 1987 is available from `http://aib.informatik.rwth-aachen.de/`**. To obtain copies consult the above URL or send your request to: Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Email:** `biblio@informatik.rwth-aachen.de`

2003-01 * Jahresbericht 2002
2003-02 Jürgen Giesl, René Thiemann: Size-Change Termination for Term Rewriting
2003-03 Jürgen Giesl, Deepak Kapur: Deciding Inductive Validity of Equations
2003-04 Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, Stephan Falke: Improving Dependency Pairs
2003-05 Christof Löding, Philipp Rohde: Solving the Sabotage Game is PSPACE-hard
2003-06 Franz Josef Och: Statistical Machine Translation: From Single-Word Models to Alignment Templates
2003-07 Horst Lichter, Thomas von der Maßen, Alexander Nyßen, Thomas Weiler: Vergleich von Ansätzen zur Feature Modellierung bei der Softwareproduktlinienentwicklung
2003-08 Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, Stephan Falke: Mechanizing Dependency Pairs
2004-01 * Fachgruppe Informatik: Jahresbericht 2003
2004-02 Benedikt Bollig, Martin Leucker: Message-Passing Automata are expressively equivalent to EMSO logic
2004-03 Delia Kesner, Femke van Raamsdonk, Joe Wells (eds.): HOR 2004 – 2nd International Workshop on Higher-Order Rewriting
2004-04 Slim Abdennadher, Christophe Ringeissen (eds.): RULE 04 – Fifth International Workshop on Rule-Based Programming
2004-05 Herbert Kuchen (ed.): WFLP 04 – 13th International Workshop on Functional and (Constraint) Logic Programming
2004-06 Sergio Antoy, Yoshihito Toyama (eds.): WRS 04 – 4th International Workshop on Reduction Strategies in Rewriting and Programming
2004-07 Michael Codish, Aart Middeldorp (eds.): WST 04 – 7th International Workshop on Termination
2004-08 Klaus Indermark, Thomas Noll: Algebraic Correctness Proofs for Compiling Recursive Function Definitions with Strictness Information
2004-09 Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Parameterized Power Domination Complexity
2004-10 Zinaida Benenson, Felix C. Gärtner, Dogan Kesdogan: Secure Multi-Party Computation with Security Modules

2005-01 * Fachgruppe Informatik: Jahresbericht 2004
2005-02 Maximillian Dornseif, Felix C. Gärtner, Thorsten Holz, Martin Mink: An Offensive Approach to Teaching Information Security: “Aachen Summer School Applied IT Security”
2005-03 Jürgen Giesl, René Thiemann, Peter Schneider-Kamp: Proving and Disproving Termination of Higher-Order Functions
2005-04 Daniel Mölle, Stefan Richter, Peter Rossmanith: A Faster Algorithm for the Steiner Tree Problem
2005-05 Fabien Pouget, Thorsten Holz: A Pointillist Approach for Comparing Honeypots
2005-06 Simon Fischer, Berthold Vöcking: Adaptive Routing with Stale Information
2005-07 Felix C. Freiling, Thorsten Holz, Georg Wicherski: Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks
2005-08 Joachim Kneis, Peter Rossmanith: A New Satisfiability Algorithm With Applications To Max-Cut
2005-09 Klaus Kursawe, Felix C. Freiling: Byzantine Fault Tolerance on General Hybrid Adversary Structures
2005-10 Benedikt Bollig: Automata and Logics for Message Sequence Charts
2005-11 Simon Fischer, Berthold Vöcking: A Counterexample to the Fully Mixed Nash Equilibrium Conjecture
2005-12 Neeraj Mittal, Felix Freiling, S. Venkatesan, Lucia Draque Penso: Efficient Reductions for Wait-Free Termination Detection in Faulty Distributed Systems
2005-13 Carole Delporte-Gallet, Hugues Fauconnier, Felix C. Freiling: Revisiting Failure Detection and Consensus in Omission Failure Environments
2005-14 Felix C. Freiling, Sukumar Ghosh: Code Stabilization
2005-15 Uwe Naumann: The Complexity of Derivative Computation
2005-16 Uwe Naumann: Syntax-Directed Derivative Code (Part I: Tangent-Linear Code)
2005-17 Uwe Naumann: Syntax-directed Derivative Code (Part II: Intraprocedural Adjoint Code)
2005-18 Thomas von der Maßen, Klaus Müller, John MacGregor, Eva Geisberger, Jörg Dörr, Frank Houdek, Harbhajan Singh, Holger Wußmann, Hans-Veit Bacher, Barbara Paech: Einsatz von Features im Software-Entwicklungsprozess - Abschlussbericht des GI-Arbeitskreises “Features”
2005-19 Uwe Naumann, Andre Vehreschild: Tangent-Linear Code by Augmented LL-Parsers

2005-20 Felix C. Freiling, Martin Mink: Bericht über den Workshop zur Ausbildung im Bereich IT-Sicherheit Hochschulausbildung, berufliche Weiterbildung, Zertifizierung von Ausbildungsangeboten am 11. und 12. August 2005 in Köln organisiert von RWTH Aachen in Kooperation mit BITKOM, BSI, DLR und Gesellschaft fuer Informatik (GI) e.V.

2005-21 Thomas Noll, Stefan Rieger: Optimization of Straight-Line Code Revisited

2005-22 Felix Freiling, Maurice Herlihy, Lucia Draque Penso: Optimal Randomized Fair Exchange with Secret Shared Coins

2005-23 Heiner Ackermann, Alantha Newman, Heiko Röglin, Berthold Vöcking: Decision Making Based on Approximate and Smoothed Pareto Curves

2005-24 Alexander Becher, Zinaida Benenson, Maximillian Dornseif: Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks

2006-01 * Fachgruppe Informatik: Jahresbericht 2005

2006-02 Michael Weber: Parallel Algorithms for Verification of Large Systems

2006-03 Michael Maier, Uwe Naumann: Intraprocedural Adjoint Code Generated by the Differentiation-Enabled NAGWare Fortran Compiler

2006-04 Ebadollah Varnik, Uwe Naumann, Andrew Lyons: Toward Low Static Memory Jacobian Accumulation

2006-05 Uwe Naumann, Jean Utke, Patrick Heimbach, Chris Hill, Derya Ozyurt, Carl Wunsch, Mike Fagan, Nathan Tallent, Michelle Strout: Adjoint Code by Source Transformation with OpenAD/F

2006-06 Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Divide-and-Color

2006-07 Thomas Colcombet, Christof Löding: Transforming structures by set interpretations

2006-08 Uwe Naumann, Yuxiao Hu: Optimal Vertex Elimination in Single-Expression-Use Graphs

2006-09 Tingting Han, Joost-Pieter Katoen: Counterexamples in Probabilistic Model Checking

2006-10 Mesut Günes, Alexander Zimmermann, Martin Wenig, Jan Ritzerfeld, Ulrich Meis: From Simulations to Testbeds - Architecture of the Hybrid MCG-Mesh Testbed

2006-11 Bastian Schlich, Michael Rohrbach, Michael Weber, Stefan Kowalewski: Model Checking Software for Microcontrollers

2006-12 Benedikt Bollig, Joost-Pieter Katoen, Carsten Kern, Martin Leucker: Replaying Play in and Play out: Synthesis of Design Models from Scenarios by Learning

2006-13 Wong Karianto, Christof Löding: Unranked Tree Automata with Sibling Equalities and Disequalities

2006-14 Danilo Beuche, Andreas Birk, Heinrich Dreier, Andreas Fleischmann, Heidi Galle, Gerald Heller, Dirk Janzen, Isabel John, Ramin Tavakoli Kolagari, Thomas von der Maßen, Andreas Wolfram: Report of the GI Work Group “Requirements Management Tools for Product Line Engineering”

2006-15 Sebastian Ullrich, Jakob T. Valvoda, Torsten Kuhlen: Utilizing optical sensors from mice for new input devices
2006-16 Rafael Ballagas, Jan Borchers: Selexels: a Conceptual Framework for Pointing Devices with Low Expressiveness
2006-17 Eric Lee, Henning Kiel, Jan Borchers: Scrolling Through Time: Improving Interfaces for Searching and Navigating Continuous Audio Timelines
2007-01 * Fachgruppe Informatik: Jahresbericht 2006
2007-02 Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, René Thiemann, and Harald Zankl: SAT Solving for Termination Analysis with Polynomial Interpretations
2007-03 Jürgen Giesl, René Thiemann, Stephan Swiderski, and Peter Schneider-Kamp: Proving Termination by Bounded Increase
2007-04 Jan Buchholz, Eric Lee, Jonathan Klein, and Jan Borchers: coJIVE: A System to Support Collaborative Jazz Improvisation
2007-05 Uwe Naumann: On Optimal DAG Reversal
2007-06 Joost-Pieter Katoen, Thomas Noll, and Stefan Rieger: Verifying Concurrent List-Manipulating Programs by LTL Model Checking
2007-07 Alexander Nyßen, Horst Lichter: MeDUSA - MethoD for UML2-based Design of Embedded Software Applications
2007-08 Falk Salewski and Stefan Kowalewski: Achieving Highly Reliable Embedded Software: An empirical evaluation of different approaches
2007-09 Tina Kraußer, Heiko Mantel, and Henning Sudbrock: A Probabilistic Justification of the Combining Calculus under the Uniform Scheduler Assumption
2007-10 Martin Neuhäußer, Joost-Pieter Katoen: Bisimulation and Logical Preservation for Continuous-Time Markov Decision Processes
2007-11 Klaus Wehrle (editor): 6. Fachgespräch Sensornetzwerke
2007-12 Uwe Naumann: An L-Attributed Grammar for Adjoint Code
2007-13 Uwe Naumann, Michael Maier, Jan Riehme, and Bruce Christianson: Second-Order Adjoints by Source Code Manipulation of Numerical Programs
2007-14 Jean Utke, Uwe Naumann, Mike Fagan, Nathan Tallent, Michelle Strout, Patrick Heimbach, Chris Hill, and Carl Wunsch: OpenAD/F: A Modular, Open-Source Tool for Automatic Differentiation of Fortran Codes
2007-15 Volker Stolz: Temporal assertions for sequential and concurrent programs
2007-16 Sadeq Ali Makram, Mesut Günec, Martin Wenig, Alexander Zimmermann: Adaptive Channel Assignment to Support QoS and Load Balancing for Wireless Mesh Networks
2007-17 René Thiemann: The DP Framework for Proving Termination of Term Rewriting
2007-20 Joost-Pieter Katoen, Daniel Klink, Martin Leucker, and Verena Wolf: Three-Valued Abstraction for Probabilistic Systems
2007-21 Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre: Compositional Modeling and Minimization of Time-Inhomogeneous Markov Chains

2007-22 Heiner Ackermann, Paul W. Goldberg, Vahab S. Mirrokni, Heiko Röglin, and Berthold Vöcking: Uncoordinated Two-Sided Markets

2008-01 * Fachgruppe Informatik: Jahresbericht 2007

2008-02 Henrik Bohnenkamp, Marielle Stoelinga: Quantitative Testing

2008-03 Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, René Thiemann, Harald Zankl: Maximal Termination

2008-04 Uwe Naumann, Jan Riehme: Sensitivity Analysis in Sisyphe with the AD-Enabled NAGWare Fortran Compiler

2008-05 Frank G. Radmacher: An Automata Theoretic Approach to the Theory of Rational Tree Relations

2008-06 Uwe Naumann, Laurent Hascoet, Chris Hill, Paul Hovland, Jan Riehme, Jean Utke: A Framework for Proving Correctness of Adjoint Message Passing Programs

2008-08 George B. Mertzios, Stavros D. Nikolopoulos: The λ-cluster Problem on Parameterized Interval Graphs

2008-09 George B. Mertzios, Walter Unger: An optimal algorithm for the k-fixed-endpoint path cover on proper interval graphs

2008-10 George B. Mertzios, Walter Unger: Preemptive Scheduling of Equal-Length Jobs in Polynomial Time

2008-11 George B. Mertzios: Fast Convergence of Routing Games with Splittable Flows

2008-12 Joost-Pieter Katoen, Daniel Klink, Martin Leucker, Verena Wolf: Abstraction for stochastic systems by Erlang's method of stages

* These reports are only available as a printed version.
Please contact `biblio@informatik.rwth-aachen.de` to obtain copies.