Dependency Pairs for Equational Rewriting*

Jürgen Giesl[1] and Deepak Kapur[2]

[1] LuFG Informatik II, RWTH Aachen, Ahornstr. 55, 52074 Aachen, Germany,
giesl@informatik.rwth-aachen.de

[2] Computer Science Dept., University of New Mexico, Albuquerque, NM 87131, USA
kapur@cs.unm.edu

Abstract. The dependency pair technique of Arts and Giesl [1–3] for termination proofs of term rewrite systems (TRSs) is extended to rewriting modulo equations. Up to now, such an extension was only known in the special case of AC-rewriting [15, 17]. In contrast to that, the proposed technique works for arbitrary non-collapsing equations (satisfying a certain linearity condition). With the proposed approach, it is now possible to perform automated termination proofs for many systems where this was not possible before. In other words, the power of dependency pairs can now also be used for rewriting modulo equations.

1 Introduction

Termination of term rewriting (e.g., [1–3, 9, 22]) and termination of rewriting modulo associativity and commutativity equations (e.g., [8, 13, 14, 20, 21]) have been extensively studied. For equations other than AC-axioms, however, there are only a few techniques available to prove termination (e.g., [6, 10, 16, 18]).

This paper presents an extension of the dependency pair approach [1–3] to rewriting modulo equations. In the special case of AC-axioms, our technique corresponds to the methods of [15, 17], but in contrast to these methods, our technique can also be used if the equations are not AC-axioms. This allows much more automated termination proofs for equational rewrite systems than those possible with directly applying simplification orderings for equational rewriting (like equational polynomial orderings or AC-versions of path orderings).

We first review dependency pairs for ordinary term rewriting in Sect. 2. In Sect. 3, we show why a straightforward extension of dependency pairs to rewriting modulo equations is not possible. Therefore, we follow an idea similar to the one of [17] for AC-axioms: We consider a restricted form of equational rewriting, which is more suitable for termination proofs with dependency pairs.

In Sect. 4, we show how to ensure that termination of this restricted equational rewrite relation is equivalent to termination of full rewriting modulo equations. Under certain conditions on the equations $\mathcal{E}$, we show how to compute an

* *Proceedings of the 12th International Conference on Rewriting Techniques and Applications, RTA-2001*, Utrecht, The Netherlands, Lecture Notes in Computer Science, Springer-Verlag. Supported by the Deutsche Forschungsgemeinschaft Grant GI 274/4-1 and the National Science Foundation Grants nos. CCR-9996150, CDA-9503064, CCR-9712396.

extended rewrite system $Ext_{\mathcal{E}}(\mathcal{R})$ from the given TRS $\mathcal{R}$ such that the restricted rewrite relation of $Ext_{\mathcal{E}}(\mathcal{R})$ modulo $\mathcal{E}$ is terminating iff $\mathcal{R}$ is terminating modulo $\mathcal{E}$. This is proved for (almost) arbitrary $\mathcal{E}$-rewriting, thus generalizing a related result for AC-rewriting. This general result may be of independent interest, and may also be useful in investigating other properties of $\mathcal{E}$-rewriting. Finally, in Sect. 5, we extend the dependency pair approach to rewriting modulo equations.

2 Dependency Pairs for Ordinary Rewriting

The dependency pair approach allows the use of standard methods like simplification orderings [9, 22] for automated termination proofs where they were not applicable before. In this section we briefly summarize the basic concepts of this approach. All results in this section are due to Arts and Giesl and we refer to [1–3] for further details, refinements, and explanations.

In contrast to the standard techniques for termination proofs, which compare left and right-hand sides of rules, in this approach one concentrates on the subterms in the right-hand sides that have a defined[1] root symbol, because these are the only terms responsible for starting new reductions.

More precisely, for every rule $f(s_1, \ldots, s_n) \to C[g(t_1, \ldots, t_m)]$ (where f and g are defined symbols), we compare the argument tuples $s_1, \ldots, s_n$ and $t_1, \ldots, t_m$. To avoid the handling of tuples, for every defined symbol f, we introduce a fresh *tuple* symbol F. To ease readability, we assume that the original signature consists of lower case function symbols only, whereas the tuple symbols are denoted by the corresponding upper case symbols. Now instead of the tuples $s_1, \ldots, s_n$ and $t_1, \ldots, t_m$ we compare the *terms* $F(s_1, \ldots, s_n)$ and $G(t_1, \ldots, t_m)$.

Definition 1 (Dependency Pair [1–3]). *If $f(s_1, \ldots, s_n) \to C[g(t_1, \ldots, t_m)]$ is a rule of a TRS $\mathcal{R}$ and g is a defined symbol, then $\langle F(s_1, \ldots, s_n), G(t_1, \ldots, t_m)\rangle$ is a* dependency pair *of $\mathcal{R}$.*

Example 2. As an example, consider the TRS $\{\mathsf{a} + \mathsf{b} \to \mathsf{a} + (\mathsf{b} + \mathsf{c})\}$, cf. [17]. Termination of this system cannot be shown by simplification orderings, since the left-hand side of the rule is embedded in the right-hand side. In this system, the defined symbol is $+$ and thus, we obtain the dependency pairs $\langle \mathsf{P}(\mathsf{a}, \mathsf{b}), \mathsf{P}(\mathsf{a}, \mathsf{b}{+}\mathsf{c})\rangle$ and $\langle \mathsf{P}(\mathsf{a}, \mathsf{b}), \mathsf{P}(\mathsf{b}, \mathsf{c})\rangle$ (where P is the tuple symbol for the plus-function "$+$").

Arts and Giesl developed the following new termination criterion. As usual, a quasi-ordering $\succsim$ is a reflexive and transitive relation, and we say that an ordering $>$ is *compatible* with $\succsim$ if we have $> \circ \succsim \subseteq >$ or $\succsim \circ > \subseteq >$.

Theorem 3 (Termination with Dependency Pairs [1–3]). *A TRS $\mathcal{R}$ is terminating iff there exists a weakly monotonic quasi-ordering $\succsim$ and a well-founded ordering $>$ compatible with $\succsim$, where both $\succsim$ and $>$ are closed under substitution, such that*

[1] Root symbols of left-hand sides are *defined* and all other functions are *constructors*.

(1) $s > t$ *for all dependency pairs* $\langle s, t \rangle$ *of* $\mathcal{R}$ *and*
(2) $l \gtrsim r$ *for all rules* $l \to r$ *of* $\mathcal{R}$.

Consider the TRS from Ex. 2 again. In order to prove its termination according to Thm. 3, we have to find a suitable quasi-ordering $\gtrsim$ and ordering $>$ such that $\mathsf{P}(\mathsf{a},\mathsf{b}) > \mathsf{P}(\mathsf{a},\mathsf{b}+\mathsf{c})$, $\mathsf{P}(\mathsf{a},\mathsf{b}) > \mathsf{P}(\mathsf{b},\mathsf{c})$, and $\mathsf{a}+\mathsf{b} \gtrsim \mathsf{a}+(\mathsf{b}+\mathsf{c})$.

Most standard orderings amenable to automation are *strongly* monotonic (cf. e.g. [9, 22]), whereas here we only need *weak* monotonicity. Hence, before synthesizing a suitable ordering, some of the arguments of function symbols may be eliminated, cf. [3]. For example, in our inequalities, one may eliminate the first argument of $+$. Then every term $s+t$ in the inequalities is replaced by $+'(t)$ (where $+'$ is a new unary function symbol). By comparing the terms resulting from this replacement instead of the original terms, we can take advantage of the fact that $+$ does not have to be strongly monotonic in its first argument. Note that there are only finitely many possibilities to eliminate arguments of function symbols. Therefore all these possibilities can be checked automatically.

In this way, we obtain the inequalities $\mathsf{P}(\mathsf{a},\mathsf{b}) > \mathsf{P}(\mathsf{a},+'(\mathsf{c}))$, $\mathsf{P}(\mathsf{a},\mathsf{b}) > \mathsf{P}(\mathsf{b},\mathsf{c})$, and $+'(\mathsf{b}) \gtrsim +'(+'(\mathsf{c}))$. These inequalities are satisfied by the recursive path ordering (rpo) [9] with the precedence $\mathsf{a} \sqsupset \mathsf{b} \sqsupset \mathsf{c} \sqsupset +'$ (i.e., we choose $\gtrsim$ to be $\gtrsim_{rpo}$ and $>$ to be $\succ_{rpo}$). So termination of this TRS can now be proved automatically. For implementations of the dependency pair approach see [4, 7].

3 Rewriting Modulo Equations

For a set $\mathcal{E}$ of equations between terms, we write $s \to_{\mathcal{E}} t$ if there exist an equation $l \approx r$ in $\mathcal{E}$, a substitution σ, and a context C such that $s = C[l\sigma]$ and $t = C[r\sigma]$. The symmetric closure of $\to_{\mathcal{E}}$ is denoted by $\vdash\dashv_{\mathcal{E}}$ and the transitive reflexive closure of $\vdash\dashv_{\mathcal{E}}$ is denoted by $\sim_{\mathcal{E}}$. In the following, we restrict ourselves to equations $\mathcal{E}$ where $\sim_{\mathcal{E}}$ is decidable.

Definition 4 (Rewriting Modulo Equations). *Let* $\mathcal{R}$ *be a TRS and let* $\mathcal{E}$ *be a set of equations. A term* s *rewrites to a term* t *modulo* $\mathcal{E}$*, denoted* $s \to_{\mathcal{R}/\mathcal{E}} t$*, iff there exist terms* s' *and* t' *such that* $s \sim_{\mathcal{E}} s' \to_{\mathcal{R}} t' \sim_{\mathcal{E}} t$*. The TRS* $\mathcal{R}$ *is called terminating modulo* $\mathcal{E}$ *iff there does not exist an infinite* $\to_{\mathcal{R}/\mathcal{E}}$ *reduction.*

Example 5. An interesting special case are equations $\mathcal{E}$ which state that certain function symbols are associative and commutative (AC). As an example, consider the TRS $\mathcal{R} = \{\mathsf{a}+\mathsf{b} \to \mathsf{a}+(\mathsf{b}+\mathsf{c})\}$ again and let $\mathcal{E}$ consist of the associativity and commutativity axioms for $+$, i.e., $\mathcal{E} = \{x_1 + x_2 \approx x_2 + x_1, x_1 + (x_2 + x_3) \approx (x_1 + x_2) + x_3\}$, cf. [17]. $\mathcal{R}$ is not terminating modulo $\mathcal{E}$, since we have

$$\mathsf{a}+\mathsf{b} \to_{\mathcal{R}} \mathsf{a}+(\mathsf{b}+\mathsf{c}) \sim_{\mathcal{E}} (\mathsf{a}+\mathsf{b})+\mathsf{c} \to_{\mathcal{R}} (\mathsf{a}+(\mathsf{b}+\mathsf{c}))+\mathsf{c} \sim_{\mathcal{E}} ((\mathsf{a}+\mathsf{b})+\mathsf{c})+\mathsf{c} \to_{\mathcal{R}} \ldots$$

There are, however, many other sets of equations $\mathcal{E}$ apart from associativity and commutativity, which are also important in practice, cf. [11]. Hence, our aim is to extend dependency pairs to rewriting modulo (almost) arbitrary equations.

The soundness of dependency pairs for ordinary rewriting relies on the fact that whenever a term starts an infinite reduction, then one can also construct an infinite reduction where only *terminating or minimal non-terminating subterms* are reduced (i.e., one only applies rules to redexes without proper non-terminating subterms). The contexts of minimal non-terminating redexes can be completely disregarded. If a rule is applied at the root position of a minimal non-terminating subterm s (i.e., $s \to_{\mathcal{R}}^{\epsilon} t$ where ϵ denotes the root position), then s and each minimal non-terminating subterm t' of t correspond to a dependency pair. Hence, Thm. 3 (1) implies $s > t'$. If a rule is applied at a non-root position of a minimal non-terminating subterm s (i.e., $s \to_{\mathcal{R}}^{>\epsilon} t$), then we have $s \succsim t$ by Thm. 3 (2). However, due to the minimality of s, after finitely many such non-root rewrite steps, a rule must be applied at the root position of the minimal non-terminating term. Thus, every infinite reduction of minimal non-terminating subterms corresponds to an infinite $>$-sequence. This contradicts the well-foundedness of $>$.

So for ordinary rewriting, any infinite reduction from a minimal non-terminating subterm involves an $\mathcal{R}$-reduction at the root position. But as observed in [15], when extending the dependency pair approach to rewriting modulo equations, this is no longer true. For an illustration, consider Ex. 5 again, where $\mathsf{a} + (\mathsf{b} + \mathsf{c})$ is a minimal non-terminating term. However, in its infinite $\mathcal{R}/\mathcal{E}$-reduction no $\mathcal{R}$-step is ever applicable at the root position. (Instead one applies an $\mathcal{E}$-step at the root position and further $\mathcal{R}$- and $\mathcal{E}$-steps below the root.)

In the rest of the paper, from a rewrite system $\mathcal{R}$, we generate a new rewrite system $\mathcal{R}'$ with the following three properties: (i) the termination of a weaker form of rewriting by $\mathcal{R}'$ modulo $\mathcal{E}$ is equivalent to the termination of $\mathcal{R}$ modulo $\mathcal{E}$, (ii) every infinite reduction of a minimal non-terminating term in this weaker form of rewriting by $\mathcal{R}'$ modulo $\mathcal{E}$ involves a reduction step at the root level, and (iii) every such minimal non-terminating term has an infinite reduction where the variables of the $\mathcal{R}'$-rules are instantiated with terminating terms only.

4 $\mathcal{E}$-Extended Rewriting

We showed why the dependency pair approach cannot be extended to rewriting modulo equations directly. As a solution for this problem, we propose to consider a restricted form of rewriting modulo equations, i.e., the so-called *$\mathcal{E}$-extended $\mathcal{R}$-rewrite relation* $\to_{\mathcal{E}\backslash\mathcal{R}}$. (This approach was already taken in [17] for rewriting modulo AC.) The relation $\to_{\mathcal{E}\backslash\mathcal{R}}$ was originally introduced in [19] in order to circumvent the problems with infinite or impractically large $\mathcal{E}$-equivalence classes.[2]

Definition 6 ($\mathcal{E}$-extended $\mathcal{R}$-rewriting [19]). *Let $\mathcal{R}$ be a TRS and let $\mathcal{E}$ be a set of equations. The $\mathcal{E}$-extended $\mathcal{R}$-rewrite relation is defined as $s \to_{\mathcal{E}\backslash\mathcal{R}}^{\pi} t$ iff $s|_{\pi} \sim_{\mathcal{E}} l\sigma$ and $t = s[r\sigma]_{\pi}$ for some rule $l \to r$ in $\mathcal{R}$, some position π of s, and some substitution σ. We also write $\to_{\mathcal{E}\backslash\mathcal{R}}$ instead of $\to_{\mathcal{E}\backslash\mathcal{R}}^{\pi}$.*

[2] In [12], the relation $\to_{\mathcal{E}\backslash\mathcal{R}}$ is denoted "$\to_{\mathcal{R},\mathcal{E}}$".

To demonstrate the difference between $\to_{\mathcal{R}/\mathcal{E}}$ and $\to_{\mathcal{E}\backslash\mathcal{R}}$, consider Ex. 5 again. We have already seen that $\to_{\mathcal{R}/\mathcal{E}}$ is not terminating, since $\mathsf{a}+\mathsf{b} \to_{\mathcal{R}/\mathcal{E}} (\mathsf{a}+\mathsf{b})+\mathsf{c} \to_{\mathcal{R}/\mathcal{E}} ((\mathsf{a}+\mathsf{b})+\mathsf{c})+\mathsf{c} \to_{\mathcal{R}/\mathcal{E}} \ldots$ But $\to_{\mathcal{E}\backslash\mathcal{R}}$ is terminating, because $\mathsf{a}+\mathsf{b} \to_{\mathcal{E}\backslash\mathcal{R}} \mathsf{a}+(\mathsf{b}+\mathsf{c})$, which is a normal form w.r.t. $\to_{\mathcal{E}\backslash\mathcal{R}}$.

The above example also demonstrates that in general, termination of $\to_{\mathcal{E}\backslash\mathcal{R}}$ is not sufficient for termination of $\to_{\mathcal{R}/\mathcal{E}}$. In this section we will show how termination of $\to_{\mathcal{R}/\mathcal{E}}$ can nevertheless be ensured by only regarding an $\mathcal{E}$-extended rewrite relation induced by a larger $\mathcal{R}' \supseteq \mathcal{R}$.

For the special case of AC-rewriting, this problem can be solved by extending $\mathcal{R}$ as follows: Let $\mathcal{G}$ be the set of all AC-symbols and

$$Ext_{AC(\mathcal{G})} = \mathcal{R} \cup \{f(l,y) \to f(r,y) \mid l \to r \in \mathcal{R},\ root(l) = f \in \mathcal{G}\},$$

where y is a new variable not occurring in the respective rule $l \to r$. A similar extension has also been used in previous work on extending dependency pairs to AC-rewriting [17]. The reason is that for AC-equations $\mathcal{E}$, the termination of $\to_{\mathcal{R}/\mathcal{E}}$ is in fact equivalent to the termination of $\to_{\mathcal{E}\backslash Ext_{AC(\mathcal{G})}(\mathcal{R})}$.

For Ex. 5, we obtain $Ext_{AC(\mathcal{G})}(\mathcal{R}) = \{\mathsf{a}+\mathsf{b} \to \mathsf{a}+(\mathsf{b}+\mathsf{c}), (\mathsf{a}+\mathsf{b})+y \to (\mathsf{a}+(\mathsf{b}+\mathsf{c}))+y\}$. Thus, in order to prove termination of $\to_{\mathcal{R}/\mathcal{E}}$, it is now sufficient to verify termination of $\to_{\mathcal{E}\backslash Ext_{AC(\mathcal{G})}(\mathcal{R})}$.

The above extension of [19] only works for AC-axioms $\mathcal{E}$. A later paper [12] treats arbitrary equations, but it does not contain any definition for extensions $Ext_{\mathcal{E}}(\mathcal{R})$, and termination of $\to_{\mathcal{R}/\mathcal{E}}$ is always a prerequisite in [12]. The reason is that [12] and also subsequent work on symmetrization and coherence were devoted to the development of completion algorithms (i.e., here the goal was to generate a convergent rewrite system and not to investigate the termination behavior of possibly non-terminating TRSs). Thus, these papers did not compare the termination behavior of full rewriting modulo equations with the termination of restricted versions of rewriting modulo equations. In fact, [12] focuses on the notion of coherence, which is not suitable for our purpose since coherence of $\mathcal{E}\backslash\mathcal{R}$ modulo $\mathcal{E}$ does not imply that termination of $\to_{\mathcal{R}/\mathcal{E}}$ is equivalent to termination of $\to_{\mathcal{E}\backslash\mathcal{R}}$.[3]

To extend dependency pairs to rewriting modulo non-AC-equations $\mathcal{E}$, we have to compute extensions $Ext_{\mathcal{E}}(\mathcal{R})$ such that termination of $\to_{\mathcal{R}/\mathcal{E}}$ is equivalent to termination of $\to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})}$. The only restriction we will impose on the equations in $\mathcal{E}$ is that they must have *identical unique variables*. This requirement is satisfied by most practical examples where $\mathcal{R}/\mathcal{E}$ is terminating. As usual, a term t is called *linear* if no variable occurs more than once in t.

Definition 7 (Equations with Identical Unique Variables [19]). *An equation $u \approx v$ is said to have* identical unique variables *if u and v are both linear and the variables in u are the same as the variables in v.*

[3] In [12], $\mathcal{E}\backslash\mathcal{R}$ is coherent modulo $\mathcal{E}$ iff for all terms s,t,u, we have that $s \sim_{\mathcal{E}} t \to^{+}_{\mathcal{E}\backslash\mathcal{R}} u$ implies $s \to^{+}_{\mathcal{E}\backslash\mathcal{R}} v \sim_{\mathcal{E}} w \leftarrow^{*}_{\mathcal{E}\backslash\mathcal{R}} u$ for some v,w. Consider $\mathcal{R} = \{\mathsf{a}+\mathsf{b} \to \mathsf{a}+(\mathsf{b}+\mathsf{c}),\ \ x+y \to \mathsf{d}\}$ with $\mathcal{E}$ being the AC-axioms for $+$. The above system is coherent, since $s \sim_{\mathcal{E}} t \to^{+}_{\mathcal{E}\backslash\mathcal{R}} u$ implies $s \to^{+}_{\mathcal{R}} \mathsf{d} \leftarrow^{*}_{\mathcal{R}} u$. However, $\to_{\mathcal{E}\backslash\mathcal{R}}$ is terminating but $\to_{\mathcal{R}/\mathcal{E}}$ is not terminating.

Let $uni_{\mathcal{E}}(s,t)$ denote a complete set of $\mathcal{E}$-unifiers of two terms s and t. As usual, δ is an *$\mathcal{E}$-unifier* of s and t iff $s\delta \sim_{\mathcal{E}} t\delta$ and a set $uni_{\mathcal{E}}(s,t)$ of $\mathcal{E}$-unifiers is *complete* iff for every $\mathcal{E}$-unifier δ there exists a $\sigma \in uni_{\mathcal{E}}(s,t)$ and a substitution ρ such that $\delta \sim_{\mathcal{E}} \sigma\rho$, cf. [5]. ("$\sigma\rho$" is the composition of σ and ρ where σ is applied first and "$\delta \sim_{\mathcal{E}} \sigma\rho$" means that for all variables x we have $x\delta \sim_{\mathcal{E}} x\sigma\rho$.)

To construct $Ext_{\mathcal{E}}(\mathcal{R})$, we consider all overlaps between equations $u \approx v$ or $v \approx u$ from $\mathcal{E}$ and rules $l \to r$ from $\mathcal{R}$. More precisely, we check whether a non-variable subterm $v|_\pi$ of v $\mathcal{E}$-unifies with l (where we always assume that rules in $\mathcal{R}$ are variable disjoint from equations in $\mathcal{E}$). In this case one adds the rules $(v[l]_\pi)\sigma \to (v[r]_\pi)\sigma$ for all $\sigma \in uni_{\mathcal{E}}(v|_\pi, l)$.[4] In Ex. 5, the subterm $x_1 + x_2$ of the right-hand side of $x_1 + (x_2 + x_3) \approx (x_1 + x_2) + x_3$ unifies with the left-hand side of the only rule $\mathsf{a} + \mathsf{b} \to \mathsf{a} + (\mathsf{b} + \mathsf{c})$. Thus, in the extension of $\mathcal{R}$, we obtain the rule $(\mathsf{a} + \mathsf{b}) + y \to (\mathsf{a} + (\mathsf{b} + \mathsf{c})) + y$.

$Ext_{\mathcal{E}}(\mathcal{R})$ is built via a kind of fixpoint construction, i.e., we also have to consider overlaps between equations of $\mathcal{E}$ and the newly constructed rules of $Ext_{\mathcal{E}}(\mathcal{R})$. For example, the subterm $x_1 + x_2$ also unifies with the left-hand side of the new rule $(\mathsf{a} + \mathsf{b}) + y \to (\mathsf{a} + (\mathsf{b} + \mathsf{c})) + y$. Thus, one would now construct a new rule $((\mathsf{a} + \mathsf{b}) + y) + z \to ((\mathsf{a} + (\mathsf{b} + \mathsf{c})) + y) + z$.

Obviously, in this way one obtains an infinite number of rules by subsequently overlapping equations with the newly constructed rules. However, in order to use $Ext_{\mathcal{E}}(\mathcal{R})$ for automated termination proofs, our aim is to restrict ourselves to finitely many rules. It turns out that we do not have to include new rules $(v[l]_\pi)\sigma \to (v[r]_\pi)\sigma$ in $Ext_{\mathcal{E}}(\mathcal{R})$ if $u\sigma \to^{\pi'}_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} q \sim_{\mathcal{E}} (v[r]_\pi)\sigma$ already holds for some position π' of u and some term q (using just the old rules of $Ext_{\mathcal{E}}(\mathcal{R})$).

When constructing the rule $((\mathsf{a} + \mathsf{b}) + y) + z \to ((\mathsf{a} + (\mathsf{b} + \mathsf{c})) + y) + z$ above, the equation $u \approx v$ used was $x_1 + (x_2 + x_3) \approx (x_1 + x_2) + x_3$ and the unifier σ replaced x_1 by $(\mathsf{a} + \mathsf{b})$ and x_2 by y. Hence, here $u\sigma$ is the term $(\mathsf{a} + \mathsf{b}) + (y + x_3)$. But this term reduces with $\to^{1}_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})}$ to $(\mathsf{a} + (\mathsf{b} + \mathsf{c})) + (y + x_3)$ which is indeed $\sim_{\mathcal{E}}$-equivalent to $(v[r]_\pi)\sigma$, i.e., to $((\mathsf{a} + (\mathsf{b} + \mathsf{c})) + y) + x_3$. Thus, we do not have to include the rule $((\mathsf{a} + \mathsf{b}) + y) + z \to ((\mathsf{a} + (\mathsf{b} + \mathsf{c})) + y) + z$ in $Ext_{\mathcal{E}}(\mathcal{R})$.

The following definition shows how suitable extensions can be computed for arbitrary equations with identical unique variables. It will turn out that with these extensions one can indeed simulate $\to_{\mathcal{R}/\mathcal{E}}$ by $\to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})}$, i.e., $s \to_{\mathcal{R}/\mathcal{E}} t$ implies $s \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} t'$ for some $t' \sim_{\mathcal{E}} t$. This constitutes a crucial contribution of the paper, since it is the main requirement needed in order to extend dependency pairs to rewriting modulo equations.

Definition 8 (Extending $\mathcal{R}$ for Arbitrary Equations). *Let $\mathcal{R}$ be a TRS and let $\mathcal{E}$ be a set of equations. Let $\mathcal{R}'$ be a set containing only rules of the form*

[4] Obviously, $uni_{\mathcal{E}}(v|_\pi, l)$ always exists, but it can be infinite in general. So when automating our approach for equational termination proofs, we have to restrict ourselves to equations $\mathcal{E}$ where $uni_{\mathcal{E}}(v|_\pi, l)$ can be chosen to be finite for all subterms $v|_\pi$ of equations and left-hand sides of rules l. This includes all sets $\mathcal{E}$ of finitary unification type, but our restriction is weaker, since we only need finiteness for certain terms $v|_\pi$ and l.

$C[l\sigma] \to C[r\sigma]$ *(where C is a context, σ is a substitution, and $l \to r \in \mathcal{R}$). $\mathcal{R}'$ is an* extension of $\mathcal{R}$ for the equations $\mathcal{E}$ *iff*

(a) $\mathcal{R} \subseteq \mathcal{R}'$ *and*
(b) for all $l \to r \in \mathcal{R}'$, $u \approx v \in \mathcal{E}$ and $v \approx u \in \mathcal{E}$, all positions π of v and $\sigma \in uni_{\mathcal{E}}(v|_\pi, l)$, there is a position π' in u and a $q \sim_{\mathcal{E}} (v[r]_\pi)\sigma$ with $u\sigma \to^{\pi'}_{\mathcal{E}\backslash\mathcal{R}'} q$.

In the following, let $Ext_{\mathcal{E}}(\mathcal{R})$ always denote an arbitrary extension of $\mathcal{R}$ for $\mathcal{E}$.

In order to satisfy Condition (b) of Def. 8, it is always sufficient to add the rule $(v[l]_\pi)\sigma \to (v[r]_\pi)\sigma$ to $\mathcal{R}'$. The reason is that then we have $u\sigma \to^{\epsilon}_{\mathcal{E}\backslash\mathcal{R}'} (v[r]_\pi)\sigma$. But if $u\sigma \to^{\pi'}_{\mathcal{E}\backslash\mathcal{R}'} q \sim_{\mathcal{E}} (v[r]_\pi)\sigma$ already holds with the other rules of $\mathcal{R}'$, then the rule $(v[l]_\pi)\sigma \to (v[r]_\pi)\sigma$ does not have to be added to $\mathcal{R}'$.

Condition (b) of Def. 8 also makes sure that as long as the equations have identical unique variables, we do not have to consider overlaps at variable positions.[5] The reason is that if $v|_\pi$ is a variable $x \in \mathcal{V}$, then we have $u\sigma = u[x\sigma]_{\pi'} \sim_{\mathcal{E}} u[l\sigma]_{\pi'} \to_{\mathcal{R}} u[r\sigma]_{\pi'} \sim_{\mathcal{E}} v[r\sigma]_\pi = (v[r]_\pi)\sigma$, where π' is the position of x in u. Hence, such rules $(v[l]_\pi)\sigma \to (v[r]_\pi)\sigma$ do not have to be included in $\mathcal{R}'$.

Overlaps at root positions do not have to be considered either. To see this, assume that π is the top position ϵ of v, i.e., that $v\sigma \sim_{\mathcal{E}} l\sigma$. In this case we have $u\sigma \sim_{\mathcal{E}} v\sigma \sim_{\mathcal{E}} l\sigma \to_{\mathcal{R}} r\sigma$ and thus, $u\sigma \to^{\epsilon}_{\mathcal{E}\backslash\mathcal{R}} r\sigma = (v[r]_\pi)\sigma$. So again, such rules $(v[l]_\pi) \to (v[r]_\pi)\sigma$ do not have to be included in $\mathcal{R}'$.

The following procedure is used to compute extensions. Here, we assume both $\mathcal{R}$ and $\mathcal{E}$ to be finite, where the equations $\mathcal{E}$ must have identical unique variables.

1. $\mathcal{R}' := \mathcal{R}$
2. For all $l \to r \in \mathcal{R}'$,
   all $u \approx v$ or $v \approx u$ from $\mathcal{E}$,
   and all positions π of v where $\pi \neq \epsilon$ and $v|_\pi \notin \mathcal{V}$ do:
   2.1. Let $\Sigma := uni_{\mathcal{E}}(v|_\pi, l)$.
   2.2. For all $\sigma \in \Sigma$ do:
      2.2.1. Let $T := \{q \mid u\sigma \to^{\pi'}_{\mathcal{E}\backslash\mathcal{R}'} q$ for a position π' of $u\}$.
      2.2.2. If there exists a $q \in T$ with $(v[r]_\pi)\sigma \sim_{\mathcal{E}} q$, then $\Sigma := \Sigma \setminus \{\sigma\}$.
   2.3. $\mathcal{R}' := \mathcal{R}' \cup \{(v[l]_\pi)\sigma \to (v[r]_\pi)\sigma \mid \sigma \in \Sigma\}$.

This algorithm has the following properties:

(a) If in Step 2.1, $uni_{\mathcal{E}}(v|_\pi, l)$ is finite and computable, then every step in the algorithm is computable.
(b) If the algorithm terminates, then the final value of $\mathcal{R}'$ is an extension of $\mathcal{R}$ for the equations $\mathcal{E}$.

[5] Note that considering overlaps at variable positions as well would still not allow us to treat equations with non-linear terms. As an example regard $\mathcal{E} = \{\mathsf{f}(x) \approx \mathsf{g}(x, x)\}$ and $\mathcal{R} = \{\mathsf{g}(\mathsf{a}, \mathsf{b}) \to \mathsf{f}(\mathsf{a}), \mathsf{a} \to \mathsf{b}\}$. Here, $\to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})}$ is well founded although $\mathcal{R}$ is not terminating modulo $\mathcal{E}$.

With the TRS of Ex. 5, $Ext_{\mathcal{E}}(\mathcal{R}) = \{\mathsf{a} + \mathsf{b} \to \mathsf{a} + (\mathsf{b} + \mathsf{c}), (\mathsf{a} + \mathsf{b}) + y \to (\mathsf{a} + (\mathsf{b} + \mathsf{c})) + y\}$. In general, if $\mathcal{E}$ only consists of AC-axioms for some function symbols $\mathcal{G}$, then Def. 8 "coincides" with the well-known extension for AC-axioms, i.e., $\mathcal{R}' = \mathcal{R} \cup \{f(l, y) \to f(r, y) \mid l \to r \in \mathcal{R}, \mathit{root}(l) = f \in \mathcal{G}\}$ satisfies the conditions (a) and (b) of Def. 8. So in case of AC-equations, our approach indeed corresponds to the approaches of [15, 17]. However, Def. 8 can also be used for other forms of equations.

Example 9. As an example, consider the following system from [18].

$$\begin{aligned} \mathcal{R} = \{ \quad & x - \mathsf{0} \to x, \\ & \mathsf{s}(x) - \mathsf{s}(y) \to x - y, \\ & \mathsf{0} \div \mathsf{s}(y) \to \mathsf{0}, \\ & \mathsf{s}(x) \div \mathsf{s}(y) \to \mathsf{s}((x - y) \div \mathsf{s}(y))\} \end{aligned} \qquad \mathcal{E} = \{(u \div v) \div w \;\approx\; (u \div w) \div v\}$$

By overlapping the subterm $u \div w$ in the right-hand side of the equation with the left-hand sides of the last two rules we obtain

$$\begin{aligned} Ext_{\mathcal{E}}(\mathcal{R}) = \mathcal{R} \cup \quad \{ \quad & (\mathsf{0} \div \mathsf{s}(y)) \div z \to \mathsf{0} \div z, \\ & (\mathsf{s}(x) \div \mathsf{s}(y)) \div z \to \mathsf{s}((x - y) \div \mathsf{s}(y)) \div z\ \}. \end{aligned}$$

Note that these are indeed all the rules of $Ext_{\mathcal{E}}(\mathcal{R})$. Overlapping the subterm $u \div v$ of the equation's left-hand side with the third rule would result in $(\mathsf{0} \div \mathsf{s}(y)) \div z' \to \mathsf{0} \div z'$. But this new rule does not have to be included in $Ext_{\mathcal{E}}(\mathcal{R})$, since the corresponding other term of the equation, $(\mathsf{0} \div z') \div \mathsf{s}(y)$, would $\to^{\epsilon}_{\mathcal{E} \backslash Ext_{\mathcal{E}}(\mathcal{R})}$-reduce with the rule $(\mathsf{0} \div \mathsf{s}(y)) \div z \to \mathsf{0} \div z$ to $\mathsf{0} \div z'$. Overlapping $u \div v$ with the left-hand side of the fourth rule is also superfluous.

Similarly, overlaps with the new rules $(\mathsf{0} \div \mathsf{s}(y)) \div z \to \mathsf{0} \div z$ or $(\mathsf{s}(x) \div \mathsf{s}(y)) \div z \to \mathsf{s}((x - y) \div \mathsf{s}(y)) \div z$ also do not give rise to additional rules in $Ext_{\mathcal{E}}(\mathcal{R})$. To see this, overlap the subterm $u \div w$ in the right-hand side of the equation with the left-hand side of $(\mathsf{0} \div \mathsf{s}(y)) \div z \to \mathsf{0} \div z$. This gives the rule $((\mathsf{0} \div \mathsf{s}(y)) \div z) \div z' \to (\mathsf{0} \div z) \div z'$. However, the corresponding other term of the equation is $((\mathsf{0} \div \mathsf{s}(y)) \div z') \div z$. This reduces at position 1 (or position 11) to $(\mathsf{0} \div z') \div z$, which is $\mathcal{E}$-equivalent to $(\mathsf{0} \div z) \div z'$. Overlaps with the other new rule $(\mathsf{s}(x) \div \mathsf{s}(y)) \div z \to \mathsf{s}((x - y) \div \mathsf{s}(y)) \div z$ are not needed either.

Nevertheless, the above algorithm for computing extensions does not always terminate. For example, for $\mathcal{R} = \{\mathsf{a}(x) \to \mathsf{c}(x)\}$, $\mathcal{E} = \{\mathsf{a}(\mathsf{b}(\mathsf{a}(x))) \approx \mathsf{b}(\mathsf{a}(\mathsf{b}(x)))\}$, it can be shown that all extensions $Ext_{\mathcal{E}}(\mathcal{R})$ are infinite.

We prove below that $Ext_{\mathcal{E}}(\mathcal{R})$ (according to Def. 8) has the desired property needed to reduce rewriting modulo equations to $\mathcal{E}$-extended rewriting. The following important lemma states that whenever s rewrites to t with $\to_{\mathcal{R}/\mathcal{E}}$ modulo $\mathcal{E}$, then s also rewrites with $\to_{\mathcal{E} \backslash Ext_{\mathcal{E}}(\mathcal{R})}$ to a term which is $\mathcal{E}$-equivalent to t.[6]

[6] Our extension $Ext_{\mathcal{E}}$ has some similarities to the construction of contexts in [23]. However, in contrast to [23] we also consider the rules of $\mathcal{R}'$ in Condition (b) of Def. 8 in order to reduce the number of rules in $Ext_{\mathcal{E}}$. Moreover, in [23] equations may also be non-linear (and thus, Lemma 10 does not hold there).

Lemma 10 (Connection between $\to_{\mathcal{R}/\mathcal{E}}$ and $\to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})}$). *Let $\mathcal{R}$ be a TRS and let $\mathcal{E}$ be a set of equations with identical unique variables. If $s \to_{\mathcal{R}/\mathcal{E}} t$, then there exists a term $t' \sim_{\mathcal{E}} t$ such that $s \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} t'$.*

Proof. Let $s \to_{\mathcal{R}/\mathcal{E}} t$, i.e., there exist terms $s_0, \ldots, s_n, p$ with $n \geq 0$ such that $s = s_n \vdash\!\dashv_{\mathcal{E}} s_{n-1} \vdash\!\dashv_{\mathcal{E}} \ldots \vdash\!\dashv_{\mathcal{E}} s_0 \to_{\mathcal{R}} p \sim_{\mathcal{E}} t$. For the lemma, it suffices to show that there is a $t' \sim_{\mathcal{E}} p$ such that $s \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} t'$, since $t' \sim_{\mathcal{E}} p$ implies $t' \sim_{\mathcal{E}} t$.

We perform induction on n. If $n = 0$, we have $s = s_n = s_0 \to_{\mathcal{R}} p$. This implies $s \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} p$ since $\mathcal{R} \subseteq Ext_{\mathcal{E}}(\mathcal{R})$. So with $t' = p$ the claim is proved.

If $n > 0$, the induction hypothesis implies $s = s_n \vdash\!\dashv_{\mathcal{E}} s_{n-1} \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} t'$ such that $t' \sim_{\mathcal{E}} p$. So there exists an equation $u \approx v$ or $v \approx u$ from $\mathcal{E}$ and a rule $l \to r$ from $Ext_{\mathcal{E}}(\mathcal{R})$ such that $s|_\tau = u\delta$, $s_{n-1} = s[v\delta]_\tau$, $s_{n-1}|_\xi \sim_{\mathcal{E}} l\delta$, and $t' = s_{n-1}[r\delta]_\xi$ for positions τ and ξ and a substitution δ. We can use the same substitution δ for instantiating the equation $u \approx v$ (or $v \approx u$) and the rule $l \to r$, since equations and rules are assumed variable disjoint. We now perform a case analysis depending on the relationship of the positions τ and ξ.

Case 1: $\tau = \xi\pi$ for some π. In this case, we have $s|_\xi = s|_\xi[u\delta]_\pi \vdash\!\dashv_{\mathcal{E}} s|_\xi[v\delta]_\pi = s_{n-1}|_\xi \sim_{\mathcal{E}} l\delta$. This implies $s \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} s[r\delta]_\xi = s_{n-1}[r\delta]_\xi = t'$, as desired.

Case 2: $\tau \bot \xi$. Now we have $s|_\xi = s_{n-1}|_\xi \sim_{\mathcal{E}} l\delta$ and thus, $s \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} s[r\delta]_\xi = s[r\delta]_\xi[u\delta]_\tau \vdash\!\dashv_{\mathcal{E}} s[r\delta]_\xi[v\delta]_\tau = s[v\delta]_\tau[r\delta]_\xi = s_{n-1}[r\delta]_\xi = t'$.

Case 3: $\xi = \tau\pi$ for some π. Thus, $(v\delta)|_\pi \sim_{\mathcal{E}} l\delta$. We distinguish two sub-cases.

Case 3.1: $u\delta \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} q \sim_{\mathcal{E}} (v[r]_\pi)\delta$ for some term q. This implies $s = s[u\delta]_\tau \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} s[q]_\tau \sim_{\mathcal{E}} s[v[r]_\pi\delta]_\tau = (s[v\delta]_\tau)[r\delta]_\xi = s_{n-1}[r\delta]_\xi = t'$.

Case 3.2: Otherwise. First assume that $\pi = \pi_1\pi_2$ where $v|_{\pi_1}$ is a variable x. Hence, $(v\delta)|_\pi = \delta(x)|_{\pi_2}$. Let $\delta'(y) = \delta(y)$ for $y \neq x$ and let $\delta'(x) = \delta(x)[r\delta]_{\pi_2}$. Since $u \approx v$ (or $v \approx u$) is an equation with identical unique variables, x also occurs in u at some position π'. This implies $u\delta|_{\pi'\pi_2} = \delta(x)|_{\pi_2} \sim_{\mathcal{E}} l\delta \to_{Ext_{\mathcal{E}}(\mathcal{R})} r\delta$. Hence, we obtain $u\delta \to^{\pi'\pi_2}_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} u\delta[r\delta]_{\pi'\pi_2} = u\delta' \sim_{\mathcal{E}} v\delta' = (v[r]_\pi)\delta$ in contradiction to the condition of Case 3.2.

Hence, π is a position of v and $v|_\pi$ is not a variable. Thus, $(v\delta)|_\pi = v|_\pi\delta \sim_{\mathcal{E}} l\delta$. Since rules and equations are assumed variable disjoint, the subterm $v|_\pi$ $\mathcal{E}$-unifies with l. Thus, there exists a $\sigma \in uni_{\mathcal{E}}(v|_\pi, l)$ such that $\delta \sim_{\mathcal{E}} \sigma\rho$.

Due to the Condition (b) of Def. 8, there is a term q' such that $u\sigma \to^{\pi'}_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} q' \sim_{\mathcal{E}} (v[r]_\pi)\sigma$. Since π' is a position in u, we have $u|_{\pi'}\sigma \sim_{\mathcal{E}} \circ \to_{Ext_{\mathcal{E}}(\mathcal{R})} q''$, where $q' = u\sigma[q'']_{\pi'}$. This also implies $u|_{\pi'}\delta \sim_{\mathcal{E}} u|_{\pi'}\sigma\rho \sim_{\mathcal{E}} \circ \to_{Ext_{\mathcal{E}}(\mathcal{R})} q''\rho$, and thus $u\delta \to^{\pi'}_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} u\delta[q''\rho]_{\pi'} \sim_{\mathcal{E}} u\sigma[q'']_{\pi'}\rho = q'\rho \sim_{\mathcal{E}} (v[r]_\pi)\sigma\rho \sim_{\mathcal{E}} (v[r]_\pi)\delta$. This is a contradiction to the condition of Case 3.2. $\square$

The following theorem shows that $Ext_{\mathcal{E}}$ indeed has the desired property.

Theorem 11 (Termination of $\mathcal{R}/\mathcal{E}$ by $\mathcal{E}$-Extended Rewriting). *Let $\mathcal{R}$ be a TRS, let $\mathcal{E}$ be a set of equations with identical unique variables, and let t be a term. Then t does not start an infinite $\to_{\mathcal{R}/\mathcal{E}}$-reduction iff t does not start* *an infinite* $\to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})}$*-reduction. So in particular,* $\mathcal{R}$ *is terminating modulo* $\mathcal{E}$ *(i.e.,* $\to_{\mathcal{R}/\mathcal{E}}$ *is well founded) iff* $\to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})}$ *is well founded.*

Proof. The "only if" direction is straightforward because $\to_{Ext_{\mathcal{E}}(\mathcal{R})} = \to_{\mathcal{R}}$ and therefore, $\to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} \subseteq \to_{Ext_{\mathcal{E}}(\mathcal{R})/\mathcal{E}} = \to_{\mathcal{R}/\mathcal{E}}$.
For the "if" direction, assume that t starts an infinite $\to_{\mathcal{R}/\mathcal{E}}$-reduction

$$t = t_0 \to_{\mathcal{R}/\mathcal{E}} t_1 \to_{\mathcal{R}/\mathcal{E}} t_2 \to_{\mathcal{R}/\mathcal{E}} \dots$$

For every $i \in \mathbb{N}$, let f_{i+1} be a function from terms to terms such that for every $t'_i \sim_{\mathcal{E}} t_i$, $f_{i+1}(t'_i)$ is a term $\mathcal{E}$-equivalent to t_{i+1} such that $t'_i \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} f_{i+1}(t'_i)$. These functions f_{i+1} must exist due to Lemma 10, since $t'_i \sim_{\mathcal{E}} t_i$ and $t_i \to_{\mathcal{R}/\mathcal{E}} t_{i+1}$ implies $t'_i \to_{\mathcal{R}/\mathcal{E}} t_{i+1}$. Hence, t starts an infinite $\to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})}$-reduction:

$$t \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} f_1(t) \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} f_2(f_1(t)) \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} f_3(f_2(f_1(t))) \to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})} \dots \quad \square$$

5 Dependency Pairs for Rewriting Modulo Equations

In this section we finally extend the dependency pair approach to rewriting modulo equations: To show that $\mathcal{R}$ modulo $\mathcal{E}$ terminates, one first constructs the extension $Ext_{\mathcal{E}}(\mathcal{R})$ of $\mathcal{R}$. Subsequently, dependency pairs can be used to prove well-foundedness of $\to_{\mathcal{E}\backslash Ext_{\mathcal{E}}(\mathcal{R})}$ (which is equivalent to termination of $\mathcal{R}$ modulo $\mathcal{E}$). The idea for the extension of the dependency pair approach is simply to modify Thm. 3 as follows.

1. The equations should be satisfied by the equivalence $\sim$ corresponding to the quasi-ordering $\succsim$, i.e., we demand $u \sim v$ for all equations $u \approx v$ in $\mathcal{E}$.
2. A similar requirement is needed for equations $u \approx v$ when the root symbols of u and v are replaced by the corresponding tuple symbols. We denote tuples of terms $s_1, \dots, s_n$ by $\boldsymbol{s}$ and for any term $t = f(\boldsymbol{s})$ with a defined root symbol f, let $t^\sharp$ be the term $F(\boldsymbol{s})$. Hence, we also have to demand $u^\sharp \sim v^\sharp$.
3. The notion of "defined symbols" must be changed accordingly. As before, all root symbols of left-hand sides of rules are regarded as being defined, but if there is an equation $f(\boldsymbol{u}) = g(\boldsymbol{v})$ in $\mathcal{E}$ and f is defined, then g must be considered defined as well, as otherwise we would not be able to trace the redex in a reduction by only regarding subterms with defined root symbols.

Definition 12 (Defined Symbols for Rewriting Modulo Equations). *Let* $\mathcal{R}$ *be a TRS and let* $\mathcal{E}$ *be a set of equations. Then the set of* defined *symbols* $\mathcal{D}$ *of* $\mathcal{R}/\mathcal{E}$ *is the smallest set such that* $\mathcal{D} = \{root(l) \mid l \to r \in \mathcal{R}\} \cup \{root(v) \mid u \approx v \in \mathcal{E} \text{ or } v \approx u \in \mathcal{E},\ root(u) \in \mathcal{D}\}$.

The constraints of the dependency pair approach as sketched above are not yet sufficient for termination of $\to_{\mathcal{E}\backslash\mathcal{R}}$ as the following example illustrates.

Example 13. Consider $\mathcal{R} = \{\mathsf{f}(x) \to x\}$ and $\mathcal{E} = \{\mathsf{f}(\mathsf{a}) \approx \mathsf{a}\}$. There is no dependency pair in this example and thus, the only constraints would be $\mathsf{f}(x) \succsim x$, $\mathsf{f}(\mathsf{a}) \sim \mathsf{a}$, and $\mathsf{F}(\mathsf{a}) \sim \mathsf{A}$. Obviously, these constraints are satisfiable (by using an equivalence relation $\sim$ where all terms are equal). However, $\to_{\mathcal{E}\backslash\mathcal{R}}$ is not terminating since we have $\mathsf{a} \vdash\!\dashv_{\mathcal{E}} \mathsf{f}(\mathsf{a}) \to_{\mathcal{R}} \mathsf{a} \vdash\!\dashv_{\mathcal{E}} \mathsf{f}(\mathsf{a}) \to_{\mathcal{R}} \mathsf{a} \vdash\!\dashv_{\mathcal{E}} \dots$

The soundness of the dependency pair approach for ordinary rewriting (Thm. 3) relies on the fact that an infinite reduction from a minimal non-terminating term can be achieved by applying only normalized instantiations of $\mathcal{R}$-rules. But for $\mathcal{E}$-extended rewriting (or full rewriting modulo equations), this is not true any more. For instance, the minimal non-terminating subterm a in Ex. 13 is first modified by applying an $\mathcal{E}$-equation (resulting in $\mathsf{f}(\mathsf{a})$) and then an $\mathcal{R}$-rule is applied whose variable is instantiated with the non-terminating term a. Hence, the problem is that the new minimal non-terminating subterm a which results from application of the $\mathcal{R}$-rule does not correspond to the right-hand side of a dependency pair, because this minimal non-terminating subterm is completely inside the instantiation of a *variable* of the $\mathcal{R}$-rule. With ordinary rewriting, this situation can never occur.

In Ex. 13, the problem can be avoided by adding a suitable *instance* of the rule $\mathsf{f}(x) \to x$ (viz. $\mathsf{f}(\mathsf{a}) \to \mathsf{a}$) to $\mathcal{R}$, since this instance is used in the infinite reduction. Now there would be a dependency pair $\langle \mathsf{F}(\mathsf{a}), \mathsf{A} \rangle$ and with the additional constraint $\mathsf{F}(\mathsf{a}) > \mathsf{A}$ the resulting inequalities are no longer satisfiable.

The following definition shows how to add the right instantiations of the rules in $\mathcal{R}$ in order to allow a sound application of dependency pairs. As usual, a substitution ν is called a *variable renaming* iff the range of ν only contains variables and if $\nu(x) \neq \nu(y)$ for $x \neq y$.

Definition 14 (Adding Instantiations). *Given a TRS $\mathcal{R}$, a set $\mathcal{E}$ of equations, let $\mathcal{R}'$ be a set containing only rules of the form $l\sigma \to r\sigma$ (where σ is a substitution and $l \to r \in \mathcal{R}$). $\mathcal{R}'$ is an* instantiation of $\mathcal{R}$ for the equations $\mathcal{E}$ *iff*

(a) $\mathcal{R} \subseteq \mathcal{R}'$,
(b) for all $l \to r \in \mathcal{R}$, all $u \approx v \in \mathcal{E}$ and $v \approx u \in \mathcal{E}$, and all $\sigma \in uni_{\mathcal{E}}(v, l)$, there exists a rule $l' \to r' \in \mathcal{R}'$ and a variable renaming ν such that $l\sigma \sim_{\mathcal{E}} l'\nu$ and $r\sigma \sim_{\mathcal{E}} r'\nu$.

In the following, let $Ins_{\mathcal{E}}(\mathcal{R})$ always denote an instantiation of $\mathcal{R}$ for $\mathcal{E}$.

Unlike extensions $Ext_{\mathcal{E}}(\mathcal{R})$, instantiations $Ins_{\mathcal{E}}(\mathcal{R})$ are never infinite if $\mathcal{R}$ and $\mathcal{E}$ are finite and if $uni_{\mathcal{E}}(v, l)$ is always finite (i.e., they are not defined via a fixpoint construction). In fact, one might even demand that for all $l \to r \in \mathcal{R}$, all equations, and all σ from the corresponding complete set of $\mathcal{E}$-unifiers, $Ins_{\mathcal{E}}(\mathcal{R})$ should contain $l\sigma \to r\sigma$. The condition that it is enough if some $\mathcal{E}$-equivalent variable-renamed rule is already contained in $Ins_{\mathcal{E}}(\mathcal{R})$ is only added for efficiency considerations in order to reduce the number of rules in $Ins_{\mathcal{E}}(\mathcal{R})$. Even without this condition, $Ins_{\mathcal{E}}(\mathcal{R})$ would still be finite and all the following theorems would hold as well.

However, the above instantiation technique only serves its purpose if there are no collapsing equations (i.e., no equations $u \approx v$ or $v \approx u$ with $v \in \mathcal{V}$).

Example 15. Consider $\mathcal{R} = \{\mathsf{f}(x) \to x\}$ and $\mathcal{E} = \{\mathsf{f}(x) \approx x\}$. Note that $Ins_{\mathcal{E}}(\mathcal{R}) = \mathcal{R}$. Although $\to_{\mathcal{E}\backslash\mathcal{R}}$ is clearly not terminating, the dependency pair approach would falsely prove termination of $\to_{\mathcal{E}\backslash\mathcal{R}}$, since there is no dependency pair.

Now we can present the main result of the paper.

Theorem 16 (Termination of Equational Rewriting using Dependency Pairs). *Let $\mathcal{R}$ be a TRS and let $\mathcal{E}$ be a set of non-collapsing equations with identical unique variables. $\mathcal{R}$ is terminating modulo $\mathcal{E}$ (i.e., $\rightarrow_{\mathcal{R}/\mathcal{E}}$ is well founded) if there exists a weakly monotonic quasi-ordering $\succsim$ and a well-founded ordering $>$ compatible with $\succsim$ where both $\succsim$ and $>$ are closed under substitution, such that*

(1) $s > t$ for all dependency pairs $\langle s, t\rangle$ of $Ins_{\mathcal{E}}(Ext_{\mathcal{E}}(\mathcal{R}))$,
(2) $l \succsim r$ for all rules $l \rightarrow r$ of $\mathcal{R}$,
(3) $u \sim v$ for all equations $u \approx v$ of $\mathcal{E}$, and
(4) $u^\sharp \sim v^\sharp$ for all equations $u \approx v$ of $\mathcal{E}$ where $root(u)$ and $root(v)$ are defined.

Proof. Suppose that there is a term t with an infinite $\rightarrow_{\mathcal{R}/\mathcal{E}}$-reduction. Thm. 11 implies that t also has an infinite $\rightarrow_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reduction. By a minimality argument, $t = C[t']$, where t' is an *minimal non-terminating* term (i.e., t' is non-terminating, but all its subterms only have finite $\rightarrow_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reductions). We will show that there exists a term t_1 with $t \rightarrow^+_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})} t_1$, t_1 contains a minimal non-terminating subterm t'_1, and $t'^\sharp \succsim \circ > {t'_1}^\sharp$. By repeated application of this construction we obtain an infinite sequence $t \rightarrow^+_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})} t_1 \rightarrow^+_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})} t_2 \rightarrow^+_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})} \ldots$ such that $t'^\sharp \succsim \circ > {t'_1}^\sharp \succsim \circ > {t'_2}^\sharp \succsim \circ > \ldots$. This, however, is a contradiction to the well-foundedness of $>$.

Let t' have the form $f(\boldsymbol{u})$. In the infinite $\rightarrow_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reduction of $f(\boldsymbol{u})$, first some $\rightarrow_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})}$-steps may be applied to $\boldsymbol{u}$ which yields new terms $\boldsymbol{v}$. Note that due to the definition of $\mathcal{E}$-extended rewriting, in these reductions, no $\mathcal{E}$-steps can be applied outside of $\boldsymbol{u}$. Due to the termination of $\boldsymbol{u}$, after a finite number of those steps, an $\rightarrow_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})}$-step must be applied on the root position of $f(\boldsymbol{v})$.

Thus, there exists a rule $l \rightarrow r \in Ext_{\mathcal{E}}(\mathcal{R})$ such that $f(\boldsymbol{v}) \sim_{\mathcal{E}} l\alpha$ and hence, the reduction yields $r\alpha$. Now the infinite $\rightarrow_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reduction continues with $r\alpha$, i.e., the term $r\alpha$ starts an infinite $\rightarrow_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reduction, too. So up to now the reduction has the following form (where $\rightarrow_{Ext_{\mathcal{E}}(\mathcal{R})}$ equals $\rightarrow_{\mathcal{R}}$):

$$t = C[f(\boldsymbol{u})] \rightarrow^*_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})} C[f(\boldsymbol{v})] \sim_{\mathcal{E}} C[l\alpha] \rightarrow_{Ext_{\mathcal{E}}(\mathcal{R})} C[r\alpha].$$

We perform a case analysis depending on the positions of $\mathcal{E}$-steps in $f(\boldsymbol{v}) \sim_{\mathcal{E}} l\alpha$.

First consider the case where all $\mathcal{E}$-steps in $f(\boldsymbol{v}) \sim_{\mathcal{E}} l\alpha$ take place below the root. Then we have $l = f(\boldsymbol{w})$ and $\boldsymbol{v} \sim_{\mathcal{E}} \boldsymbol{w}\alpha$. Let $t_1 := C[r\alpha]$. Note that $\boldsymbol{v}$ do not start infinite $\rightarrow_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reductions and by Thm. 11, they do not start infinite $\rightarrow_{\mathcal{R}/\mathcal{E}}$-reductions either. But then $\boldsymbol{w}\alpha$ also cannot start infinite $\rightarrow_{\mathcal{R}/\mathcal{E}}$-reductions and therefore they also do not start infinite $\rightarrow_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reductions. This implies that for all variables x occurring in $f(\boldsymbol{w})$ the terms $\alpha(x)$ are terminating. Thus, since $r\alpha$ starts an infinite reduction, there occurs a non-variable subterm s in r, such that $t'_1 := s\alpha$ is a minimal non-terminating term. Since $\langle l^\sharp, s^\sharp\rangle$ is a dependency pair, we obtain $t'^\sharp = F(\boldsymbol{u}) \succsim F(\boldsymbol{v}) \sim l^\sharp\alpha > s^\sharp\alpha = {t'_1}^\sharp$. Here, $F(\boldsymbol{u}) \succsim F(\boldsymbol{v})$ holds since $\boldsymbol{u} \rightarrow^*_{\mathcal{E}\setminus Ext_{\mathcal{E}}(\mathcal{R})} \boldsymbol{v}$ and since $l \succsim r$ for every rule $l \rightarrow r \in Ext_{\mathcal{E}}(\mathcal{R})$.

Now we consider the case where there are $\mathcal{E}$-steps in $f(\boldsymbol{v}) \sim_{\mathcal{E}} l\alpha$ at the root position. Thus we have $f(\boldsymbol{v}) \sim_{\mathcal{E}} f(\boldsymbol{q}) \vdash\!\dashv_{\mathcal{E}} p \sim_{\mathcal{E}} l\alpha$, where $f(\boldsymbol{q}) \vdash\!\dashv_{\mathcal{E}} p$ is the first

$\mathcal{E}$-step at the root position. In other words, there is an equation $u \approx v$ or $v \approx u$ in $\mathcal{E}$ such that $f(\boldsymbol{q})$ is an instantiation of v.

Note that since $\boldsymbol{v} \sim_{\mathcal{E}} \boldsymbol{q}$, the terms $\boldsymbol{q}$ only have finite $\to_{\mathcal{E} \setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reductions (the argumentation is similar as in the first case). Let δ be the substitution which operates like α on the variables of l and which yields $v\delta = f(\boldsymbol{q})$. Thus, δ is an $\mathcal{E}$-unifier of l and v. Since l is $\mathcal{E}$-unifiable with v, there also exists a corresponding complete $\mathcal{E}$-unifier σ from $uni_{\mathcal{E}}(l, v)$. Thus, there is also a substitution ρ such that $\delta \sim_{\mathcal{E}} \sigma\rho$. As l is a left-hand side of a rule from $Ext_{\mathcal{E}}(\mathcal{R})$, there is a rule $l' \to r'$ in $Ins_{\mathcal{E}}(Ext_{\mathcal{E}}(\mathcal{R}))$ and a variable renaming ν such that $l\sigma \sim_{\mathcal{E}} l'\nu$ and $r\sigma \sim_{\mathcal{E}} r'\nu$.

Hence, $v\sigma\rho \sim_{\mathcal{E}} v\delta = f(\boldsymbol{q})$, $l'\nu\rho \sim_{\mathcal{E}} l\sigma\rho \sim_{\mathcal{E}} l\delta = l\alpha$, and $r'\nu\rho \sim_{\mathcal{E}} r\sigma\rho \sim_{\mathcal{E}} r\delta = r\alpha$. So instead we now consider the following reduction (where $\to_{Ins_{\mathcal{E}}(Ext_{\mathcal{E}}(\mathcal{R}))}$ equals $\to_{\mathcal{R}}$):

$$t = C[f(\boldsymbol{u})] \to^*_{\mathcal{E} \setminus Ext_{\mathcal{E}}(\mathcal{R})} C[f(\boldsymbol{v})] \sim_{\mathcal{E}} C[l'\nu\rho] \to_{Ins_{\mathcal{E}}(Ext_{\mathcal{E}}(\mathcal{R}))} C[r'\nu\rho] = t_1.$$

Since all proper subterms of $v\delta$ only have finite $\to_{\mathcal{R}/\mathcal{E}}$-reductions, for all variables x of $l'\nu$, the term $x\rho$ only has finite $\to_{\mathcal{R}/\mathcal{E}}$-reductions and hence, also only finite $\to_{\mathcal{E} \setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reductions. To see this, note that since all equations have identical unique variables, $v\sigma \sim_{\mathcal{E}} l\sigma \sim_{\mathcal{E}} l'\nu$ implies that all variables of $l'\nu$ also occur in $v\sigma$. Thus, if x is a variable from $l'\nu$, then there exists a variable y in v such that x occurs in $y\sigma$. Since $\mathcal{E}$ does not contain collapsing equations, y is a proper subterm of v and thus, $y\delta$ is a proper subterm of $v\delta$. As all proper subterms of $v\delta$ only have finite $\to_{\mathcal{R}/\mathcal{E}}$-reductions, this implies that $y\delta$ only has finite $\to_{\mathcal{R}/\mathcal{E}}$-reductions, too. But then, since $y\delta \sim_{\mathcal{E}} y\sigma\rho$, the term $y\sigma\rho$ only has finite $\to_{\mathcal{R}/\mathcal{E}}$-reductions, too. Then this also holds for all subterms of $y\sigma\rho$, i.e., all $\to_{\mathcal{R}/\mathcal{E}}$-reductions of $x\rho$ are also finite.

So for all variables x of l', $x\nu\rho$ only has finite $\to_{\mathcal{E} \setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reductions. (Note that this only holds because ν is just a variable renaming.) Since $r\alpha$ starts an infinite $\to_{\mathcal{E} \setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reduction, $r'\nu\rho \sim_{\mathcal{E}} r\alpha$ must start an infinite $\to_{\mathcal{R}/\mathcal{E}}$-reduction (and hence, an infinite $\to_{\mathcal{E} \setminus Ext_{\mathcal{E}}(\mathcal{R})}$-reduction) as well. As for all variables x of r', $x\nu\rho$ is $\to_{\mathcal{E} \setminus Ext_{\mathcal{E}}(\mathcal{R})}$-terminating, there must be a non-variable subterm s of r', such that $t'_1 := s\nu\rho$ is a minimal non-terminating term. As $\langle l'^\sharp, s^\sharp \rangle$ is a dependency pair, we obtain $t'^\sharp = F(\boldsymbol{u}) \succsim F(\boldsymbol{v}) \sim_{\mathcal{E}} l'^\sharp\nu\rho > s^\sharp\nu\rho = {t'_1}^\sharp$. Here, $F(\boldsymbol{v}) \sim_{\mathcal{E}} l'^\sharp\nu\rho$ is a consequence of Condition (4). $\Box$

Now termination of the division-system (Ex. 9) can be proved by dependency pairs. Here we have $Ins_{\mathcal{E}}(Ext_{\mathcal{E}}(\mathcal{R})) = Ext_{\mathcal{E}}(\mathcal{R})$ and thus, the resulting constraints are

$$\begin{array}{ll}
\mathsf{M}(\mathsf{s}(x), \mathsf{s}(y)) > \mathsf{M}(x, y) & \mathsf{Q}(0 \div \mathsf{s}(y), z) > \mathsf{Q}(0, z) \\
\mathsf{Q}(\mathsf{s}(x), \mathsf{s}(y)) > \mathsf{M}(x, y) & \mathsf{Q}(\mathsf{s}(x) \div \mathsf{s}(y), z) > \mathsf{M}(x, y) \\
\mathsf{Q}(\mathsf{s}(x), \mathsf{s}(y)) > \mathsf{Q}(x - y, \mathsf{s}(y)) & \mathsf{Q}(\mathsf{s}(x) \div \mathsf{s}(y), z) > \mathsf{Q}(x - y, \mathsf{s}(y)) \\
 & \mathsf{Q}(\mathsf{s}(x) \div \mathsf{s}(y), z) > \mathsf{Q}(\mathsf{s}((x - y) \div \mathsf{s}(y)), z)
\end{array}$$

as well as $l \succsim r$ for all rules $l \to r$, $(u \div v) \div w \sim (u \div w) \div v$, and $\mathsf{Q}(u \div v, w) \sim \mathsf{Q}(u \div w, v)$. (Here, M and Q are the tuple symbols for the minus-symbol "$-$" and the quot-symbol "$\div$".) As explained in Sect. 2 one may again eliminate arguments of function symbols before searching for suitable orderings. In this example we will eliminate the second arguments of $-$, $\div$, M, and Q (i.e., every term $s - t$ is replaced by $-'(s)$, etc.). Then the resulting inequalities are satisfied by the rpo with the precedence $\div' \sqsupset \mathsf{s} \sqsupset -'$, $\mathsf{Q}' \sqsupset \mathsf{M}'$. Thus, with the method of the present paper, one can now verify termination of this example *automatically* for the first time. This example also demonstrates that by using dependency pairs, termination of equational rewriting can sometimes even be shown by *ordinary* base orderings (e.g., the ordinary rpo which on its own cannot be used for rewriting modulo equations).

6 Conclusion

We have extended the dependency pair approach to equational rewriting. In the special case of AC-axioms, our method is similar to the ones previously presented in [15, 17]. In fact, as long as the equations only consist of AC-axioms, one can show that using the instances $Ins_{\mathcal{E}}$ in Thm. 16 is not necessary.[7] (Hence, such a concept cannot be found in [17]). However, even then the only additional inequalities resulting from $Ins_{\mathcal{E}}$ are instantiations of other inequalities already present and inequalities which are special cases of an AC-deletion property (which is satisfied by all known AC-orderings and similar to the one required in [15]). This indicates that in practical examples with AC-axioms, our technique is at least as powerful as the ones of [15, 17] (actually, we conjecture that for AC-examples, these three techniques are virtually equally powerful). But compared to the approaches of [15, 17], our technique has a more elegant treatment of tuple symbols. (For example, if the TRS contains a rule $\mathsf{f}(t_1, t_2) \to \mathsf{g}(\mathsf{f}(s_1, s_2), s_3)$ were f and g are defined AC-symbols, then we do not have to extend the TRS by rules with tuple symbols like $\mathsf{f}(t_1, t_2) \to \mathsf{G}(\mathsf{f}(s_1, s_2), s_2)$ in [17]. Moreover, we do not need dependency pairs where tuple symbols occur outside the root position such as $\langle \mathsf{F}(\mathsf{F}(t_1, t_2), y), \ldots \rangle$ in [17] and [15] and $\langle \mathsf{F}(t_1, t_2), \mathsf{G}(\mathsf{F}(s_1, s_2), s_3) \rangle$ in [15]. Finally, we also do not need the "AC-marked condition" $\mathsf{F}(\mathsf{f}(x, y), z) \sim \mathsf{F}(\mathsf{F}(x, y), z)$ of [15].) But most significantly, unlike [15, 17] our technique works for *arbitrary* non-collapsing equations $\mathcal{E}$ with identical unique variables where $\mathcal{E}$-unification is finitary (for subterms of equations and left-hand sides of rules). Obviously, an implementation of our technique also requires $\mathcal{E}$-unification algorithms [5] for the concrete sets of equations $\mathcal{E}$ under consideration.

In [1–3], Arts and Giesl presented the *dependency graph* refinement which is based on the observation that it is possible to treat subsets of the dependency pairs separately. This refinement carries over to the equational case in a straightforward way (by using $\mathcal{E}$-unification to compute an estimation of this graph). For details on this refinement and for further examples to demonstrate the power and the usefulness of our technique, the reader is referred to [11].

Acknowledgments. We thank A. Middeldorp, T. Arts, and the referees for comments.

[7] Then in the proof of Thm. 16, instead of a minimal non-terminating term t' one regards a term t' which is non-terminating and minimal up to some extra f-occurrences on the top (where f is an AC-symbol).

References

1. T. Arts and J. Giesl, Automatically Proving Termination where Simplification Orderings Fail, in *Proc. TAPSOFT '97*, LNCS 1214, 261-272, 1997.
2. T. Arts and J. Giesl, Modularity of Termination Using Dependency Pairs, in *Proc. RTA '98*, LNCS 1379, 226-240, 1998.
3. T. Arts and J. Giesl, Termination of Term Rewriting Using Dependency Pairs, *Theoretical Computer Science*, 236:133-178, 2000.
4. T. Arts, System Description: The Dependency Pair Method, in *Proc. RTA '00*, LNCS 1833, 261-264, 2000.
5. F. Baader and W. Snyder, Unification Theory, in *Handbook of Automated Reasoning*, J. A. Robinson and A. Voronkov (eds.), Elsevier. To appear.
6. A. Ben Cherifa and P. Lescanne, Termination of Rewriting Systems by Polynomial Interpretations and its Implementation, *Sc. Comp. Prog.*, 9(2):137-159, 1987.
7. CiME 2. Pre-release available at http://www.lri.fr/~demons/cime-2.0.html.
8. C. Delor and L. Puel, Extension of the Associative Path Ordering to a Chain of Associative Commutative Symbols, in *Proc. RTA '93*, LNCS 690, 389-404, 1993.
9. N. Dershowitz, Termination of Rewriting, *J. Symbolic Computation*, 3:69-116, 1987.
10. M. C. F. Ferreira, Dummy Elimination in Equational Rewriting, in *Proc. RTA '96*, LNCS 1103, 78-92, 1996.
11. J. Giesl and D. Kapur, Dependency Pairs for Equational Rewriting, Technical Report TR-CS-2000-53, University of New Mexico, USA, 2000. Available from `http://www.cs.unm.edu/soe/cs/tech_reports`
12. J.-P. Jouannaud and H. Kirchner, Completion of a Set of Rules Modulo a Set of Equations, *SIAM Journal on Computing*, 15(4):1155-1194, 1986.
13. D. Kapur and G. Sivakumar, A Total Ground Path Ordering for Proving Termination of *AC*-Rewrite Systems, in *Proc. RTA '97*, LNCS 1231, 142-156, 1997.
14. D. Kapur and G. Sivakumar, Proving Associative-Commutative Termination Using RPO-Compatible Orderings, in *Proc. Automated Deduction in Classical and Non-Classical Logics*, LNAI 1761, 40-62, 2000.
15. K. Kusakari and Y. Toyama, On Proving *AC*-Termination by *AC*-Dependency Pairs, Research Report IS-RR-98-0026F, School of Information Science, JAIST, Japan, 1998. Revised version in K. Kusakari, Termination, *AC*-Termination and Dependency Pairs of Term Rewriting Systems, PhD Thesis, JAIST, Japan, 2000.
16. J.-P. Jouannaud and M. Muñoz, Termination of a Set of Rules Modulo a Set of Equations, in *Proc. 7th CADE*, LNCS 170, 175-193, 1984.
17. C. Marché and X. Urbain, Termination of Associative-Commutative Rewriting by Dependency Pairs, in *Proc. RTA '98*, LNCS 1379, 241-255, 1998.
18. H. Ohsaki, A. Middeldorp, and J. Giesl, Equational Termination by Semantic Labelling, in *Proc. CSL '00*, LNCS 1862, 457-471, 2000.
19. G. E. Peterson and M. E. Stickel, Complete Sets of Reductions for Some Equational Theories, *Journal of the ACM*, 28(2):233-264, 1981.
20. A. Rubio and R. Nieuwenhuis, A Total *AC*-Compatible Ordering based on RPO, *Theoretical Computer Science*, 142:209-227, 1995.
21. A. Rubio, A Fully Syntactic *AC*-RPO, *Proc. RTA-99*, LNCS 1631, 133-147, 1999.
22. J. Steinbach, Simplification Orderings: History of Results, *Fundamenta Informaticae*, 24:47-87, 1995.
23. L. Vigneron, Positive Deduction modulo Regular Theories, in *Proc. CSL '95*, LNCS 1092, 468-485, 1995.