A Dependency Pair Framework for $A \vee C$-Termination *

Beatriz Alarcón[1], Salvador Lucas[1], and José Meseguer[2]

[1] ELP group, DSIC, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, Spain
{balarcon,slucas}@dsic.upv.es
[2] CS Dept. University of Illinois at Urbana-Champaign, IL, USA

Abstract. The development of powerful techniques for proving termination of rewriting modulo a set of equations is essential when dealing with rewriting logic-based programming languages like CafeOBJ, Maude, OBJ, etc. One of the most important techniques for proving termination over a wide range of variants of rewriting (strategies) is the *dependency pair approach*. Several works have tried to adapt it to rewriting modulo *associative and commutative* (AC) equational theories, and even to more general theories. However, as we discuss in this paper, no appropriate notion of minimality (and minimal chain of dependency pairs) which is well-suited to develop a *dependency pair framework* has been proposed to date. In this paper we carefully analyze the structure of infinite rewrite sequences for rewrite theories whose equational part is a (free) *combination* of associative and commutative axioms which we call $A\vee C$*-rewrite theories*. Our analysis leads to a more accurate and optimized notion of dependency pairs through the new notion of *stably minimal term*. Then, we have developed a suitable dependency pair framework for proving termination of $A\vee C$-rewrite theories.

Key words: equational rewriting, termination, dependency pairs

1 Introduction

Rewriting with rules R modulo axioms E is a widely used technique in both rule-based programming languages and in automated deduction. Consequently, termination of rewriting modulo specific axioms E (e.g., associativity-commutativity, AC) has been studied. Methods for proving termination of rewriting systems modulo AC-axioms are known and even implemented. Several works have tried to adapt the *dependency pair approach* (DP-approach [1]) to rewriting modulo *associative and commutative* (AC) theories [13, 9–11, 14]. The corresponding

* Partially supported by EU (FEDER) and MICINN grant TIN 2007-68093-C02-02. José Meseguer has been partially supported by NSF Grants CCF-0905584, CNS-07-16038, and CNS-08-34709. Beatriz Alarcón was partially supported by the Spanish MEC/MICINN under FPU grant AP2005-3399.

```
fmod LIST&SET is
  sorts Bool Nat List Set .
  subsorts Nat < List Set .
  ops true false : -> Bool .
  ops _and_ _or_ : Bool Bool -> Bool [assoc comm] .
  op 0 : -> Nat .
  op s_ : Nat -> Nat .
  op _;_ : List List -> List [assoc] .
  op null : -> Set .
  op __ : Set Set -> Set [assoc comm] .
  op _in_ : Nat Set -> Bool .
  op _==_ : List List -> Bool [comm] .
  op list2set : List -> Set .
  var  B : Bool .              vars N M : Nat .
  vars L L' : List .           var  S : Set .
  eq N N = N .
  eq true and B = B .          eq false and B = false .
  eq true or B = true .        eq false or B = B .
  eq 0 == s N = false .        eq s N == s M = N == M .
  eq N ; L == M = false .      eq N ; L == M ; L' = (N == M) and L == L' .
  eq L == L = true .
  eq list2set(N) = N .         eq list2set(N ; L) = N list2set(L) .
  eq N in null = false .       eq N in M S = (N == M) or N in S .
endfm
```

Fig. 1. Example in Maude syntax [3]

proof methods, though, cannot be applied to commonly occurring combinations of axioms that fall outside their scope. For instance, they could not be applied to prove termination of the TRS in Figure 1, (specified in Maude with self-explanatory syntax; we would not care about sort information here) where we have a (free) *combination* of associative and commutative axioms which we call an *A*∨*C-rewrite theory* in this paper. Furthermore, the *Dependency Pair Framework* (DP-framework [6]), which is the basis of state-of-the-art tools for proving termination of (different variants of) term rewriting has not yet been adapted to the AC case.

In this paper, we address these two problems. Giesl and Kapur generalized the previous works on AC-termination with dependency pairs to deal with more general kinds of equational theories E satisfying some restrictions [5]. In principle, the *A*∨*C*-theories that we are going to investigate here fit Giesl and Kapur's approach. However, as we discuss below, they did not provide any definition of *minimal chain* needed for further developments in the DP-framework. In the DP-framework, the central notion regarding termination proofs is that of *DP problem*: the goal is checking the absence (or presence) of the so-called *infinite minimal chains*, where the notion of minimal chain can be thought as an abstraction of the infinite rewrite sequences starting from *minimal non-terminating* *terms*. The most important notion regarding mechanization of the proofs is that of *processor*. A (correct) processor basically transforms DP problems into (hopefully) *simpler* ones, in such a way that the existence of an infinite chain in the original DP problem implies the existence of an infinite chain in the transformed one. Here 'simpler' usually means that fewer pairs are involved. Processors are used in a pipe (more precissely, a *tree*) to incrementally simplify the original DP problem as much as possible, possibly decomposing it into smaller pieces which are then independently treated in the very same way. This is the crucial new feature of the DP-framework w.r.t. the DP-approach of [1]. This makes it so powerful as a basis for implementing termination provers.

Before being able to adapt the DP-framework to deal with $A\vee C$-theories, we start by giving a more refined notion of minimality. In fact, the notion of minimality which is used in [5] is the straightforward extension of the one which is used to prove termination of standard rewriting but without dealing with *equivalence preservation* which, as we show below, is essential to provide an appropriate notion of minimal non-E-terminating term for $A\vee C$-theories E which can be used to define a suitable $A\vee C$-DP-framework. We carefully analyze the structure of infinite rewrite sequences for $A\vee C$-rewrite theories. This leads to appropriate definitions of $A\vee C$-dependency pair and minimal chain.

After some technical preliminaries, in Section 3 we investigate the drawbacks of previous notions of minimal term when modeling infinite $A\vee C$-rewrite sequences. Then, we introduce the notion of *stably minimal* non-E-terminating term which is the basis of our development. Section 4 investigates the structure of infinite sequences starting from such stably minimal terms. Section 5 uses these results to formalize our notion of $A\vee C$-dependency pairs and minimal chains. Section 6 introduces an $A\vee C$-DP-framework for proving $A\vee C$-termination using $A\vee C$-DPs. in particular, we introduce the notion of $A\vee C$-dependency graph and a first processor for proving termination in the $A\vee C$-DP-framework. We also show how to use orderings for defining a second processor. Section 7 compares our approach with the related work and concludes.

2 Rewriting modulo equational theories

Given a rewrite theory $\mathcal{R} = (\Sigma, E, R)$, we write $s \rightarrow_{R/E} t$ if there exist u, v such that $s \sim_E u$, $u \rightarrow_R v$, and $v \sim_E t$. We say that a rewrite theory $\mathcal{R} = (\Sigma, E, R)$ is *E-terminating*, iff $\rightarrow_{R/E}$ is terminating. In general, given terms s and t, the problem of whether $s \rightarrow_{R/E} t$ holds is undecidable: in order to check whether $s \rightarrow_{R/E} t$ we have to search through the possibly infinite equivalence classes $[s]_E$ and $[t]_E$ to see whether a matching is found for a subterm of some $u \in [s]_E$ and the result of rewriting u belongs to the equivalence class $[t]_E$. For this reason, a much simpler relation $\rightarrow_{R,E}$ is defined, which becomes decidable if an E-matching algorithm exists. For any terms s, t, $s \rightarrow_{R,E} t$ holds iff there is a position p in s, a rule $l \rightarrow r$ in R, and a substitution σ such that $s|_p \sim_E \sigma(l)$ and $t = s[\sigma(r)]_p$ (see [15]). We say that a rewrite theory $\mathcal{R} = (\Sigma, E, R)$ is *(R, E)-terminating*, if $\rightarrow_{R,E}$ is terminating.

Regarding E-termination analysis using *dependency pairs* (DPs), Kusakari and Toyama observed that there is no simple extension of DPs to directly deal with $\rightarrow_{R/E}$-computations [11, 9]. In contrast, several approaches have been developed for $\rightarrow_{R,E}$-computations [5, 11, 13]. Since $\rightarrow_{R,E}\subseteq\rightarrow_{R/E}$ (but the opposite inclusion does not hold, in general), E-termination cannot be concluded from (R, E)-termination. Actually, Marché and Urbain showed that there are (R, E)-terminating rewrite theories $\mathcal{R}$ which are *not* E-terminating.

Example 1. Consider the following rewrite theory $\mathcal{R} = (\Sigma, E, R)$, where '+' is an AC symbol [13]: $a + b \rightarrow a + (b + c)$. Note that $t = a + (b + c)$ is an $\rightarrow_{R,E}$-normal form (hence (R, E)-terminating). However, $t \sim_{AC} (a + b) + c$ which is non-E-terminating.

Giesl and Kapur [5] proved the equivalence of both notions of termination with respect to a notion of *extension completion* $\mathcal{E}xt_E(R)$ of a rewrite theory $\mathcal{R} = (\Sigma, E, R)$ which, for E being a set containing associative or commutative axioms, goes back to Peterson and Stickel [15].

Theorem 1. [5, Theorem 11] *Let $\mathcal{R} = (\Sigma, E, R)$ be a rewrite theory with E a regular and linear equational theory and $t \in \mathcal{T}(\Sigma, \mathcal{X})$. Then, t starts an infinite $\rightarrow_{R/E}$-reduction if and only if t starts an infinite $\rightarrow_{\mathcal{E}xt_E(R),E}$-reduction. Therefore, $\mathcal{R}$ is E-terminating if and only if $\rightarrow_{\mathcal{E}xt_E(R),E}$ is terminating.*

2.1 Combination of Associative and Commutative Theories

Let E be a set of equations that has the modular decomposition $E = \bigcup_{f\in\Sigma} E_f$, where if $k = ar(f) \neq 2$, then $E_f = \varnothing$, and if $k = 2$, then $E_f \subseteq \{A_f, C_f\}$, where:

- A_f is the associativity axiom $f(f(x, y), z) = f(x, f(y, z))$,
- C_f is the commutativity axiom $f(x, y) = f(y, x)$.

We also define $\Sigma = \Sigma_A \uplus \Sigma_C \uplus \Sigma_{AC} \uplus \Sigma_\varnothing$ where $f \in \Sigma_A \Leftrightarrow E_f = \{A_f\}$, $f \in \Sigma_C \Leftrightarrow E_f = \{C_f\}$, $f \in \Sigma_{AC} \Leftrightarrow E_f = \{A_f, C_f\}$, $f \in \Sigma_\varnothing \Leftrightarrow E_f = \varnothing$. In the following, we often say that a symbol $f \in \Sigma$ *is associative* if $f \in \Sigma_A \cup \Sigma_{AC}$.

Definition 1 ($A{\vee}C$-rewrite theory). *An equational theory $E = \bigcup_{f\in\Sigma} E_f$, where if $k = ar(f) \neq 2$, then $E_f = \varnothing$, and if $k = 2$, then $E_f \subseteq \{A_f, C_f\}$ is called an $A{\vee}C$-*theory*. A rewrite theory $\mathcal{R} = (\Sigma, E, R)$ such that E is an $A{\vee}C$-theory, is called an $A{\vee}C$-*rewrite theory*.

To deal with rewriting modulo $A{\vee}C$-theories by using (R, E)-rewriting we have to extend R by following [15, Definition 10.4]:

$$\begin{aligned}
\mathcal{E}xt_{AC}(R) &= R \cup \{f(l, w) \rightarrow f(r, w) \mid l \rightarrow r \in R, f = root(l) \in \Sigma_{AC}\}\\
\mathcal{E}xt_A(R) &= R \cup \{f(l, w) \rightarrow f(r, w), f(w, l) \rightarrow f(w, r), f(z, f(l, w)) \rightarrow f(z, f(r, w))\\
&\qquad \mid l \rightarrow r \in R, f = root(l) \in \Sigma_A\}\\
\mathcal{E}xt_C(R) &= R
\end{aligned}$$

where w and z are fresh variables which do not occur in the original rule of R. Therefore, given an $A{\vee}C$ theory E, we let: $\mathcal{E}xt_E(R) = \mathcal{E}xt_{AC}(R) \cup \mathcal{E}xt_A(R) \cup \mathcal{E}xt_C(R)$. Note that $R \subseteq \mathcal{E}xt_E(R)$.

2.2 Minimal terms and infinite rewrite sequences

Given a TRS $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$, with $\mathcal{C}$ a subsignature of constructors and $\mathcal{D}$ a subsignature of defined symbols, the *minimal* nonterminating terms associated to $\mathcal{R}$ are nonterminating terms t whose proper subterms u (i.e., $t \rhd u$) are terminating; $\mathcal{T}_\infty$ is the set of minimal nonterminating terms associated to $\mathcal{R}$ [7]. Minimal nonterminating terms have two important properties:

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

Considering the structure of the infinite rewrite sequences starting from a minimal nonterminating term $t = f(t_1, \ldots, t_k) \in \mathcal{T}_\infty$ is helpful to arrive at the notion of dependency pair. Such sequences proceed as follows (see, e.g., [7]):

1. a finite number of reductions can be performed *below* the root of t, thus rewriting t into t'; then
2. a rule $f(l_1, \ldots, l_k) \to r$ applies *at the root* of t' (i.e., $t' = \sigma(f(l_1, \ldots, l_k))$ for some substitution σ); and
3. there is a minimal nonterminating term $u \in \mathcal{T}_\infty$ (hence $root(u) \in \mathcal{D}$) at some position p of $\sigma(r)$ which is a *nonvariable position of* r which 'continues' the infinite sequence initiated by t in a similar way.

This means that considering the occurrences of defined symbols in the right-hand sides of the rewrite rules suffices to 'catch' *every possible infinite rewrite sequence starting from* $\sigma(r)$. In particular, no infinite sequence can be issued from t' *below* the variables of r (more precisely: all bindings $\sigma(x)$ are *terminating* terms). The standard definition of dependency pair [1] and (minimal) chain of dependency pairs [6] relies on (1)–(3) above [7]. These facts are formalized as follows:

Proposition 1. [7, Lemma 1] *Let $\mathcal{R} = (\mathcal{C} \uplus \mathcal{D}, R)$ be a TRS. For all $t \in \mathcal{T}_\infty$, there exist $l \to r \in R$, a substitution σ and a term $u \in \mathcal{T}_\infty$ such that $root(u) \in \mathcal{D}$, $t \xrightarrow{>\Lambda}{}^* \sigma(l) \xrightarrow{\Lambda} \sigma(r) \unrhd u$ and there is a nonvariable subterm v of r, $r \unrhd v$, such that $u = \sigma(v)$.*

In the following section we begin the analysis of infinite E-rewrite sequences according to this schema. We aim at providing an appropriate notion of minimal non-E-terminating term (for $A{\vee}C$-theories E) which allows us to reach a result similar to Proposition 1.

3 Stably minimal non-E-terminating terms

In the dependency pair approach [1, 7, 6], the analysis of infinite rewrite sequences is restricted to those starting from *minimal nonterminating terms* $t \in \mathcal{T}_\infty$. The following notion of minimal non-E-terminating term is implicit in [5, proof of Theorem 16]. Similar definitions can be found in [10, 11, 9, 14].

Definition 2 (Minimal non-E-terminating term [5]). *Let $\mathcal{R} = (\Sigma, E, R)$ be a rewrite theory. A non-E-terminating term $t \in \mathcal{T}(\Sigma, \mathcal{X})$ is said to be* minimal *(written $t \in \mathcal{T}_{\infty,R,E}$) if every strict subterm s of t (i.e., $t \rhd s$) is $(\mathcal{E}xt_E(R), E)$-terminating.*

Remark 1. In Definition 2, if we assume that E is linear and regular (like $A{\vee}C$-theories), then, by Theorem 1, we could equivalently start by saying that t is non-$(\mathcal{E}xt_E(R), E)$-terminating. This leads to a more symmetric definition which we often use in the following without further comment.

Every non-E-terminating term s contains a minimal non-E-terminating term $t \in \mathcal{T}_{\infty,R,E}$ (this is stated without proof in [5, proof of Theorem 16]).

Remark 2 (Root symbols of minimal terms). Note that, if E is an $A{\vee}C$-equational theory, then $root(t) \in \mathcal{D}$ whenever $t \in \mathcal{T}_{\infty,R,E}$. As remarked by Giesl and Kapur (see also Example 5 below) this is not true for arbitrary equational theories.

The problem with Giesl and Kapur's Definition 2 is that minimality is *not* preserved under E-equivalence.

Example 2. Consider the following TRS $\mathcal{R}$:

$$f(x, x) \to f(0, f(1, 2)) \tag{1}$$

where $f \in \Sigma_{AC}$. Hence, $\mathcal{E}xt_{AC}(R)$ only adds the following rule to $\mathcal{R}$:

$$f(f(x, x), y) \to f(f(0, f(1, 2)), y) \tag{2}$$

Note that $t = f(f(0, 1), f(0, f(1, 2)))$ is non-$(\mathcal{E}xt_{AC}(R), AC)$-terminating:

$$\underline{f(f(0,1), f(0, f(1,2)))} \sim_A f(0, \underline{f(1, f(0, f(1,2)))}) \sim_A f(0, f(\underline{f(1,0)}, f(1,2))) \sim_C f(0, \underline{f(f(0,1), f(1,2))}) \sim_A \underline{f(0, f(0, f(1, f(1,2))))} \sim_A \underline{f(f(0,0), f(1, f(1,2)))} \xrightarrow{\Lambda}_{\mathcal{E}xt_{AC}(R)} f(f(0, f(1,2)), f(1, f(1,2))) \to_{\mathcal{E}xt_{AC}(R),AC} \cdots$$

Since $f(0, 1)$ and $f(0, f(1, 2))$ are in $(\mathcal{E}xt_{AC}(R), AC)$-normal form, we have that $t \in \mathcal{T}_{\infty,R,AC}$. However, $t' = f(f(0, 0), f(1, f(1, 2)))$, which is AC-equivalent to t (i.e., $t \sim_{AC} t'$), is non-AC-terminating but it is *not* minimal because its strict subterm $f(1, f(1, 2)))$ is non-$(\mathcal{E}xt_{AC}(R), AC)$-terminating:

$$\underline{f(1, f(1,2)))} \sim_A \underline{f(f(1,1), 2)} \xrightarrow{\Lambda}_{\mathcal{E}xt_{AC}(R)} \underline{f(f(0, f(1,2)), 2)} \sim_A f(0, \underline{f(f(1,2), 2)}) \sim_A \underline{f(0, f(1, f(2,2)))} \sim_A \underline{f(f(0,1), f(2,2))} \sim_C \underline{f(f(2,2), f(0,1))} \xrightarrow{\Lambda}_{\mathcal{E}xt_{AC}(R)} f(f(0, f(1,2)), f(0,1)) \to_{\mathcal{E}xt_{AC}(R),AC} \cdots$$

Example 2 shows that an essential property of minimal terms when considered as part of infinite $(\mathcal{E}xt_E(R), E)$-rewriting sequences for $A{\vee}C$-theories E gets lost: the application of $(\mathcal{E}xt_E(R), E)$-rewrite steps *at the root* of a minimal term s by means of a rule $l \to r$ (i.e., $s \sim_{AC} \sigma(l) \xrightarrow{\Lambda}_{\mathcal{E}xt_E(R)} \sigma(r)$) does *not* guarantee that there is a *nonvariable subterm* v of the right-hand side r which is a prefix of the 'next' minimal term in the infinite sequence.

Example 3. Term t in Example 2 can be rewritten at the root *only* by rule (2) of $\mathcal{E}xt_{AC}(R)$. We can apply this rule to t' in Example 2 (for instance) to obtain $s' = \sigma(r) = f(f(0, f(1,2)), f(1, f(1,2)))$ (where $r = f(f(0, f(1,2)), y))$, which is non-$(\mathcal{E}xt_{AC}(R), AC)$-terminating. Note that s' contains a minimal term $u \in \mathcal{T}_{\infty,R,E}$. Since $s'|_2 = f(1, f(1,2))$ is non-$(\mathcal{E}xt_{AC}(R), AC)$-terminating, it follows that s' is *not* minimal. Since $s'|_1 = f(0, f(1,2))$ is $(\mathcal{E}xt_{AC}(R), AC)$-terminating, the only possibility is that u occurs in $s'|_2$. Actually, $s'|_2$ is minimal already; hence, $u = s'|_2$. But note the absence of any nonvariable position $p \in \mathcal{P}os(r)$ in the right-hand side of the considered rule such that $\sigma(r|_p) = u = f(1, f(1,2))$.

This is in sharp contrast with the situation of the DP-approach for ordinary rewriting. Furthermore, it is not difficult to see that for all $t'' \sim_{AC} t$ such $t'' = \sigma'(l)$ for some substitution σ', we have a similar situation. Thus, the problem illustrated here cannot be solved by using a different $\sim_{AC}$ sequence before performing the $\mathcal{E}xt_{AC}(R)$-root-step.

In the following we introduce a new notion of minimality which solves these problems.

3.1 A new notion of minimal non-E-terminating terms

The following definition solves the problems discussed above by explicitly requiring that the condition defining minimality is preserved under E-equivalence.

Definition 3 (Stably minimal non-E-terminating term). *Let $\mathcal{R} = (\Sigma, E, R)$ be a rewrite theory. Let $\mathcal{M}_{\infty,R,E}$ be a set of stably minimal non-E-terminating terms in the following sense: $t \in \mathcal{T}(\Sigma, \mathcal{X})$ belongs to $\mathcal{M}_{\infty,R,E}$ if t is non-E-terminating, and for all $t' \sim_E t$ and every proper subterm s' of t' (i.e., $t' \rhd s'$), s' is $(\mathcal{E}xt_E(R), E)$-terminating.*

We have the following useful characterization of minimality.

Proposition 2 (Characterization of stably minimal terms). *Let $\mathcal{R} = (\Sigma, R, E)$ be a rewrite theory and $t \in \mathcal{T}(\Sigma, \mathcal{X})$. Then, $t \in \mathcal{M}_{\infty,R,E}$ if and only if $[t]_E \subseteq \mathcal{T}_{\infty,R,E}$. Therefore, $\mathcal{M}_{\infty,R,E} = \{t \in \mathcal{T}(\Sigma, \mathcal{X}) \mid [t]_E \subseteq \mathcal{T}_{\infty,R,E}\}$.*

The problem in Example 2 disappears now: t is *not* minimal according to Definition 3. The following result shows how to *find* stably minimal non-E-terminating terms associated to a given non-E-terminating term. This is essential in our development.

Proposition 3. *Let $\mathcal{R} = (\Sigma, E, R)$ be a rewrite theory such that $[t]_E = \{t\}$ for all constant and variable terms t. Let $s \in \mathcal{T}(\Sigma, \mathcal{X})$. If s is non-E-terminating, then there is a subterm t of some $s' \sim_E s$ ($s' \unrhd t$) such that $t \in \mathcal{M}_{\infty,R,E}$.*

Clearly, Proposition 3 holds whenever $\mathcal{R}$ is an $A\vee C$-rewrite theory.

Example 4. Consider the term t in Example 2. Although $t \in \mathcal{T}_{\infty,R,E}$, $t \notin \mathcal{M}_{\infty,R,E}$: the term $t' = f(f(0,0), f(1, f(1,2)))$, which is AC-equivalent to t, contains a subterm $u = f(1, f(1,2))$ which is non-E-terminating. It is not difficult to see that actually $u \in \mathcal{M}_{\infty,R,E}$.

In general, Proposition 3 does *not* hold for arbitrary sets of equations E.

Example 5. Consider the following example [5, Example 13]:

$$R : f(x) \to x \qquad E : f(a) = a$$

Note that $a \in \mathcal{T}_{\infty,R,E}$. However, a is *not* stably minimal because $a \sim_E f(a)$ but $f(a) \notin \mathcal{T}_{\infty,R,E}$. Thus, Proposition 3 does not hold.

Now we provide a more precise result about where we can find stably minimal subterms within a non-E-terminating term for $A{\vee}C$-rewrite theories $\mathcal{R} = (\Sigma, E, R)$. In the following theorem, given a term s and a symbol f, by an f-subterm t of s (written $s \trianglerighteq_f t$) we mean a subterm t of s such that $t = s|_p$ and for all $q < p$, $root(s|_q) = f$. We also write $s \triangleright_f t$ if $s \trianglerighteq_f t$ and $s \neq t$.

Theorem 2. *Let $\mathcal{R} = (\Sigma, E, R)$ be an $A{\vee}C$-rewrite theory. If s is non-E-terminating, then there is a subterm $t \in \mathcal{T}_{\infty,R,E}$ of s ($s \trianglerighteq t$) and*

1. *If (1) $A_{root(t)} \notin E_{root(t)}$ or (2) $t = f(t_1, t_2)$, $A_f \in E_f$, $root(t_1) \neq f$, and $root(t_2) \neq f$, then $t \in \mathcal{M}_{\infty,R,E}$.*
2. *If $t = f(t_1, t_2)$, $A_f \in E_f$, and $root(t_1) = f$ or $root(t_2) = f$, and $t \notin \mathcal{M}_{\infty,R,E}$, then there is $s' \sim_E t$ and a strict f-subterm u of s' ($s' \triangleright_f u$) such that $root(u) = f$ and $u \in \mathcal{M}_{\infty,R,E}$.*

The following result is just a convenient reformulation of the previous one.

Corollary 1. *Let $\mathcal{R} = (\Sigma, E, R)$ be an $A{\vee}C$-rewrite theory. If s is non-E-terminating, then either there is a subterm $t \in \mathcal{M}_{\infty,R,E}$ of s ($s \trianglerighteq t$), or there is a subterm $t \in \mathcal{T}_{\infty,R,E}$ of s satisfying that $t = f(t_1, t_2)$, $A_f \in E_f$, and $root(t_1) = f$ or $root(t_2) = f$, and such that there is $s' \sim_E t$ and a strict f-subterm u of s' ($s' \triangleright_f u$) such that $root(u) = f$ and $u \in \mathcal{M}_{\infty,R,E}$.*

4 Structure of (stably) minimal infinite $A{\vee}C$-rewrite sequences

Now we analyze $A{\vee}C$-rewrite sequences starting from stably minimal non-$A{\vee}C$-terminating terms. First we consider a restricted case.

Proposition 4. *Let $\mathcal{R} = (\Sigma, E, R) = (\mathcal{C} \uplus \mathcal{D}, E, R)$ be an $A{\vee}C$-rewrite theory. Let $s \in \mathcal{M}_{\infty,R,E}$ be such that $f = root(s)$ and either (1) $A_f \notin E_f$, or (2) $s = f(s_1, s_2)$, $A_f \in E_f$, and $root(s_1)$, $root(s_2) \in \mathcal{C}$. Assume that for all $l \to r \in R$ such that $root(l) = f$ and all subterms v of r ($r \trianglerighteq v$) such that $v = g(v_1, v_2)$ for some associative symbol g, we have that $root(v_1), root(v_2) \notin \mathcal{X} \cup \{g\}$. Then, there exist $l \to r \in R$, a substitution σ and terms $t \in \mathcal{T}(\Sigma, \mathcal{X})$ and $u \in \mathcal{M}_{\infty,R,E}$ such that*

$$s \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}xt_E(R),E} t \sim_E \sigma(l) \xrightarrow{\Lambda}_R \sigma(r) \trianglerighteq u$$

and there is a nonvariable subterm v of r, $r \trianglerighteq v$, such that $u = \sigma(v)$.

Unfortunately, stable minimality of (arbitrary) non-E-terminating terms s for $A{\vee}C$-theories E is not preserved under inner $(\mathcal{E}xt_E(R), E)$-rewritings.

Example 6. Term $u = f(f(1,1),2)$ in Example 2 is stably minimal: $u \in \mathcal{M}_{\infty,R,E}$. We have that $f(\underline{f(1,1)},2) \xrightarrow{>\Lambda}_R f(f(0,f(1,2)),2)$. Note that $f(f(0,f(1,2)),2) \notin \mathcal{M}_{\infty,R,E}$: we have $\underline{f(f(0,f(1,2)),2)} \sim_A f(0,\underline{f(f(1,2),2))} \sim_A f(0,f(1,f(2,2)))$ where $f(0,f(1,\underline{f(2,2)}))$ contains a subterm $f(1,f(2,2))$ which is non-$(\mathcal{E}xt_E(R),E)$-terminating.

In the following, we show how to avoid this problem. We define *deep reduction* as a restriction $\xrightarrow{>1,2}_{\mathcal{E}xt_E(R),E}$ of inner $(\mathcal{E}xt_E(R), E)$-rewriting which restricts reductions on terms like u above. We will show that deep reduction preserves stable minimality of non-E-terminating terms for $A{\vee}C$-rewrite theories $\mathcal{R} = (\Sigma, E, R)$.

Definition 4 (Deep reduction). *Let $\mathcal{R} = (\Sigma, E, R)$ be an $A{\vee}C$-rewrite theory. Given $t \in \mathcal{T}(\Sigma, \mathcal{X})$, $t \xrightarrow{>1,2}_{\mathcal{E}xt_E(R),E} s$ if $t \xrightarrow{q}_{\mathcal{E}xt_E(R),E} s$ for some position $q \in \mathcal{P}os(t)$ such that $q > p$ for $p \in \{1,2\}$ if $t = \sigma(u)$ for some $u = v \in E$ or $v = u \in E$ and $u|_p \notin \mathcal{X}$; otherwise, $q > \Lambda$.*

Obviously, $\xrightarrow{>1,2}_{\mathcal{E}xt_E(R),E} \subseteq \xrightarrow{>\Lambda}_{\mathcal{E}xt_E(R),E}$. The following proposition shows that *deep reduction* preserves stable minimality.

Proposition 5. *Let $\mathcal{R} = (\Sigma, E, R)$ be an $A{\vee}C$-rewrite theory and $t \in \mathcal{M}_{\infty,R,E}$. If $t \xrightarrow{>1,2}{}^{*}_{\mathcal{E}xt(\mathcal{R}),E} s$ and s is non-E-terminating, then $s \in \mathcal{M}_{\infty,R,E}$.*

As a consequence, the following theorem establishes the desired property for stable minimal non-$A{\vee}C$-terminating terms.

Theorem 3. *Let $\mathcal{R} = (\Sigma, E, R)$ be an $A{\vee}C$-rewrite theory. For all $s \in \mathcal{M}_{\infty,R,E}$, there exist $l \to r \in \mathcal{E}xt_E(R)$ and a substitution σ such that*

$$s\ (\sim_E \circ \xrightarrow{>1,2}_{\mathcal{E}xt_E(R),E})^* t \sim_E \sigma(l) \xrightarrow{\Lambda}_{\mathcal{E}xt_E(R)} \sigma(r)$$

and there is a nonvariable subterm v of r $(r \trianglerighteq v)$, such that either

1. *$v = f(v_1, v_2)$ for some associative symbol f, $root(v_1) \in \mathcal{X} \cup \{f\}$ or $root(v_2) \in \mathcal{X} \cup \{f\}$, $root(\sigma(v_1)) = f$ or $root(\sigma(v_2)) = f$, $\sigma(v) \in \mathcal{T}_{\infty,R,E}$ and there is a term $t' \sim_E \sigma(v)$ containing a strict f-subterm $u = f(u_1, u_2)$ $(t' \triangleright_f u)$ such that $u \in \mathcal{M}_{\infty,R,E}$, or*
2. *$\sigma(v) \in \mathcal{M}_{\infty,R,E}$ otherwise.*

Example 2 shows that Theorem 3 does not hold for Giesl and Kapur's minimal terms $s \in \mathcal{T}_{\infty,R,E}$.

5 $A \vee C$-Dependency Pairs and Chains

Propositions 3 and 4 together with Theorem 3 are the basis for our definition of *$A\vee C$-Dependency Pairs* and the corresponding *chains*. Together, they show that given an $A\vee C$-rewrite theory $\mathcal{R} = (\Sigma, E, R)$, every non-$E$-terminating term s has an associated infinite $(\mathcal{E}xt_E(R), E)$-rewrite sequence starting from a stably minimal subterm $t \in \mathcal{M}_{\infty,R,E}$. Such a sequence proceeds as described in Proposition 4 and Theorem 3, depending on the shape of t.

This process is abstracted in the following definition of $A\vee C$-dependency pairs (Definition 5) and in the definition of chain below (Definition 6).

Given a signature Σ and $f \in \Sigma$, we let $f^\sharp$ denote a fresh new symbol (often called *tuple* symbol or DP-symbol) associated to a symbol f [1]. Let $\Sigma^\sharp$ be the set of tuple symbols associated to symbols in Σ. As usual, for $t = f(t_1, \ldots, t_k) \in \mathcal{T}(\Sigma, \mathcal{X})$, we write $t^\sharp$ to denote the *marked* term $f^\sharp(t_1, \ldots, t_k)$ (written sometimes $F(t_1, \ldots, t_k)$). Given a set of rules R and a symbol $f \in \Sigma$, we let $R_f = \{l \to r \in R \mid root(l) = f\}$. Given a set of rules R, the set $\mathsf{DP}(R)$ of dependency pairs associated to R is [1]: $\mathsf{DP}(R) = \{l^\sharp \to s^\sharp \mid l \to r \in R, r \trianglerighteq s, root(s) \in \mathcal{D}\}$.

Definition 5 ($A\vee C$-Dependency Pairs). *Let $\mathcal{R} = (\Sigma, E, R) = (\mathcal{C} \uplus \mathcal{D}, E, R)$ be an $A\vee C$-rewrite theory. Then, $\mathsf{DP}_E(R) = \mathsf{DP}(\mathcal{E}xt_E(R))$ is the set of $A\vee C$-dependency pairs ($A\vee C$-DPs) of $\mathcal{R}$.*

In general, the set of $A\vee C$-DPs which is obtained from Definition 5 is a subset of those which are obtained by particularizing Giesl and Kapur's definitions to the $A\vee C$ case [5].

Example 7. Consider the AC-rewrite theory $\mathcal{R} = (\Sigma, E, R)$ in Example 2. The set $DP_E(R)$ consists of the following pairs:

$$F(x, x) \to F(0, f(1, 2)) \quad (3)$$

$$F(x, x) \to F(1, 2) \quad (4)$$

$$F(f(x, x), y) \to F(f(0, f(1, 2)), y) \quad (5)$$

$$F(f(x, x), y) \to F(0, f(1, 2)) \quad (6)$$

$$F(f(x, x), y) \to F(1, 2) \quad (7)$$

5.1 Chains of $A\vee C$-DPs

An essential property of the dependency pair method is that it provides a *characterization* of termination of TRSs $\mathcal{R}$ as the absence of infinite (minimal) *chains of dependency pairs* [1, 6]. If we want to prove the same for $A\vee C$-rewrite theories, we have to introduce a suitable notion of chain which can be used with $A\vee C$-DPs. As in the DP-framework, where the origin of *pairs* does not matter, we should rather think of another rewrite theory $\mathcal{P} = (\Gamma, F, P)$ which is used together with $\mathcal{R}$ to build the chains. According to the usual terminology [6], we often call *pairs* to the rules $u \to v \in P$.

Definition 6 (Chain of pairs - Minimal chain). *Let $\mathcal{P} = (\Gamma, F, P)$ and $\mathcal{R} = (\Sigma, E, R)$ be rewrite theories, and $\mathcal{S} = (\mathcal{F}, S)$ be a TRS. An (F, P, E, R, S)-chain is a finite or infinite sequence of pairs $u_i \to v_i \in P$, together with substitutions σ and θ_i satisfying that, for all $i \geq 1$:*

1. *If $\sigma(v_i) = f_i(v_{i1}, v_{i2})$ satisfies $\sigma(v_i) = \theta_i(u'_i)$ for some $u'_i = v'_i \in F$ or $v'_i = u'_i \in F$ such that $u'_i = f_i(u'_{i1}, u'_{i2})$ satisfies $u'_{i1} \notin \mathcal{X}$ or $u'_{i2} \notin \mathcal{X}$, then*

$$\sigma(v_i) \sim_F \circ \xrightarrow{\Lambda}{}^{+}_{\mathcal{S}_{f_i}} t_i \, (\sim_F \circ \xrightarrow{>1,2}_{\mathcal{E}xt_E(R),E})^* \circ \sim_F \sigma(u_{i+1})$$

2. *and $\sigma(v_i) = t_i \to^*_{\mathcal{E}xt_E(R),E} \circ \sim_F \sigma(u_{i+1})$, otherwise.*

An (F, P, E, R, S)-chain is called minimal *if for all $i \geq 1$, and $t'_i \in [t_i]_F$, t'_i is $(\mathcal{E}xt_E(R), E)$-terminating.*

As usual, in Definition 6 we assume that different occurrences of dependency pairs do not share any variable (renaming substitutions are used if necessary).

This more abstract notion of chain can be particularized to be used with $A\vee C$-DPs, by just taking

1. $P = \mathsf{DP}_E(R)$,
2. $F = E \cup E^\sharp$, where $E^\sharp = \{s^\sharp = t^\sharp \mid s = t \in E\}$, and
3. $\mathcal{S} = \{f^\sharp(f(x, y), z) \to f^\sharp(x, y), f^\sharp(x, f(y, z)) \to f^\sharp(y, z) \mid f \in \Sigma_A \cup \Sigma_{AC}\}$.

Theorem 4 (Characterization of $A\vee C$-termination). *Let $\mathcal{R} = (\Sigma, E, R)$ be an $A\vee C$-rewrite theory. Let $\mathcal{S} = (\Sigma \cup \mathcal{D}^\sharp, S)$ be a TRS such that $S = \{f^\sharp(f(x, y), z) \to f^\sharp(x, y), f^\sharp(x, f(y, z)) \to f^\sharp(y, z) \mid f \in \Sigma_A \cup \Sigma_{AC}\}$. Then, $\mathcal{R}$ is $(\mathcal{E}xt_E(R), E)$-terminating if and only if there is no infinite minimal $(E^\sharp \cup E, \mathsf{DP}_E(R), E, R, S)$-chain.*

6 An $A\vee C$-Dependency Pair Framework

In the following, we adapt Giesl et al. DP-framework to provide a suitable framework for mechanizing proofs of $A\vee C$-termination using $A\vee C$-DPs.

Definition 7 ($A\vee C$ problem). *An $A\vee C$ problem τ is a tuple $\tau = (F, P, E, R, S)$, where $\mathcal{R} = (\Sigma, E, R)$ is an $A\vee C$-rewrite theory, $\mathcal{P} = (\Gamma, F, P)$ is a rewrite theory, and $\mathcal{S} = (\mathcal{F}, S)$ is a TRS. An $A\vee C$ problem is finite if there is no infinite minimal (F, P, E, R, S)-chain. An $A\vee C$ problem τ is infinite if $\mathcal{R}$ is non-$A\vee C$-terminating or there is an infinite minimal (F, P, E, R, S)-chain.*

The following definition adapts the notion of *processor* [6] to prove termination of $A\vee C$-rewrite theories.

Definition 8 ($A\vee C$ processor). *An $A\vee C$ processor* Proc *is a mapping from $A\vee C$ problems into sets of $A\vee C$ problems. Alternatively, it can also return* "no"*. An $A\vee C$ processor* Proc *is*

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

Similar to [6] for the DP-framework, we construct a tree whose nodes are labeled with $A \vee C$ problems or "yes" or "no", and whose root is labeled with $(E^\sharp \cup E, \mathsf{DP}_E(R), E, R, S)$. Now we have the following result which adapts [6, Corollary 5] to $A \vee C$-rewrite theories.

Theorem 5 **(**$A \vee C$**-DP framework).** *Let* $\mathcal{R} = (\Sigma, E, R)$ *be an* $A \vee C$*-theory. We construct a tree whose nodes are labeled with* $A \vee C$ *problems or* "yes" *or* "no"*, and whose root is labeled with* $(E^\sharp \cup E, \mathsf{DP}_E(R), E, R, S)$*, where* $S = \{f^\sharp(f(x,y),z) \to f^\sharp(x,y), f^\sharp(x, f(y,z)) \to f^\sharp(y,z) \mid f \in \Sigma_A \cup \Sigma_{AC}\}$*. For every inner node labeled with* τ*, there is a sound processor* Proc *satisfying one of the following conditions:*

1. $\mathsf{Proc}(\tau) = \mathsf{no}$ *and the node has just one child, labeled with* "no"*.*
2. $\mathsf{Proc}(\tau) = \varnothing$ *and the node has just one child, labeled with* "yes"*.*
3. $\mathsf{Proc}(\tau) \neq \mathsf{no}$*,* $\mathsf{Proc}(\tau) \neq \varnothing$*, and the children of the node are labeled with the* $A \vee C$ *problems in* $\mathsf{Proc}(\tau)$*.*

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

6.1 $A \vee C$-Dependency Graph

$A \vee C$ problems focus our attention on the analysis of *infinite minimal chains*. Our aim here is obtaining a notion of graph which is able to represent all infinite *minimal* chains of pairs as given in Definition 6.

Definition 9 **(**$A \vee C$**-Graph of Pairs).** *Let* $\mathcal{R} = (\Sigma, E, R)$ *and* $\mathcal{P} = (\Gamma, F, P)$ *be rewrite theories and* $\mathcal{S} = (\mathcal{F}, S)$ *be a TRS. The* $A \vee C$-graph *associated to them (denoted* $\mathsf{G}(F, P, E, R, S)$*) has* P *as the set of nodes. There is an arc from* $u \to v \in P$ *to* $u' \to v' \in P$ *if* $u \to v, u' \to v'$ *is an* (F, P, E, R, S)*-chain.*

In termination proofs, we are concerned with the so-called *strongly connected components* (SCCs) of the dependency graph, rather than with the cycles themselves (which are exponentially many) [8]. A strongly connected component in a graph is a *maximal cycle*, i.e., a cycle which is not contained in any other cycle. In the following result, given two sets of rules S and Q, we let S_Q be the least subset of S satisfying that whenever there is a rule $u \to v \in Q$, such that v unifies with s for some $s = t \in F$ or $t = s \in F$ such that $s = f(s_1, s_2)$ and $s_1 \notin \mathcal{X}$ or $s_2 \notin \mathcal{X}$, then $S_f \subseteq S_Q$.

Theorem 6 (SCC processor). *Let* $\mathcal{R} = (\Sigma, E, R)$ *and* $\mathcal{P} = (\Gamma, F, P)$ *be rewrite theories and* $\mathcal{S} = (\mathcal{F}, S)$ *be a TRS. Then, the processor* Proc_{SCC} *given by*

$$\mathsf{Proc}_{SCC}(F, P, E, R, S) = \{(F, Q, E, R, S_Q) \mid Q \text{ are the pairs of an SCC in } \mathsf{G}(F, P, E, R, S)\}$$

is sound and complete.

As a consequence, we can *separately* work with the strongly connected components of $\mathsf{G}(F, P, E, R, S)$, disregarding other parts of the graph. Now we can use these notions to introduce the $A{\vee}C$-dependency graph, i.e., the $A{\vee}C$-graph whose nodes are the $A{\vee}C$-DPs instead of an arbitrary set of pairs.

Definition 10 ($A{\vee}C$-Dependency Graph). *Let $\mathcal{R} = (\Sigma, E, R)$ be an $A{\vee}C$-rewrite theory with $\Sigma = \mathcal{C} \uplus \mathcal{D}$. Let $\mathcal{S} = (\Sigma \cup \mathcal{D}^\sharp, S)$ be a TRS such that $S = \{f^\sharp(f(x, y), z) \to f^\sharp(x, y), f^\sharp(x, f(y, z)) \to f^\sharp(y, z) \mid f \in \Sigma_A \cup \Sigma_{AC}\}$. The $A{\vee}C$-*Dependency Graph *associated to $\mathcal{R}$ is:* $\mathsf{DG}(\mathcal{R}) = \mathsf{G}(E^\sharp \cup E, \mathsf{DP}_E(R), E, R, S)$.

6.2 Estimating the $A{\vee}C$-dependency graph

As in standard rewriting, the $A{\vee}C$-dependency graph of an $A{\vee}C$-rewrite theory is in general *not* computable. So, we need to use some approximation of it. For any term $t \in \mathcal{T}(\Sigma, \mathcal{X})$ let $\textsc{Cap}(t)$ result from replacing all proper subterms rooted by a defined symbol by fresh variables and let $\textsc{Ren}(t)$ which *independently* renames all *occurrences* of variables in t by using new fresh variables [1].

As usual, we do not have to talk about *mgu* when dealing with rewriting modulo equations. Instead, it is used the notion of complete set of E-unifiers. However, although in theory, all these unifiers have to be considered, for our results of reachability it is enough to check the existence of one.

Proposition 6. *Let $\mathcal{R} = (\Sigma, E, R)$ be an $A{\vee}C$-rewrite theory with $\Sigma = \mathcal{C} \uplus \mathcal{D}$. Let $u, t \in \mathcal{T}(\Sigma, \mathcal{X})$ be such that $\mathcal{V}ar(u) \cap \mathcal{V}ar(t) = \varnothing$ and θ, θ' be substitutions. If $\theta(t) \to^*_{\mathcal{E}xt_E(R),E} \circ \sim_E \theta'(u)$, then* $\textsc{Ren}(\textsc{Cap}(t))$ *and u E-unify.*

Now, we are ready to provide a correct estimation of our graph of pairs. Correctness of our definition relies on Proposition 6.

Definition 11 (Estimated $A{\vee}C$-Graph of Pairs). *Let $\mathcal{R} = (\Sigma, E, R)$ and $\mathcal{P} = (\Gamma, F, P)$ be rewrite theories and $\mathcal{S} = (\mathcal{F}, S)$ be a TRS. The* estimated $A{\vee}C$-graph *associated to them (denoted $\mathsf{EG}(F, P, E, R, S)$) has P as the set of nodes and arcs which connect them as follows:*

1. *If v unifies with s for some $s = t \in F$ or $t = s \in F$ such that $s = f(s_1, s_2)$ and $s_1 \notin \mathcal{X}$ or $s_2 \notin \mathcal{X}$, then, there is an arc from $u \to v \in P$ to $u' \to v' \in P$ if $root(u') = f$.*
2. *Otherwise, there is an arc from $u \to v \in P$ to $u' \to v' \in P$ if* $\textsc{Ren}(\textsc{Cap}(v))$ *and u' E-unify.*

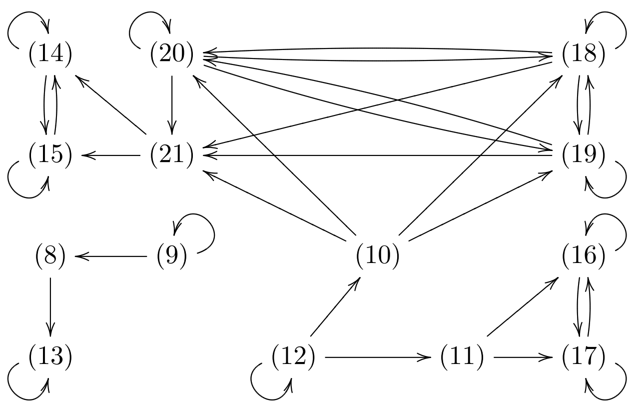
According to Definition 9, we would have the corresponding one for the *estimated* $A{\vee}C$-DG: $\mathsf{EDG}(\mathcal{R}) = \mathsf{EG}(E^\sharp \cup E, \mathsf{DP}_E(R), E, R, S)$, where

$$S = \{f^\sharp(f(x, y), z) \to f^\sharp(x, y), f^\sharp(x, f(y, z)) \to f^\sharp(y, z) \mid f \in \Sigma_A \cup \Sigma_{AC}\}.$$

Example 8. For the $A\vee C$-rewrite theory in Figure 1, the set $\mathsf{DP}_E(R)$ is[3]:

$$LIST2SET(cons(N,L)) \to UNION(N, list2set(L)) \tag{8}$$
$$LIST2SET(cons(N,L)) \to LIST2SET(L) \tag{9}$$
$$IN(N, union(M,S)) \to EQ(N,M) \tag{10}$$
$$IN(N, union(M,S)) \to OR(eq(N,M), in(N,S)) \tag{11}$$
$$IN(N, union(M,S)) \to IN(N,S) \tag{12}$$
$$UNION(union(N,N),Z) \to UNION(N,Z) \tag{13}$$
$$AND(and(true,B),Z) \to AND(B,Z) \tag{14}$$
$$AND(and(false,B),Z) \to AND(false,Z) \tag{15}$$
$$OR(or(true,B),Z) \to OR(true,Z) \tag{16}$$
$$OR(or(false,B),Z) \to OR(B,Z) \tag{17}$$
$$EQ(s(N),s(M)) \to EQ(N,M) \tag{18}$$
$$EQ(cons(N,L), cons(M,L')) \to EQ(N,M) \tag{19}$$
$$EQ(cons(N,L), cons(M,L')) \to EQ(L,L') \tag{20}$$
$$EQ(cons(N,L), cons(M,L')) \to AND(eq(N,M), eq(L,L')) \tag{21}$$

The (estimated) $A\vee C$-DG is:



By Theorem 6 we transform the $A\vee C$ problem $(E\cup E^\sharp, \mathsf{DP}(R), E, R, S)$ into a set $\mathsf{Proc}_{SCC}(E\cup E^\sharp, \mathsf{DP}(R), E, R, S)$ given by

$$\{(E\cup E^\sharp, \{(9)\}, E, R, \varnothing), (E\cup E^\sharp, \{(12)\}, E, R, \varnothing), (E\cup E^\sharp, \{(13)\}, E, R, S_{union}),$$
$$(E\cup E^\sharp, \{(14),(15)\}, E, R, S_{and}), (E\cup E^\sharp, \{(16),(17)\}, E, R, S_{or}), (E\cup E^\sharp, \{(18),(19),(20)\}, E, R, \varnothing)\}$$

which contains six new (but simpler) $A\vee C$ problems.

6.3 Use of reduction pairs

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

[3] We have introduced new 'prefix' symbols *eq*, *cons* and *union* instead of the original 'infix' ones `_==_`, `_;_`, `__`.

In the dependency pair framework reduction pairs are used to obtain *smaller* sets of pairs $\mathcal{P}' \subseteq \mathcal{P}$ by removing the *strict* pairs, i.e., those pairs $u \to v \in \mathcal{P}$ such that $u \sqsupset v$. Stability is required both for $\gtrsim$ and $\sqsupset$ because, although we only check the left- and right-hand sides of the rewrite rules $l \to r$ (with $\gtrsim$) and pairs $u \to v$ (with $\gtrsim$ or $\sqsupset$), the chains of pairs involve *instances* $\sigma(l)$, $\sigma(r)$, $\sigma(u)$, and $\sigma(v)$ of rules and pairs and we aim at concluding $\sigma(l) \gtrsim \sigma(r)$, and $\sigma(u) \gtrsim \sigma(v)$ or $\sigma(u) \sqsupset \sigma(v)$, respectively. Monotonicity is required for $\gtrsim$ to deal with the application of rules $l \to r$ to an arbitrary depth in terms. Since the pairs are 'applied' only at the root level, no monotonicity is required for $\sqsupset$ (but, for this reason, we cannot compare the rules in $\mathcal{R}$ using $\sqsupset$). Dealing with associative-commutative axioms, we will compare them with the equivalence relation defined by the stable, reflexive, transitive, and symmetric equivalence $\sim$ induced by $\gtrsim$, i.e., $\sim = \gtrsim \cap \lesssim$, since we need to impose compatibility with the equational theories E and F. The following theorem formalizes a generic processor to remove pairs from $\mathcal{P}$ by using reduction pairs.

Theorem 7 (Reduction pair processor). *Let $\mathcal{P} = (\Gamma, F, P)$ be a rewrite theory, $\mathcal{R} = (\Sigma, E, R)$ be an $A{\vee}C$-rewrite theory, and $\mathcal{S} = (\mathcal{F}, S)$ be a TRS. Let $(\gtrsim, \sqsupset)$ be a reduction pair such that*

1. $R \subseteq \gtrsim$,
2. $P \cup S \subseteq \gtrsim \cup \sqsupset$, *and*
3. $E \cup F \subseteq \sim$.

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

$$\mathsf{Proc}_{RP}(F, P, E, R, S) = \begin{cases} \{(F, P - P_{\sqsupset}, E, R, S)\} & \textit{if (1), (2), and (3) hold} \\ \{(F, P, E, R, S)\} & \textit{otherwise} \end{cases}$$

is sound and complete.

7 Related work and conclusions

As remarked in the introduction, this is not the first work which tries to use dependency pairs for proving termination of rewriting modulo an equational theory, see [5, 10, 11, 9, 13, 14]. Our work, however, is, as far as the authors know, the first one which provides a correct notion of minimal non-terminating term for an $A{\vee}C$-rewrite theory $\mathcal{R} = (\Sigma, E, R)$ which can be used to provide a suitable definition of minimal chain of dependency pairs which can be used to characterize $A{\vee}C$-termination (Theorem 4). In order to substantiate this claim, consider the AC-rewrite theory $\mathcal{R} = (\Sigma, E, R)$ in Example 2 again. The $A{\vee}C$-DPs for $\mathcal{R}$ are enumerated in Example 7. Such dependency pairs coincide with the ones which would be computed by, e.g., [5, 10, 11]. Remember that t in Example 2 is *minimal* in Giesl and Kapur's sense (Definition 2). We should, then, be able to find an infinite *minimal* chain of DPs starting from $t^\sharp$. According to [5, 10, 11], 'minimal' means that $\sigma(v_i)$ is $(\mathcal{E}xt_E(R), E)$-terminating for all pairs $u_i \to v_i \in \mathsf{DP}_E(R)$ in the chain of dependency pairs induced by the substitution σ. However, this is *not* possible: the marked version $t^\sharp$ of t is $F(f(0,1), f(0, f(1,2)))$, which is an $(\mathcal{E}xt_E(R), E)$-terminating term. After some $E^\sharp \cup E$-equivalence steps we would be able to apply one of the rules in $\mathsf{DP}_E(R)$. Note, however, that *no rule* $u \to v \in \mathsf{DP}_E(R)$ except (5) has a right-hand side v which can be rewritten (after instantiation into $\sigma(v)$) into an instance $\sigma(u')$ of the left-hand side u' of any other pair in $\mathsf{DP}_E(R)$ by means of $(\mathcal{E}xt_E(R), E^\sharp \cup E)$-rewriting steps. This means that only the dependency pair (5) could be used in any infinite minimal chain of dependency pairs starting from $t^\sharp$. But such a chain would start as follows:

$$F(f(0,1), f(0, f(1,2))) \sim_{E^\sharp \cup E} F(f(0,0), f(1, f(1,2))) \to_{(5)} F(f(0, f(1,2)), f(1, f(1,2)))$$

where $F(f(0, f(1,2)), f(1, f(1,2)))$ contains a subterm $f(1, f(1,2))$ which, as showed in Example 2, is non-$(\mathcal{E}xt_E(R), E)$-terminating. Therefore, this chain of dependency pairs is *not* minimal. We conclude that, according to the notion of minimal chain in the aforementioned papers, *there is no minimal chain of pairs starting from* $t^\sharp$. This means that no *sound* approach to proving AC-termination on the basis of such notion of minimal chain is possible. In this paper we have introduced the notion of *stably minimal term* (Definition 3) which overcomes these problems (Proposition 4 and Theorem 3) and leads to an appropriate characterization of $A\vee C$-termination as the absence of infinite minimal chains of $A\vee C$-DPs (Definitions 5 and 6, and Theorem 4).

Furthermore, we note that [10, 11] deal with *AC-rewrite theories* only, and that [5], which considers more general rewrite theories E including $A\vee C$-theories do not cover our work in a second respect: when purely associative theories are considered (i.e., rewrite theories $\mathcal{R} = (\Sigma, E, R)$ such that $E_f \subseteq \{A_f\}$ for all $f \in \Sigma$), then Giesl and Kapur's technique requires the computation of *instances* of the rules in $\mathcal{E}xt_E(R)$ for which the computation of *all* the E-unifiers $uni_E(v,l)$ of v and l for the rules $l \to r$ in $\mathcal{E}xt_E(R)$ and equations $u = v \in E$ or $v = u \in E$ is required. It is well-known, however, that the E-unification problem for associative theories E is *infinitary*, which means that $uni_E(v,l)$ is not guaranteed to be finite, in general. In sharp contrast, we do not have to do that for dealing with purely associative rewrite theories $\mathcal{R}$.

Our second main (and novel) contribution is the formalization of an $A\vee C$-dependency pair framework (Definitions 7 and 8) which, on the basis of the previously developed theory, can be used to develop automatic tools for proving termination of $A\vee C$-rewrite theories (Theorem 5). Two important processors have been adapted as well: the SCC processor (Theorem 6) and the reduction pair processor (Theorem 7).

Much work remains ahead both in terms of further developing the new $A\vee C$-dependency pair framework and in tool support. Appropriate reduction orderings which are well-suited for being used in the reduction pair processor should be investigated. It would also be very useful to explore how the requirements on E can be relaxed to handle even more general sets of axioms. Regarding tool support for the method we have presented, we plan to integrate it within the tool MU-TERM [2]. In this way, our termination technique modulo combinations of associative and commutative axioms will become applicable to an even wider range of rewrite theories, that can be transformed into $A \vee C$-theories by non-termination-preserving transformations [3, 4, 12].

References

1. Arts, T., Giesl, J.: Termination of Term Rewriting Using Dependency Pairs. Theoretical Computer Science. 236(1–2), 133–178 (2000)
2. Alarcón, B., Gutiérrez, R., Iborra, J., Lucas, S.: Proving Termination of Context-Sensitive Rewriting with MU-TERM. Electronic Notes in Theoretical Computer Science. 188, 105–115 (2007)
3. Durán, F., Lucas, S., Meseguer, J.: Termination Modulo Combinations of Equational Theories. In: Ghilardi, S., Sebastiani, R. (eds.). 7th International Symposium on Frontiers of Combining Systems, FroCoS'09. LNAI, vol. 5749, pp. 246-262. Springer-Verlag (2009)
4. Durán, F., Lucas, S., Marché, C., Meseguer, J., Urbain, X.: Proving Operational Termination of Membership Equational Programs. Higher-Order and Symbolic Computation. 21(1-2), 59–88 (2008)
5. Giesl, J., Kapur, D.: Dependency Pairs for Equational Rewriting. In: Middeldorp, A. (ed.) 12th International Conference of Rewriting Techniques and Applications, RTA'01. LNCS, vol. 2051, pp. 93-108. Springer (2001)
6. Giesl, J., Thiemann, R., Schneider-Kamp, P., Falke, S.: Mechanizing and Improving Dependency Pairs. Journal of Automatic Reasoning, 37(3), 155–203 (2006)
7. Hirokawa, N., Middeldorp, A.: Dependency Pairs Revisited. In: van Oostrom, V. (ed) 15th International Conference on Rewriting Techniques and Applications, RTA'04. LNCS, vol. 3091, pp. 249–268. Springer (2004)
8. Hirokawa, N., Middeldorp, A.: Automating the Dependency Pair Method. Information and Computation. 199, 172–199 (2005)
9. Kusakari, K.: Termination, AC-Termination and Dependency Pairs of Term Rewriting Systems. PhD. Thesis, School of Information Science, JAIST (2000)
10. Kusakari, K., Nakamura, M., Toyama, Y.: Elimination Transformations for Associative-Commutative Rewriting Systems. Journal of Automated Reasoning. 37, 205-229 (2006)
11. Kusakari, K., Toyama, Y.: On Proving AC-Termination by AC-Dependency Pairs. IEICE Transactions on Information and Systems. E84-D, 604-612 (2001)
12. Lucas, S., Meseguer, J.: Operational Termination of Membership Equational Programs: the Order-Sorted Way. Electronic Notes in Theoretical Computer Science. 238(3), 207-225 (2009)
13. Marché, C., Urbain, X.: Termination of associative-commutative rewriting by dependency pairs. In: Nipkow, T. (ed.) 9th International Conference on Rewriting Techniques and Applications, RTA'98. LNCS, vol. 1379, pp. 241-255. Springer (1998)
14. Marché, C., Urbain, X.: Modular and incremental proofs of AC-termination. Journal of Symbolic Computation. 38, 873-897 (2004)
15. Peterson, G.E., Stickel, M.E.: Complete Sets of Reductions for Some Equational Theories. Journal of the ACM. 28(2), 233-264 (1981)