On Proving AC-Termination by AC-Dependency Pairs

KUSAKARI Keiichirou, TOYAMA Yoshihito

Research Institute of Electrical Communication, Tohoku University
e-mail: {kusakari,toyama}@nue.riec.tohoku.ac.jp

Abstract. Arts and Giesl introduced the notion of dependency pairs, which gives effective methods for proving termination of term rewriting systems (TRSs). In this paper, we extend the notion to AC-TRSs, and introduce new methods for effectively proving AC-termination. It is impossible to directly apply the notion of dependency pairs to AC-TRSs. To avoid this difficulty, we introduce the notion of extended dependency pairs. Finally we define the AC-dependency pair and the AC-dependency chain. Furthermore, we propose approximated AC-dependency graphs, which is very useful for proving AC-termination in practice, using the approximation technique based on Ω-reduction and Ω_V-reduction.

Keyword. AC-term rewriting system, AC-termination, AC-dependency pair, AC-dependency graph

1 Introduction

Term rewriting systems (TRSs) have been widely applied to automated theorem proving, program optimization and program verification [3, 6, 8]. Term rewriting systems with associative and commutative law (AC-TRSs) can be regarded as a model for computation in which terms are reduced using a set of directed equations with AC-equations. AC-TRSs themselves can be regarded as functional programs. They can be used to represent abstract interpreters of functional programs with AC-functions such as the addition and the multiplication. For example, the representation

$$Node(Leaf(1), Node(Leaf(1), Leaf(2)))$$

for a binary tree naturally plays as the multiset $\{1, 1, 2\}$ by interpreting $Node$ as an AC-function symbol, which denotes the union over multisets. Functions over multisets can be easily defined with this representation. A membership function *member* is non-recursively defined as follows:

$$\begin{cases} member(x, Leaf(x)) & \rightarrow \quad True \\ member(x, Node(Leaf(x), y)) & \rightarrow \quad True \end{cases}$$

The AC-termination property is one of the most fundamental properties of AC-TRSs. To prove AC-termination, we commonly design an AC-reduction order by which all rules are ordered. Several AC-reduction orders have been proposed [4, 13]. The order in [4] is based on the polynomial interpretation, and the order in [13] is based on the recursive path order [5] with flattening. However, in order to ensure the AC-compatibility, they have many limitations for AC-TRSs.

On TRSs, Arts and Giesl introduced the notion of dependency pairs [1, 2], which can offer effective methods for analyzing an infinite reduction sequence. Using dependency pairs, we can easily show the termination property of TRSs to which traditional techniques cannot be applied directly. Furthermore, to strengthen the dependency pair method for proving termination, Arts and Giesl introduced the notion of dependency graphs [1, 2].

First, we extend the notion of dependency pairs to AC-TRSs. It is impossible to directly apply the notion of dependency pairs to AC-TRSs. We express this difficulty through an example. To avoid this difficulty we introduce the head parts in terms and show an analogy between the root positions in infinite reduction sequences by TRSs and the head positions in those of AC-TRSs. Indeed, this analogy is essential for the extension of dependency pairs to AC-TRSs. Based on this analogy, we propose AC-dependency pairs and AC-dependency chains, and show that existing no infinite AC-dependency chain is a necessary and sufficient condition of AC-termination. Using this relation, we introduce a new method for effectively proving AC-termination. Next, we introduce the notion of AC-dependency graphs and propose an algorithm, which improves the method by Arts and Giesl [1, 2], for generating an approximated AC-dependency graph. Our algorithm can be also applied to TRSs, because TRSs are AC-TRSs without AC-symbols.

The rest of the paper is organized as follows. The next section gives preliminary definitions. In section 3, we explain the notion of dependency pairs and extend to AC-TRSs. In section 4, we introduce AC-dependency graphs on AC-TRSs and design an algorithm for generating an approximated AC-dependency graph. In section 5, we compare our AC-dependency pair and another AC-dependency pair, which was introduced by Marché and Urbain [10].

2 Preliminaries

We assume that the reader is familiar with notions of term rewriting systems [3].

Let Σ be a finite set of function symbols, and $\mathcal{V}$ an enumerable set of variables with $\Sigma \cap \mathcal{V} = \emptyset$. The set of terms constructed from Σ and $\mathcal{V}$ is written by $\mathcal{T}(\Sigma, \mathcal{V})$. Identity of terms is denoted by $\equiv$. The size of a term t is denoted by $|t|$. The set of variables occurring in a term t is denoted by $Var(t)$. The set Σ_{AC} of AC-function symbols, which have fixed arity 2, is a subset of Σ. The binary relation $\underset{AC}{\sim}$ is the congruence relation generated by $f(f(x,y),z) =_A f(x, f(y,z))$ and $f(x,y) =_C f(y,x)$ for all $f \in \Sigma_{AC}$.

A substitution $\theta : \mathcal{V} \to \mathcal{T}(\Sigma, \mathcal{V})$ is a mapping. A substitution over terms is defined as a homomorphic extension. We write $t\theta$ instead of $\theta(t)$. A position of a term is a sequence of positive integers. The root position is denoted by the empty sequence ε. The prefix order $\prec$ on term positions is defined by $p \prec q$ iff $pw = q$ for some w $(\neq \varepsilon)$. $(t)_p$ denotes the symbol at position p in t, and $t|_p$ denotes the subterm of t at position p. A context C is a term with occurrences of a special constant $\square$, called a hole. $C[t_1, \ldots, t_n]$ denotes the result of placing $t_1, \ldots, t_n$ in the n holes of C from left to right. In particular $C[\ ]$ denotes a context containing precisely one hole and we sometimes write $C[t]_p$ to indicate at which position the replacement takes place. A term s is called a subterm of t if $t \equiv C[s]$ for some context C. A subterm s of t is called a proper subterm if $s \not\equiv t$.

A rewrite rule is a pair of terms, written by $l \to r$, with $l \notin \mathcal{V}$ and $Var(l) \supseteq Var(r)$. An AC term rewriting system (AC-TRS) is a finite set of rules. The set of defined symbols in R is $DF(R) = \{(l)_\varepsilon \mid l \to r \in R\}$. A reduction relation $\underset{R}{\to}$ and an AC-reduction relation $\underset{R\backslash AC}{\to}$ are defined as follows:

$$
\begin{aligned}
s \underset{R}{\to} t \quad &\overset{\text{def}}{\Longleftrightarrow} \quad \exists\, l \to r \in R, \exists C[\ ], \exists \theta, \\
&\qquad\quad s \equiv C[l\theta] \land t \equiv C[r\theta] \\
s \underset{R\backslash AC}{\to} t \quad &\overset{\text{def}}{\Longleftrightarrow} \quad \exists\, l \to r \in R, \exists C[\ ], \exists \theta, \\
&\qquad\quad s \underset{AC}{\sim} C[l\theta] \land t \equiv C[r\theta]
\end{aligned}
$$

Note that $\underset{R\backslash AC}{\to} = \underset{AC}{\sim} \cdot \underset{R}{\to}$. When we want to specify the position p of $C[l\theta]_p$ in the above reductions, we write $s \underset{R}{\to}^p t$ and $s \underset{R\backslash AC}{\to}^p t$. We often omit the subscripts R and $R\backslash AC$ whenever no confusion arises. The transitive-reflexive closure of a binary relation $\to$ is denoted by $\overset{*}{\to}$. The transitive closure of a binary relation $\to$ is denoted by $\overset{+}{\to}$. R is terminating if there is no infinite sequence such that $t_0 \underset{R}{\to} t_1 \underset{R}{\to} t_2 \underset{R}{\to} \cdots$, and AC-terminating if there is no infinite sequence such that $t_0 \underset{R\backslash AC}{\to} t_1 \underset{R\backslash AC}{\to} t_2 \underset{R\backslash AC}{\to} \cdots$.

A binary relation $>$ is a strict order if $>$ is transitive and irreflexive. A binary relation $\geq$ is a partial order if $\geq$ is reflexive, transitive and antisymmetric. A binary relation $\gtrsim$ is a quasi-order if $\gtrsim$ is transitive and reflexive. The extension of a strict order $>$, denoted $\geq$, is defined as $> \cup \equiv$. The strict part of a quasi-order $\gtrsim$, denoted $>$, is defined as $\gtrsim \setminus \lesssim$. A reduction order $>$ is a well-founded strict order that is monotonic and stable. Here $>$ is monotonic iff $s > t \Rightarrow C[s] > C[t]$ and stable iff $s > t \Rightarrow s\theta > t\theta$. An AC-reduction order $>$ is a reduction order that is AC-compatible, i.e., $s \underset{AC}{\sim} s' > t \Rightarrow s > t$.

Proposition 2.1 *[3, 7, 8] A TRS R is terminating iff there is a reduction order $>$ that satisfies $l > r$ for all $l \to r \in R$. An AC-TRS R is AC-terminating iff there is an AC-reduction order $>$ that satisfies $l > r$ for all $l \to r \in R$.*

3 AC-Dependency Pairs and Chains

This section presents the notions of AC-dependency pairs and AC-dependency chains for AC-TRSs. We first explain the notions of dependency pairs and dependency chains for TRSs [1, 2]. We show why the notion of dependency pairs for TRSs cannot be directly applied to AC-TRSs through an example. To avoid this difficulty, we introduce the head and the body parts in terms and show an analogy between the root positions in infinite reduction sequences by TRSs and the head positions in those of AC-TRSs. Based on this analogy, we define AC-dependency pairs and AC-dependency chains. To achieve better readability, we first introduce a simpler notion of AC-dependency pairs without marked symbols, and explain some basic ideas of techniques involved in section 3.2. We next extend it with marked symbols in section 3.3.

3.1 Dependency Pairs and Dependency Chains

Following Arts and Giesl [1, 2], we explain dependency pairs and dependency chains.

Definition 3.1 *Let $\{f^\# \mid f \in \Sigma\}$ be the set of marked function symbols disjoint from $\Sigma \cup \mathcal{V}$. We define the marked term $t^\#$ by $t^\# \equiv (f(t_1, \cdots, t_n))^\# \equiv f^\#(t_1, \cdots, t_n)$, and $\Sigma^\# = \Sigma \cup \{f^\# \mid f \in \Sigma\}$.*

Definition 3.2 *Let R be a TRS. A pair $\langle u^\#, v^\# \rangle$ of terms is a dependency pair of R if there is a rule $l \to r \in R$ such that $u \equiv l$, $v \equiv r|_p$ and $(r)_p \in DF(R)$. The set of dependency pairs of R is written by $DP^\#(R)$.*

Definition 3.3 *A sequence of dependency pairs $\langle u_0^\#, v_0^\# \rangle \langle u_1^\#, v_1^\# \rangle \langle u_2^\#, v_2^\# \rangle \cdots$ is a dependency chain if there is a substitution θ such that $v_i^\# \theta \overset{*}{\underset{R}{\to}} u_{i+1}^\# \theta$ for all i. We assume that different dependency pair occurrences do not own jointly variables without loss of generality.*

Proposition 3.4 *[1, 2] A TRS R is terminating iff there exists no infinite dependency chain of R.*

Definition 3.5 *A quasi-order $\gtrsim$ is a weak reduction order if $\gtrsim$ is stable and monotonic and its strict part $>$ is stable and well-founded.*

While the extension $\geq$ of a reduction order $>$ is a weak reduction order, the strict part $>$ of a weak reduction order $\gtrsim$ is not a reduction order in general. The reason is that a weak reduction order does not guarantee the monotonicity of its strict part.

Proposition 3.6 *[1, 2] Let R be a TRS. If there is a weak reduction order $\gtrsim$ such that*

- *$l \gtrsim r$ for all $l \to r \in R$,*
- *$u^\# > v^\#$ for all $\langle u^\#, v^\# \rangle \in DP^\#(R)$,*

then R is terminating.

3.2 Unmarked AC-Dependency Pairs and Chains

The notion of original dependency pairs cannot be directly applied to AC-TRSs as follows:

Example 3.7 *Consider the AC-TRS $R = \{f(x, x) \to f(0, 1)\}$ with $\Sigma_{AC} = \{f\}$. Here, $DP^\#(R) = \{\langle f^\#(x, x), f^\#(0, 1)\rangle\}$. The AC-TRS R is not AC-terminating. For example,*

$$\begin{array}{rcl} f(\underline{f(0,0)}, 1) & \underset{R}{\to} & f(f(0,1), 1) \\ & \underset{AC}{\sim} & f(0, \underline{f(1,1)}) \\ & \underset{R}{\to} & f(0, f(0,1)) \\ & \underset{AC}{\sim} & f(f(0,0), 1). \end{array}$$

However, there is not an infinite dependency chain, i.e., $f^\#(0,1)\theta \overset{*}{\underset{R \setminus AC}{\not\to}} f^\#(x,x)\theta$ for all θ.

The above example indicates that proposition 3.4 does not work for AC-TRSs. Proposition 3.4 is proved from the fact that any infinite reduction sequence from a minimal size counterexample must include a reduction at the root position [1, 2]. However, this property does not hold for AC-TRSs. In fact, the infinite AC-reduction sequence in the above example does not include such the reduction though $f(f(0,0), 1)$ is a minimal size counterexample in R. To avoid this difficulty, we introduce the notion of head positions, which behaves like the root position in a minimal size counterexample in TRSs.

Definition 3.8 *We introduce head positions and body terms as follows:*

- $(t)_\varepsilon \in \Sigma_{AC}$

$$\begin{array}{rcl} \mathcal{O}_{head}(t) & = & \{p \mid \forall q \preceq p,\ (t)_q = (t)_\varepsilon\} \\ T_{body}(t) & = & \{t|_p \mid (t)_p \neq (t)_\varepsilon,\ \forall q \prec p\ [(t)_q = (t)_\varepsilon]\} \end{array}$$

- $(t)_\varepsilon \notin \Sigma_{AC}$

$$\begin{array}{rcl} \mathcal{O}_{head}(t) & = & \{\varepsilon\} \\ T_{body}(t) & = & \{t|_i \mid 1 \leq i \leq arity((t)_\varepsilon)\} \end{array}$$

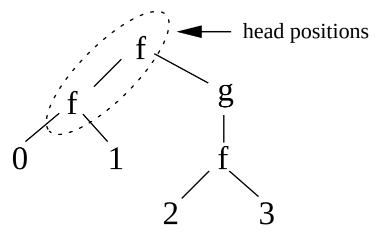

Figure 1: Head positions

For example, let $t \equiv f(f(0,1), g(f(2,3)))$ and $\Sigma_{AC} = \{f\}$. Then $\mathcal{O}_{head}(t) = \{\varepsilon, 1\}$ and $T_{body}(t) = \{0, 1, g(f(2,3))\}$ (Fig.1).

Definition 3.9 *We introduce the head reduction $\overset{hd}{\to}$, the body reduction $\overset{bd}{\to}$ and the head subterm relation $\trianglerighteq_{hd}$ as follows (Fig.2):*

$$s \overset{hd}{\to} t \quad \overset{\text{def}}{\iff} \quad s \underset{AC}{\sim} s' \underset{R}{\to}^{p} t$$

$$
\begin{array}{rcl}
 & & \textit{for some } s' \textit{ and } p \in \mathcal{O}_{head}(s') \\
s \overset{bd}{\rightarrow} t & \overset{\text{def}}{\Longleftrightarrow} & s \underset{AC}{\sim} s' \underset{R}{\rightarrow}^{p} t \\
 & & \textit{for some } s' \textit{ and } p \notin \mathcal{O}_{head}(s') \\
s \trianglerighteq_{hd} t & \overset{\text{def}}{\Longleftrightarrow} & s \underset{AC}{\sim} C[t]_p \\
 & & \textit{for some } C[\ ] \textit{ and } p \in \mathcal{O}_{head}(C[t])
\end{array}
$$

For example, let $R = \{f(x,y) \rightarrow a\}$ with $\Sigma_{AC} = \{f\}$. Then

$$
\begin{array}{lcl}
f(\underline{f(0,1)}, g(f(2,3))) & \overset{hd}{\rightarrow} & f(a, g(f(2,3))), \\
f(f(0,1), g(\underline{f(2,3)})) & \overset{bd}{\rightarrow} & f(f(0,1), g(a)), \text{ and} \\
f(\underline{f(0,1)}, g(f(2,3))) & \trianglerighteq_{hd} & f(0,1).
\end{array}
$$

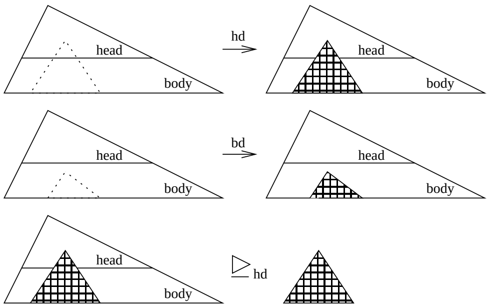

Figure 2: Relations in Definition 3.9

Definition 3.10 *Let R be an AC-TRS. A pair $\langle u, v\rangle$ of terms is an unmarked dependency pair of R if there is a rule $l \rightarrow r \in R$ such that $u \equiv l$, $v \equiv r|_p$ and $(r)_p \in DF(R)$. A pair $\langle u, v\rangle$ of terms is an unmarked extended dependency pair of R if there is a rule $l \rightarrow r \in R$ such that $(l)_\varepsilon = f \in \Sigma_{AC}$, $u \equiv f(l,z)$ and $v \equiv f(r,z)$ where z is a fresh variable. A pair $\langle u, v\rangle$ of terms is an unmarked AC-dependency pair of R if it is either an unmarked dependency pair or an unmarked extended one. We denote by $DP_{AC}(R)$ all unmarked AC-dependency pairs of R.*

Example 3.11 *Let $\Sigma_{AC} = \{add\}$ and*

$$
R = \left\{ \begin{array}{rcl} add(x,0) & \rightarrow & x \\ add(x,s(y)) & \rightarrow & s(add(x,y)) \end{array} \right.
$$

Here, $DF(R) = \{add\}$. There is one unmarked dependency pair

$$\langle add(x,s(y)), add(x,y)\rangle$$

and two unmarked extended dependency pairs

$$
\begin{array}{c}
\langle add(add(x,0),z), add(x,z)\rangle, \\
\langle add(add(x,s(y)),z), add(s(add(x,y)),z)\rangle.
\end{array}
$$

Definition 3.12 *A sequence of unmarked AC-dependency pairs $\langle u_0, v_0\rangle\langle u_1, v_1\rangle\langle u_2, v_2\rangle \cdots$ is an unmarked AC-dependency chain if there is a substitution θ such that $v_i\theta \overset{bd}{\rightarrow}{}^{*} \trianglerighteq_{hd} u_{i+1}\theta$ for all i. We assume that different unmarked AC-dependency pair occurrences do not own jointly variables without loss of generality.*

Lemma 3.13 *Let $s \equiv C[s_1, \cdots, s_n] \overset{bd}{\rightarrow} t$ such that each $s' \in T_{body}(s)$ is a subterm of some s_i. Then, there are $t_1, \cdots, t_n$ such that $C[t_1, \cdots, t_n] \underset{AC}{\sim} t$, each $t' \in T_{body}(C[t_1, \cdots, t_n])$ is a subterm of some t_i, and there is some k such that $s_k \underset{R\setminus AC}{\rightarrow} t_k$ and $s_j \equiv t_j$ for all $j \neq k$.*

proof: Follows from the definitions of $\overset{bd}{\rightarrow}$ and $\underset{AC}{\sim}$. □

Lemma 3.14 *Let t be an arbitrary term such that any term in $T_{body}(t)$ is AC-terminating. Then, there is no infinite $\overset{bd}{\rightarrow}$ sequence starting from t.*

proof: Assume that there is an infinite $\overset{bd}{\rightarrow}$ sequence $t \equiv t_0 \overset{bd}{\rightarrow} t_1 \overset{bd}{\rightarrow} \cdots$. Let $t_0 \equiv C[t_{01}, \cdots, t_{0n}]$ where all body terms of t_0 are presented by $t_{01}, \cdots, t_{0n}$. Applying lemma 3.13 repeatedly, for any i there are $t_{i1}, \cdots, t_{in}$ such that $t_i \underset{AC}{\sim} C[t_{i1}, \cdots, t_{in}]$, each term of $T_{body}(t_i)$ is a subterm of some t_{ij}, and there is a k_i such that $s_{i-1,k_i} \underset{R\setminus AC}{\rightarrow} t_{ik_i}$ and $s_{i-1,j} \equiv t_{ij}$ for all $j \neq k_i$. Thus, there is a t_{0j} that is not AC-terminating. It is a contradiction to AC-termination of $t_{0j} \in T_{body}(t)$. □

The above lemma means that "any infinite AC-reduction sequence from a minimal size counterexample must include a reduction at the head position". This property corresponds to the property "any infinite reduction sequence from a minimal size counterexample must include a reduction at the root position" in TRSs. Indeed, this analogy is essential in our extension of dependency pairs to AC-TRSs.

Lemma 3.15 *If each t_i is AC-terminating and $g(t_1, \cdots, t_n)$ is not AC-terminating, then $g \in DF(R)$.*

proof: From lemma 3.14, there exists a head reduction $\overset{hd}{\rightarrow}$ in any infinite AC-reduction sequence from $g(t_1, \cdots, t_n)$. Therefore, g is a defined symbol. □

Lemma 3.16 *Let t be a term such that t is not AC-terminating and each element of $T_{body}(t)$ is AC-terminating. Then, there exist t' and s such that $t \overset{bd}{\rightarrow}{}^{*} \trianglerighteq_{hd} t' \underset{R}{\overset{hd}{\rightarrow}} s$, each proper subterm of t' is AC-terminating, and s is not AC-terminating.*

proof: Let $T_i = \{t' \mid t \overset{bd}{\to}{}^{i} \trianglerighteq_{hd} t' \overset{hd}{\underset{R}{\to}} s$ and s is not AC-terminating, for some $s\}$, where $\overset{bd}{\to}{}^{i}$ denotes a $\overset{bd}{\to}$ reduction of i steps. Since t is not AC-terminating and each element of $T_{body}(t)$ is AC-terminating, there is an n such that $T_n \neq \emptyset$ and $T_j = \emptyset$ for all $j > n$ from lemma 3.14. Let t' be a minimal size element in T_n. We assume that t' has a proper subterm t'' which is not AC-terminating. Since each element of $T_{body}(t)$ is AC-terminating, so is each element of $T_{body}(t')$. Thus, each element of $T_{body}(t'')$ is AC-terminating and $t' \triangleright_{hd} t''$. From lemma 3.14, there is an infinite AC-reduction sequence such that $t'' \overset{bd}{\to}{}^{k} \trianglerighteq_{hd} \hat{t} \overset{hd}{\underset{R}{\to}} \hat{s} \underset{R\setminus AC}{\to} \cdots$ for some $\hat{t}$ and $\hat{s}$. Thus, $\hat{t} \in T_{n+k}$. From the maximality of n, $T_{n+k} = T_n$. Thus, $t'' \trianglerighteq_{hd} \hat{t}$. From the definition of $\trianglerighteq_{hd}$, we have $|t'| > |t''| \geq |\hat{t}|$. It is a contradiction to the minimality of t'. □

Theorem 3.17 *An AC-TRS R is AC-terminating iff there exists no infinite unmarked AC-dependency chain of R.*

proof: ($\Rightarrow$) Let $\langle u_0, v_0\rangle\langle u_1, v_1\rangle\langle u_2, v_2\rangle \cdots$ be an infinite unmarked AC-dependency chain with a substitution θ such that $v_i\theta \overset{bd}{\to}{}^{*} \trianglerighteq_{hd} u_{i+1}\theta$ for all i. Thus, $v_i\theta \overset{*}{\underset{R\setminus AC}{\to}} \underset{AC}{\sim} C'_{i+1}[u_{i+1}\theta] \underset{R}{\to} C_{i+1}[v_{i+1}\theta]$ for all i. Therefore, there is an infinite AC-reduction sequence $v_0\theta \overset{+}{\underset{R\setminus AC}{\to}} C_1[v_1\theta] \overset{+}{\underset{R\setminus AC}{\to}} C_1[C_2[v_2\theta]] \overset{+}{\underset{R\setminus AC}{\to}} \cdots$.
($\Leftarrow$) Let t_0 be a minimal size counterexample, i.e., each proper subterm of t_0 is AC-terminating and t_0 is not AC-terminating. From lemma 3.16, we have $t_0 \overset{bd}{\to}{}^{*} \trianglerighteq_{hd} C[l\theta] \overset{hd}{\underset{R}{\to}} C[r\theta]$ for some C, $l \to r$ and θ such that each proper subterm of $C[l\theta]$ is AC-terminating and $C[r\theta]$ is not AC-terminating. We show that there exists a term t_1 and an unmarked AC-dependency pair $\langle u_0, v_0\rangle$ such that $t_0 \overset{bd}{\to}{}^{*} \trianglerighteq_{hd} u_0\theta_0$, $v_0\theta_0 \underset{AC}{\sim} t_1$, t_1 is not AC-terminating and each proper subterm of t_1 is AC-terminating. We have two following cases:

(a) $C \equiv \Box$.
Let t_1 be a minimal size counterexample in $r\theta$. Since each proper subterm of $l\theta$ is AC-terminating, $x\theta$ is AC-terminating for all $x \in Var(r) \subseteq Var(l)$. Thus, $t_1 \equiv v_0\theta$ for some non-variable subterm v_0 of r. Therefore, there is an unmarked dependency pair $\langle u_0, v_0\rangle$ and substitution θ_0 such that $u_0\theta_0 \equiv l\theta$ and $v_0\theta_0 \equiv t_1$ from the minimality of t_1 and lemma 3.15.

(b) $C \not\equiv \Box$.
Since each proper subterm of $C[l\theta]$ is AC-terminating, so is each proper subterm of $C[r\theta]$. From the definition of head parts, $(l)_\varepsilon = f \in \Sigma_{AC}$. Thus, there is an unmarked extended dependency pair $\langle u_0, v_0\rangle$ such that $u_0 \equiv f(l, z)$, $v_0 \equiv f(r, z)$ and $f(l, z)\theta_0 \underset{AC}{\sim} C[l\theta] \underset{R}{\to} C[r\theta] \underset{AC}{\sim} f(r, z)\theta_0$. We obtain $t_1 \equiv C[r\theta]$.

Repeating this procedure to $t_0, t_1, t_2, \cdots$, we get AC-dependency pairs $\langle u_0, v_0\rangle, \langle u_1, v_1\rangle, \langle u_2, v_2\rangle \cdots$ such that $v_i\theta_i \underset{AC}{\sim} t_{i+1} \overset{bd}{\to}{}^{*} \trianglerighteq_{hd} u_{i+1}\theta_{i+1}$ for all i, which makes an infinite unmarked AC-dependency chain $\langle u_0, v_0\rangle\langle u_1, v_1\rangle\langle u_2, v_2\rangle \cdots$. □

Note that the AC-TRS $R = \{f(x, x) \to f(0, 1)\}$ with $\Sigma_{AC} = \{f\}$, displayed in example 3.7, has the following infinite unmarked AC-dependency chain

$$\langle f(f(x_0, x_0), z_0), f(f(0, 1), z_0)\rangle$$
$$\langle f(f(x_1, x_1), z_1), f(f(0, 1), z_1)\rangle$$
$$\langle f(f(x_2, x_2), z_2), f(f(0, 1), z_2)\rangle \cdots$$

with the substitution θ such that $x_{2i}\theta = z_{2i+1}\theta = 0$ and $x_{2i+1}\theta = z_{2i}\theta = 1$ ($i = 0, 1, 2, \ldots$).

3.3 AC-Dependency Pairs and Chains

On the dependency pair method in TRSs, the notion of marked symbols is very useful for proving termination [1, 2]. In this subsection, we discuss the marking for AC-dependency pairs. The root position of a term in TRSs is corresponding to the head positions of a term in AC-TRSs in the behavior of a minimal size counterexample of infinite reduction sequences. Thus, we define a term $t^\#$ for AC-TRSs, which is the result of marking with # all head positions instead of the root position of t.

Definition 3.18

$$\left\{\begin{array}{rcl} x^\# & \equiv & x \\ (f(t_1, t_2))^\# & \equiv & f^\#(t_1^{\#f}, t_2^{\#f}) \\ & & \textit{if}\ \ f \in \Sigma_{AC} \\ (f(t_1, \cdots, t_n))^\# & \equiv & f^\#(t_1, \cdots, t_n) \\ & & \textit{if}\ \ f \notin \Sigma_{AC} \end{array}\right.$$

$$\left\{\begin{array}{rcl} x^{\#f} & \equiv & x \\ (f(t_1, \cdots, t_n))^{\#f} & \equiv & f^\#(t_1^{\#f}, \cdots, t_n^{\#f}) \\ (g(t_1, \cdots, t_m))^{\#f} & \equiv & g(t_1, \cdots, t_m) \quad \textit{if}\ \ f \neq g \end{array}\right.$$

We regard $f^\#$ as an AC-function symbol for all $f \in \Sigma_{AC}$, i.e., $\Sigma_{AC}^\# = \Sigma_{AC} \cup \{f^\# \mid f \in \Sigma_{AC}\}$.

For example, if $t \equiv f(f(0, 1), g(f(2, 3)))$ and $\Sigma_{AC} = \{f\}$ then $t^\# \equiv f^\#(f^\#(0, 1), g(f(2, 3)))$.

Definition 3.19 *Let R be an AC-TRS. A pair $\langle u^\#, v^\#\rangle$ is a AC-dependency pair of R if $\langle u, v\rangle$ is an unmarked AC-dependency pair of R. We denote by $DP_{AC}^\#(R)$ all AC-dependency pairs of R.*

Unlike the marking for TRSs, the marking for AC-TRSs is not compatible with $\overset{bd}{\to}$, i.e., it does not hold that $s \xrightarrow{bd} t \Rightarrow s^{\#} \rightarrow t^{\#}$. For example, consider $R = \{g(x) \rightarrow f(x,x), h(x) \rightarrow x\}$ with $\Sigma_{AC} = \{f\}$. Then, we have the follows:

$$\begin{array}{rcl} s & \equiv & f(g(0), h(f(1,1))) \\ & \xrightarrow{bd} & f(f(0,0), h(f(1,1))) \\ & \xrightarrow{bd} & f(f(0,0), f(1,1)) \\ & \equiv & t \\ s^{\#} & \equiv & f^{\#}(g(0), h(f(1,1))) \\ & \xrightarrow{bd} & f^{\#}(f(0,0), h(f(1,1))) \\ & \xrightarrow{bd} & f^{\#}(f(0,0), f(1,1)) \\ & \not\equiv & f^{\#}(f^{\#}(0,0), f^{\#}(1,1)) \\ & \equiv & t^{\#} \end{array}$$

To avoid this problem we introduce the AC-TRS $R^{\#} = \{f^{\#}(f(x,y),z) \rightarrow f^{\#}(f^{\#}(x,y),z) \mid f \in \Sigma_{AC}\}$. We denote by $t\downarrow_{\#}$ the normal form of t in $\underset{R^{\#}/AC}{\rightarrow}$. We write $s \xrightarrow{\#} t$ if $s \underset{R\backslash AC}{\rightarrow} t' \wedge t'\downarrow_{\#} \equiv t$ for some t'.
Note that the marking is compatible with $\xrightarrow{\#}$ for the above example:

$$\begin{array}{rcl} s^{\#} & \equiv & f^{\#}(g(0), h(f(1,1))) \\ & \xrightarrow{\#} & f^{\#}(f^{\#}(0,0), h(f(1,1))) \\ & \xrightarrow{\#} & f^{\#}(f^{\#}(0,0), f^{\#}(1,1)) \\ & \equiv & t^{\#} \end{array}$$

Indeed we can show the following compatibility for non-variable terms.

Lemma 3.20 *The following properties hold for any terms $s, t \in \mathcal{T}(\Sigma, \mathcal{V})\backslash\mathcal{V}$.*

(i) $s \xrightarrow{bd} t \iff s^{\#} \xrightarrow{\#} t^{\#}$

(ii) $s \trianglerighteq_{hd} t \iff s^{\#} \trianglerighteq_{hd} t^{\#}$

proof: Trivial. □

Definition 3.21 *A sequence of AC-dependency pairs $\langle u_0^{\#}, v_0^{\#}\rangle\langle u_1^{\#}, v_1^{\#}\rangle\langle u_2^{\#}, v_2^{\#}\rangle \cdots$ is an AC-dependency chain if there is a substitution θ over $\mathcal{T}(\Sigma, \mathcal{V})$ such that $(v_i\theta)^{\#} \xrightarrow{\#}{}^{*} \trianglerighteq_{hd} (u_{i+1}\theta)^{\#}$ for all i. We assume that different AC-dependency pair occurrences do not own jointly variables without loss of generality.*

Theorem 3.22 *An AC-TRS R is AC-terminating iff there exists no infinite AC-dependency chain of R.*

proof: From lemma 3.20, it follows that $v_i\theta \xrightarrow{bd}{}^{*} \trianglerighteq_{hd} u_{i+1}\theta$ iff $(v_i\theta)^{\#} \xrightarrow{\#}{}^{*} \trianglerighteq_{hd} (u_{i+1}\theta)^{\#}$, for any $v_i, u_{i+1} \in \mathcal{T}(\Sigma, \mathcal{V})\backslash\mathcal{V}$ and θ over $\mathcal{T}(\Sigma, \mathcal{V})$. Thus, there is an infinite unmarked AC-dependency chain iff there is an infinite AC-dependency chain. From theorem 3.17, our proof is completed. □

Definition 3.23 *A weak reduction order $\gtrsim$ is a weak AC-reduction order if*

- *$\gtrsim$ is AC-compatible, i.e., $s \underset{AC}{\sim} t \Rightarrow s \gtrsim t$, and*
- *$\gtrsim$ has the AC-deletion property, i.e., $f(f(x,y),z) \gtrsim f(x,y)$ for all AC-symbols f.*

Definition 3.24 *A quasi-order $\gtrsim$ satisfies the AC-marked condition if $\gtrsim$ satisfies the following two conditions:*

(i) $f^{\#}(f(x,y),z) \gtrsim f^{\#}(f^{\#}(x,y),z)$ *for all* $f \in \Sigma_{AC}$

(ii) $f^{\#}(f^{\#}(x,y),z) \gtrsim f^{\#}(f(x,y),z)$ *for all* $f \in \Sigma_{AC}$

Theorem 3.25 *Let R be an AC-TRS. If there is a weak AC-reduction order $\gtrsim$ satisfying the AC-marked condition such that*

(i) $l \gtrsim r$ *for all* $l \rightarrow r \in R$,

(ii) $u^{\#} > v^{\#}$ *for all* $\langle u^{\#}, v^{\#}\rangle \in DP_{AC}^{\#}(R)$,

then R is AC-terminating.

proof: We assume that R is not AC-terminating. From theorem 3.22, there is an infinite AC-dependency chain $\langle u_0^{\#}, v_0^{\#}\rangle\langle u_1^{\#}, v_1^{\#}\rangle\langle u_2^{\#}, v_2^{\#}\rangle \cdots$ with a substitution θ such that $(v_i\theta)^{\#} \xrightarrow{\#}{}^{*} \trianglerighteq_{hd} (u_{i+1}\theta)^{\#}$ for all i. It follows that $\xrightarrow{\#}{}^{*} \subseteq \gtrsim$ from the assumption (i), the AC-compatibility, the stability, the monotonicity and the AC-marked condition (i). It follows that $\trianglerighteq_{hd} \subseteq \gtrsim$ from the AC-compatibility, the AC-deletion and the stability. It follows that $(u_i\theta)^{\#} \gtrsim u_i^{\#}\theta$ from the stability, the monotonicity, and the AC-marked condition (ii). It follows that $u_i^{\#}\theta > v_i^{\#}\theta$ from the assumption (ii) and the stability of $>$. It follows that $v_i^{\#}\theta \gtrsim (v_i\theta)^{\#}$ from the stability, the monotonicity and the AC-marked condition (i). Therefore, we get an infinite decreasing sequence $(u_0\theta)^{\#} \gtrsim u_0^{\#}\theta > v_0^{\#}\theta \gtrsim (v_0\theta)^{\#} \gtrsim (u_1\theta)^{\#} \gtrsim u_1^{\#}\theta > v_1^{\#}\theta \gtrsim (v_1\theta)^{\#} \cdots$. It is a contradiction. □

In order to show the usefulness of AC-dependency pairs, we prove the AC-termination of an AC-TRS to which traditional techniques in [4, 13] cannot be applied, because the AC-TRS is not simply terminating.

Example 3.26 *Consider the following AC-TRS R with $\Sigma_{AC}^{\#} = \{g, h, h^{\#}\}$.*

$$R = \left\{ \begin{array}{rcl} f(a) & \rightarrow & f(b) \\ b & \rightarrow & g(h(a,a),a) \\ h(x,x) & \rightarrow & x \end{array} \right.$$

$$DP^{\#}_{AC}(R) = \begin{cases} \langle f^{\#}(a), f^{\#}(b)\rangle \\ \langle f^{\#}(a), b^{\#}\rangle \\ \langle b^{\#}, h^{\#}(a,a)\rangle \\ \langle h^{\#}(h^{\#}(x,x),z), h^{\#}(x,z)\rangle \end{cases}$$

Let $A = \mathcal{N}\backslash\{0\}$. We associate the polynomial $f_A = f^{\#}_A = X + 5$, $h_A = h^{\#}_A = X + Y$, $g_A = 1$, $a_A = 3$, $b_A = 2$ and $b^{\#}_A = 7$. From theorem A.4, this interpretation offers a weak AC-reduction order $\gtrsim$. Moreover, it satisfies the AC-marked condition. It is trivial that $l \gtrsim r$ for all $l \to r \in R$ and $u^{\#} \gtrsim v^{\#}$ for all $\langle u^{\#}, v^{\#}\rangle \in DP^{\#}_{AC}(R)$. Therefore, R is AC-terminating by theorem 3.25.

4 AC-Dependency Graph

The notion of dependency graphs introduced by Arts and Giesl is very useful to prove termination of TRSs [1, 2]. In this section, we extend the notion to AC-TRSs.

Definition 4.1 *An AC-dependency graph of R is a directed graph of which the nodes are AC-dependency pairs, and there is an edge from $\langle u^{\#}, v^{\#}\rangle$ to $\langle u'^{\#}, v'^{\#}\rangle$ if $\langle u^{\#}, v^{\#}\rangle\langle u'^{\#}, v'^{\#}\rangle$ is an AC-dependency chain.*

Theorem 4.2 *Let R be an AC-TRS. If there is a weak AC-reduction order $\gtrsim$ such that*

- *$l \gtrsim r$ for all $l \to r \in R$,*
- *$u^{\#} \gtrsim v^{\#}$ for all $\langle u^{\#}, v^{\#}\rangle$ on a cycle in the AC-dependency graph, and*
- *$u^{\#} > v^{\#}$ for at least one $\langle u^{\#}, v^{\#}\rangle$ on each cycle in the AC-dependency graph,*

then R is AC-terminating.

proof: As similar to the proof of theorem 3.25. □

In general, AC-dependency graphs are not computable, because it is undecidable whether there is a substitution θ such that $(v\theta)^{\#} \xrightarrow{\#}{}^* \trianglerighteq_{hd} (u'\theta)^{\#}$ for two AC-dependency pairs $\langle u^{\#}, v^{\#}\rangle$ and $\langle u'^{\#}, v'^{\#}\rangle$. Thus decidable approximations of dependency graphs were introduced [1, 2]. We also propose an approximation of AC-dependency graphs, using the techniques of Ω-reduction and Ω_V-reduction, which are introduced to analyze decidable call-by-need computations in TRSs [7, 11, 12].

Definition 4.3 *Let Ω be a special constant symbol. A term t is an Ω-term if $t \in \mathcal{T}(\Sigma \cup \{\Omega\}, \emptyset)$. The prefix order $\underset{\Omega}{\geq}$ over Ω-terms is defined as follows:*

- *$t \underset{\Omega}{\geq} \Omega$ for all t,*
- *$f(s_1, \cdots, s_n) \underset{\Omega}{\geq} f(t_1, \cdots, t_n)$ if $s_i \underset{\Omega}{\geq} t_i$ $(1 \leq i \leq n)$.*

We denote by t_Ω the Ω-term obtained from t by replacing all variables in t by Ω. Notice that $t_\Omega \underset{\Omega}{\leq} t\theta_\Omega$ for any term t and for any substitution θ.

In this section, $\mathcal{T}(\Sigma, \mathcal{V})$ and $\mathcal{T}(\Sigma \cup \{\Omega\}, \emptyset)$ are abbreviated to $\mathcal{T}$ and $\mathcal{T}_\Omega$, respectively.

Definition 4.4 *Two Ω-terms t_1 and t_2 are compatible, written by $t_1 \uparrow t_2$, if there is an Ω-term s such that $s \underset{\Omega}{\geq} t_1$ and $s \underset{\Omega}{\geq} t_2$. A body AC-$\Omega$-reduction relation over Ω-terms, written by $\underset{\Omega}{\xrightarrow{bd}}$, is defined as follows:*

$$\begin{array}{lll} s \underset{\Omega}{\xrightarrow{bd}} t & \overset{\text{def}}{\iff} & s \underset{AC}{\sim} C[s']_p \wedge s' \uparrow l_\Omega \wedge s' \not\equiv \Omega \\ & & \wedge\ t \equiv C[\Omega]_p \wedge p \notin \mathcal{O}_{head}(C[s']_p) \\ & & \textit{for some}\ \ l \to r \in R,\ s',\ C[\ ]_p. \end{array}$$

It is trivial to see that the body AC-Ω-reduction $\underset{\Omega}{\xrightarrow{bd}}$ is terminating and finite-branching, hence the set of all normal forms of t, denoted $NF_\Omega(t)$, is computable.

Definition 4.5 *A body AC-Ω_V-reduction from an Ω-term s to an Ω-term t, denoted by $s \underset{\Omega_V}{\xrightarrow{bd}} t$, is defined as follows:*

$$\begin{array}{lll} s \underset{\Omega_V}{\xrightarrow{bd}} t & \overset{\text{def}}{\iff} & s \underset{AC}{\sim} C[s']_p \wedge s' \uparrow l_\Omega \wedge s' \not\equiv \Omega \\ & & \wedge\ t \equiv C[r_\Omega]_p \wedge p \notin \mathcal{O}_{head}(C[s']_p) \\ & & \textit{for some}\ \ l \to r \in R,\ C[\ ]_p,\ s'. \end{array}$$

$\underset{\Omega_V}{\xrightarrow{bd}}{}^n$ denotes a $\underset{\Omega_V}{\xrightarrow{bd}}$ reduction of n steps.

Lemma 4.6 *For any $s, t \in \mathcal{T}_\Omega$, if $s \underset{\Omega_V}{\xrightarrow{bd}}{}^* t$ then $t' \underset{\Omega}{\leq} t$ for some $t' \in NF_\Omega(s)$.*

proof: It is trivial. □

Lemma 4.7 *The following properties hold for any $s, t \in \mathcal{T}$ and $s' \in \mathcal{T}_\Omega$ such that $s' \underset{\Omega}{\leq} s_\Omega$.*

(a) $s \underset{AC}{\sim} t \Rightarrow \exists t' \in \mathcal{T}_\Omega.\ s' \underset{AC}{\sim} t' \underset{\Omega}{\leq} t_\Omega$

(b) $s \xrightarrow{bd}{}^* t \Rightarrow \exists t' \in \mathcal{T}_\Omega.\ s' \underset{\Omega_V}{\xrightarrow{bd}}{}^* t' \underset{\Omega}{\leq} t_\Omega$

proof: It is routine. □

The predicate $isConnect(v, u')$ is defined as $isConnect(v, u') \iff \exists\theta.\ v\theta \xrightarrow{bd}{}^* \trianglerighteq_{hd} u'\theta$. Note that two AC-dependency pairs $\langle u, v\rangle$ and $\langle u', v'\rangle$ is connectable if $isConnect(v, u')$ holds. For a given approximation level n $(n \geq 0)$, the above two lemmas offer a decidable approximation $isConnect_n(v, u')$ of $isConnect(v, u')$ as follows:

- $(v)_\varepsilon \notin \Sigma_{AC}$.
$isConnect_n(v,u') = \begin{cases} True & \text{if } Connect_n(v,u') \neq \emptyset \\ False & \text{if } Connect_n(v,u') = \emptyset \end{cases}$
$Connect_n(v,u') = \{t \mid t \underset{AC}{\sim} t' \in Reach_n(v_\Omega),\ t \uparrow u'_\Omega\}$
$Reach_n(t) = \{t' \mid t \overset{bd}{\underset{\Omega_V}{\rightarrow}}{}^m t',\ m < n\} \cup \{t' \mid t \overset{bd}{\underset{\Omega_V}{\rightarrow}}{}^n t'',\ t' \in NF_\Omega(t'')\}$

- $(v)_\varepsilon \in \Sigma_{AC}$.
$isConnect_n(v,u') = \begin{cases} True & \text{if } (v)_\varepsilon = (u')_\varepsilon \\ False & \text{if } (v)_\varepsilon \neq (u')_\varepsilon \end{cases}$

Note that the predicate $isConnect_n$ is decidable for every approximation level n.

Definition 4.8 *The n-approximated AC-dependency graph of R is a directed graph of which the nodes are marked AC-dependency pairs, and there is an edge from $\langle u^\#, v^\# \rangle$ to $\langle u'^\#, v'^\# \rangle$ if $isConnect_n(v,u')$ holds.*

Example 4.9 *Let $\Sigma_{AC} = \{N\}$ and R be the following AC-TRS:*

$$\begin{cases} max(L(x)) \rightarrow x \\ max(N(L(0), L(y))) \rightarrow y \\ max(N(L(s(x)), L(s(y)))) \rightarrow s(max(N(L(x), L(y)))) \\ max(N(L(x), N(y,z))) \rightarrow max(N(L(x), L(max(N(y,z))))) \end{cases}$$

The representation $N(L(0), N(L(0), L(s(0))))$ for a binary tree naturally expresses the multiset $\{0, 0, s(0)\}$ by interpreting N as an AC-function symbol. The normal form of the term $max(t)$ corresponds with the maximal number in non-empty multiset t. For this AC-TRS R, the 1-approximated AC-dependency graph is Fig. 3.

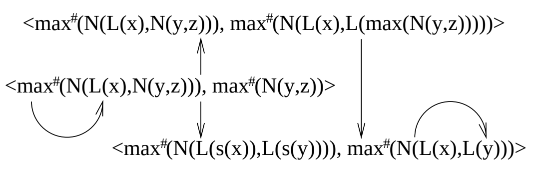

Figure 3: AC-dependency graph

Lemma 4.10 *Let R be an arbitrary AC-TRS and n an arbitrary approximation level. An AC-dependency graph of R is a subgraph of the n-approximated AC-dependency graph of R.*

proof: For each edge $(\langle u^\#, v^\# \rangle, \langle u'^\#, v'^\# \rangle)$ of an AC-dependency graph, it is enough to show $isConnect_n(v,u') = True$. If $(v)_\varepsilon \in \Sigma_{AC}$ then it is trivial. Let $(v)_\varepsilon \notin \Sigma_{AC}$ and $v\theta \overset{bd}{\rightarrow}{}^* t \underset{AC}{\sim} u'\theta$ for some t and θ. From lemma 4.7(b), $v_\Omega \overset{bd}{\underset{\Omega_V}{\rightarrow}}{}^m t'$ for some m and $t' \in \mathcal{T}_\Omega$ such that $t' \underset{\Omega}{\leq} t_\Omega$. We distinguish the following two cases:

- $m < n$:
By the assumption, $t' \in Reach_n(v_\Omega)$. From lemma 4.7(a), $t' \underset{AC}{\sim} u''$ for some $u'' \in \mathcal{T}_\Omega$ such that $u'' \underset{\Omega}{\leq} u'\theta_\Omega$. Thus, $u'' \uparrow u'_\Omega$. Therefore, $isConnect_n(v,u') = True$.

- $n \leq m$:
Let $v_\Omega \overset{bd}{\underset{\Omega_V}{\rightarrow}}{}^n t'' \overset{bd}{\underset{\Omega_V}{\rightarrow}}{}^* t'$. From lemma 4.6, $t''' \underset{\Omega}{\leq} t'$ for some $t''' \in NF_\Omega(t'')$. Thus, $t''' \in Reach_n(v_\Omega)$. From lemma 4.7(a), $t''' \underset{AC}{\sim} u''$ for some $u'' \in \mathcal{T}_\Omega$ such that $u'' \underset{\Omega}{\leq} u'\theta_\Omega$. Thus, $u'' \uparrow u'_\Omega$. Therefore, $isConnect_n(v,u') = True$. □

Theorem 4.11 *Let R be an AC-TRS and n an approximation level. If there is a weak AC-reduction order $\gtrsim$ with the AC-marked condition such that*

- *$l \gtrsim r$ for all $l \rightarrow r \in R$,*
- *$u^\# \gtrsim v^\#$ for all $\langle u^\#, v^\# \rangle$ on a cycle in the n-approximated AC-dependency graph, and*
- *$u^\# > v^\#$ for at least one $\langle u^\#, v^\# \rangle$ on each cycle in the n-approximated AC-dependency graph,*

then R is AC-terminating.

proof: It is a direct consequence of theorem 4.2 and lemma 4.10. □

Example 4.12 *Let R be the AC-TRS defined in example 4.9 and $A = \mathcal{N} \backslash \{0\}$. We associate the polynomial $0_A = 1$, $s_A = X + 1$, $L_A = X$, $N_A = X + Y$ and $max_A = max^\#_A = X$. From theorem A.4, this interpretation offers a weak AC-reduction order $\gtrsim$. Moreover, it satisfies the AC-marked condition. It is trivial that $l \gtrsim r$ for all $l \rightarrow r \in R$, and for all $\langle u^\#, v^\# \rangle$ on each cycle in the 1-approximated AC-dependency graph (Fig.3) we have $u^\# > v^\#$ as follows:*

$$max^\#(N(L(s(x)), L(s(y)))) > max^\#(N(L(x), L(y)))$$
$$max^\#(N(L(x), N(y,z))) > max^\#(N(y,z))$$

Therefore, R is AC-terminating by theorem 4.11.

This example displays the usefulness of approximated AC-dependency graphs, because it is difficult to give a weak AC-reduction order directly whose

strict part is compatible to the AC-dependency pair $\langle max^{\#}(N(L(x), N(y,z))),$
$$max^{\#}(N(L(x), L(max(N(y,z))))) \rangle.$$
Even for TRS R, it is difficult to give a weak reduction order directly whose strict part is compatible to the pair.

Note that for TRSs, that is $\Sigma_{AC} = \emptyset$, the approximated dependency graph generated by the CAP/REN method of Arts and Giesl in [1, 2] corresponds to our 0-approximated dependency graph, that is, the approximation level 0.

5 Another notion of AC-dependency Pairs

The results presented here are based on the technique developed by the authors [9]. Marché and Urbain recently proposed another idea of AC-dependency pairs in the framework of flattening terms [10], which was done independently of our work. Their definition of the extended dependency pair, which is the most basic notion of AC-dependency pairs, differs from ours as follows:

Definition 5.1 *[10] Let R be an AC-TRS. A pair $\langle u^{\#}, v^{\#} \rangle$ of terms is an extended dependency pair of R if there is a rule $l \to r \in R$ such that either*

(i) $u \equiv f(l,z)$ *and* $v \equiv f(r,z)$*, or*

(ii) $u \equiv f(l,z)$ *and* $v \equiv r|_p$ *and* $(r)_p \in DF(R)$*,*

where $(l)_\varepsilon = f \in \Sigma_{AC}$ *and z is a fresh variable.*

Note that our extended dependency pairs do not require the above condition (ii), which is essential in their framework. Hence, we have smaller extended dependency pairs in number for a given AC-TRS than they have. This fact is very useful when the AC-dependency pair method is effectively applied to automatic theorem proving. On the other hand, their method is slightly powerful than our one in a theoretical sense. In fact, suppose that our method proves AC-termination by our weak AC-reduction order $\gtrsim$ and theorem 3.25. Then, any extended dependency pairs $\langle f(l,z), r|_p \rangle$ in the above condition (ii) satisfies $f(l,z) > r|_p$. Therefore, their method also prove the AC-termination. Conversely, our approach does not always work well when their method proves AC-termination, because we additionally require the AC-deletion property for our weak AC-reduction order. However, this requirement is not strong in practice, because the AC-deletion property automatically holds for all AC-reduction orders and all weak AC-reduction orders that we have known [4, 10, 13]. Of course, the polynomial interpretation introduced in theorem A.4 also satisfies the AC-deletion property.

Acknowledgments

We would like to thank Claude Marché and Xavier Urbain for their useful discussion, and Taro Suzuki for his useful comments.

References

[1] T.Arts, "Automatically Proving Termination and Innermost Normalization of Term Rewriting Systems," PhD thesis, Univ. of Utrecht, 1997.

[2] T.Arts, J.Giesl, "Automatically Proving Termination where Simplification Orderings Fail," LNCS, vol.1214 (TAPSOFT 97), pp.261-272, 1997.

[3] F.Baader, T.Nipkow, "Term Rewriting and All That," Cambridge University Press, 1998.

[4] A.B.Cherifa, P.Lescanne, "An Actual Implementation of a Procedure that Mechanically Proves Termination of Rewriting Systems based on Inequalities between Polynomial Interpretations," LNCS, vol.230, pp.42-51, 1986.

[5] N.Dershowitz, "Orderings for Term-Rewriting Systems," Theoretical Computer Science, vol.17(3), pp.279-301, 1982.

[6] N.Dershowitz, J.-P.Jouannaud, "Rewrite Systems," in Handbook of Theoretical Computer Science, Vol.B, ed. J. van Leeuwen, North-Holland, pp.243–320, 1990.

[7] G.Huet, J.-J.Levy, "Computations in Orthogonal Rewriting Systems, I and II," in Computational Logic, Essays in Honor of Alan Robinson, eds. J.-L.Lassez and G.Plotkin, MIT Press, pp.396-443, 1991.

[8] J.W.Klop, "Term Rewriting Systems," in Handbook of Logic in Computer Science, vol.2, eds. S.Abramasky, D.Gabbay, and T.Maibaum, pp.1-116, Oxford University Press, 1992.

[9] K.Kusakari, Y.Toyama, "On Proving AC-Termination by AC-Dependency Pairs," Res. Rep. IS-RR-98-0026F, School of Information Science, JAIST, 1998.

[10] C.Marché, X.Urbain, "Termination of Associative-Commu-tative Rewriting by Dependency Pairs," LNCS, vol.1379 (RTA 98), pp.241-255, 1998.

[11] T.Nagaya, M.Sakai, Y.Toyama, "Index Reduction of Overlapping Strongly Sequential Systems," IEICE Trans. Inf.& Syst., Vol.E81-D, No.5, pp.419-426, 1998.

[12] M. Oyamaguchi, "NV-Sequentiality: A Decidable Condition for call-by-need Computations in Term Rewriting Systems," SIAM J. Computation, Vol.22, no.1, pp.112-135, 1993.

[13] A.Rubio, R.Nieuwenhuis, "A Total AC-compatible Ordering based on RPO," Theoretical Computer Science, vol.142, pp.209-227, 1995.

A Polynomial Interpretation

We introduce a weak AC-reduction order based on polynomial interpretation used in examples.

Definition A.1 *Let $A \subseteq \mathcal{N} \backslash \{0\}$. We denote a polynomial $f_A : A^n \to A$ for any $f \in \Sigma$ with arity n. We call a polynomial f_A a monotone polynomial if it depends on all its indeterminates, i.e., for all i $(1 \leq i \leq n)$, it contains a monomial (with non-zero coefficient) in which x_i occurs with an exponent at least 1. A polynomial interpretation $[\![t]\!]_\sigma$ is defined as follows: $[\![x]\!]_\sigma = \sigma(x)$ and $[\![f(t_1, \ldots, t_n)]\!]_\sigma = f_A([\![t_1]\!]_\sigma, \ldots, [\![t_n]\!]_\sigma)$ for any $\sigma : \mathcal{V} \to A$. The polynomial strict order $>_A$ is defined as follows: $s >_A t \stackrel{\text{def}}{\Longleftrightarrow} \forall\sigma([\![s]\!]_\sigma > [\![t]\!]_\sigma)$. A monotone polynomial strict order is a polynomial strict order in which all f_A is a monotone polynomial. A polynomial quasi-order $\gtrsim_A$ is defined as follows: $s \gtrsim_A t \stackrel{\text{def}}{\Longleftrightarrow} \forall\sigma([\![s]\!]_\sigma > [\![t]\!]_\sigma) \vee \forall\sigma([\![s]\!]_\sigma = [\![t]\!]_\sigma)$.*

Proposition A.2 *[3] A monotone polynomial strict order is a reduction order.*

Proposition A.3 *[4] If $f_A = f_2XY + f_1(X + Y) + f_0$ and $f_2 f_0 = f_1(f_1 - 1)$ for all $f \in \Sigma_{AC}$ then a monotone polynomial strict order is an AC-reduction order.*

Theorem A.4 *A polynomial quasi-order $\gtrsim_A$ is a weak reduction order. Furthermore, if all AC-symbols satisfy the condition of proposition A.3 then $\gtrsim_A$ is a weak AC-reduction order.*

proof: It suffices to show the monotonicity of $\gtrsim$. Let $s \gtrsim_A t$. We prove $C[s] \gtrsim_A C[t]$ by induction on the structure of $C[\ ]$. The case $C[\ ] \equiv \Box$ is trivial. Suppose that $C[\ ] \equiv f(\ldots, s_{i-1}, C'[\ ], s_{i+1}, \ldots)$. If f_A does not include X_i then it is trivial that $[\![C[s]]\!]_\sigma = [\![C[t]]\!]_\sigma$. Otherwise, from the induction hypothesis, it follows that $\forall\sigma,\ [\![C'[t]]\!]_\sigma > [\![C'[t]]\!]_\sigma$ or $\forall\sigma,\ [\![C'[t]]\!]_\sigma = [\![C'[t]]\!]_\sigma$. In the former case, $[\![C[s]]\!]_\sigma = [\![f(\ldots, C'[s], \ldots)]\!]_\sigma = f_A(\ldots, [\![C'[s]]\!]_\sigma, \ldots) > f_A(\ldots, [\![C'[t]]\!]_\sigma, \ldots) = [\![f(\ldots, C'[t], \ldots)]\!]_\sigma = [\![C[t]]\!]_\sigma$. In the latter case, it is trivial that $[\![C[s]]\!]_\sigma = [\![C[t]]\!]_\sigma$. Therefore it follows that $C[s] \gtrsim_A C[t]$. $\Box$

In general, a weak AC-reduction order generated by polynomial interpretation does not satisfy the AC-marked condition. Hence we need to restrict that $f_A = f_A^\#$ or $f_A^\#$ is constant. Under this restriction, the AC-marked condition is always fulfilled.