Regular Paper

On Proving AC-Termination by Argument Filtering Method

Keiichirou Kusakari† and Yoshihito Toyama†

The notion of dependency pairs is widely used for proving termination of TRSs. Recently, this notion was extended to AC-TRSs. Using AC-dependency pairs, we can easily show the AC-termination property of AC-TRSs to which traditional techniques cannot be applied. On this notion, a weak AC-reduction pair plays an important role. In this paper, we introduce the argument filtering method, which designs a weak AC-reduction pair from an arbitrary AC-reduction order. Moreover, we improve the method in two directions. One is the lexicographic argument filtering method, which lexicographically combines argument filtering functions to compare AC-dependency pairs. Another one is an extension by AC-multiset extension. These methods offer useful means to prove AC-termination of complicated AC-TRSs.

1. Introduction

Term rewriting systems with associative-commutative equations (AC-TRSs) can be regarded as a model for computation in which terms are reduced by directed equations modulo associative-commutative equations. AC-TRSs themselves can be regarded as functional programming languages. They can represent abstract interpreters of functional programming languages with AC-functions such as the addition and the multiplication. For example, the data structure $Node(Leaf(1), Node(Leaf(1), Leaf(2)))$ for a binary tree naturally represents 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 data structure. A membership function *member* is non-recursively defined as follows:

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

Thus AC-TRSs can model formal manipulating systems used in various applications, such as program optimization, program verification and automatic theorem proving[5),8),11)].

The AC-termination property is one of the most fundamental properties of AC-TRSs. In general, the AC-termination property is undecidable. Thus, it is important to find methods for proving AC-termination. In order to prove AC-termination, we commonly design an AC-reduction order by which all rules are ordered.

The notion of dependency pairs was introduced for proving termination of TRSs by Arts and Giesl[1),3),4)]. This notion was extended to AC-TRSs in different ways by the authors[13)] and by Marché and Urbain[14)]. Using AC-dependency pairs, we can easily show the AC-termination property of AC-TRSs to which traditional techniques cannot be applied. We explain these two AC-dependency pairs and compare two methods in the same framework. We show that the notions of weak AC-reduction orders and weak AC-reduction pairs play an important role on the method of AC-dependency pairs.

Next, we introduce the argument filtering method, which designs a weak AC-reduction order and a weak AC-reduction pair. The original idea of the argument filtering method for TRSs without AC-function symbols was first proposed by Arts and Giesl[2),10)]. The method was slightly improved by combining the subterm relation[12)]. We extend these methods to AC-TRSs. Our extension designs a weak AC-reduction order and a weak AC-reduction pair from an arbitrary AC-reduction order. Moreover, in order to strengthen the power of the argument filtering method, we improve the method in two directions. One is the lexicographic argument filtering method, which lexicographically combines argument filtering functions to compare AC-dependency pairs. Another one is an extension by AC-multiset extension. In the argument filtering method on AC-TRSs, any argument filtering function must be compatible with AC-equations. We relax this restriction using AC-multisets. These methods are effective for proving not only AC-termination but also termination of TRSs.

Lastly, using AC-dependency pairs and the argument filtering method, we analyze the dis-

† School of Information Science, JAIST

tribution elimination transformation[15),17)].

2. Preliminaries

We assume that the reader is familiar with notions of term rewriting systems[5)].

Let $\circ$ be a binary relation on a set A. We write $a_1 \circ a_2$ instead of $(a_1, a_2) \in \circ$. The binary relation $\circ$ is transitive if $\forall a_1, a_2, a_3 \in A.\ a_1 \circ a_2 \wedge a_2 \circ a_3 \Rightarrow a_1 \circ a_3$, reflexive if $\forall a \in A. a \circ a$, irreflexive if $\forall a \in A. \neg(a \circ a)$, symmetric if $\forall a_1, a_2 \in A.\ a_1 \circ a_2 \Rightarrow a_2 \circ a_1$, and antisymmetric if $\forall a_1, a_2 \in A.\ a_1 \circ a_2 \wedge a_2 \circ a_1 \Rightarrow a_1 = a_2$. We write the reflexive closure, the transitive closure and the reflexive-transitive closure of $\circ$ as $\circ^=$, $\circ^+$ and $\circ^*$, respectively. A binary relation $\circ$ is well-founded if there exists no infinite sequence such that $a_1 \circ a_2 \circ a_3 \circ \cdots$. An equivalence relation is a reflexive, transitive and symmetric relation.

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 as $\mathcal{T}(\Sigma, \mathcal{V})$. The set of variables in t is denoted by $Var(t)$. Identity of terms is denoted by $\equiv$. A term t is linear if every variable in t occurs only once. A substitution is a mapping $\theta : \mathcal{V} \to \mathcal{T}(\Sigma, \mathcal{V})$. 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. We denote the empty sequence by ε. The prefix order $\prec$ on term positions is defined as $p \prec q$ iff $pw = q$ for some $w\ (\neq \varepsilon)$. We recursively define $(t)_p$ as follows:

$$\begin{cases} (x)_\varepsilon &= x \\ (f(t_1, \cdots, t_n))_\varepsilon &= f \\ (f(t_1, \cdots, t_n))_{i \cdot p} &= (t_i)_p \end{cases}$$

A context is a term which has one special constant $\square$, called a hole. $C[t]_p$ denotes the result of replacing the hole with t at position p. A term s is called a subterm of t if $t \equiv C[s]$ for some context C.

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 strict part of a quasi-order $\gtrsim$, written by $\gnsim$, is defined as $\gtrsim \setminus \lesssim$. The equivalence part of a quasi-order $\gtrsim$ is defined as $\gtrsim \cap \lesssim$. Note that for all quasi-order its strict part is a strict order and its equivalence part is an equivalence relation. A binary relation $\circ$ on terms is monotonic if $\forall C. s \circ t \Rightarrow C[s] \circ C[t]$. A binary relation $\circ$ on terms is stable if $\forall \theta. s \circ t \Rightarrow s\theta \circ t\theta$. A congruence relation is an equivalence, monotonic and stable relation.

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 each $f \in \Sigma_{AC}$. A set $\{t_1, t_2, \ldots\}$ is AC-unifiable by θ if $t_1\theta \underset{AC}{\sim} t_2\theta \underset{AC}{\sim} \cdots$.

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 associative-commutative term rewriting system (AC-TRS) is a finite set of rules. If AC-symbols Σ_{AC} is empty, it is said to be a term rewriting system (TRS). The set of defined symbols in R is $DF(R) = \{(l)_\varepsilon \mid l \to r \in R\}$. An AC-reduction relation $\underset{R/AC}{\to}$ is defined as follows: $s \underset{R/AC}{\to} t \overset{\text{def}}{\iff} \exists\, l \to r \in R, \exists C[\ ], \exists \theta, s \underset{AC}{\sim} C[l\theta] \wedge t \equiv C[r\theta]$. We often omit the subscript R/AC when no confusion arises. An AC-TRS R is AC-terminating if $\underset{R/AC}{\to}$ is well-founded. A strict order $>$ is an AC-reduction order if $>$ is AC-compatible ($s \underset{AC}{\sim} s' > t \Rightarrow s > t$), well-founded, monotonic and stable.

Proposition 2.1 [6),13),14)] An AC-TRS R is AC-terminating iff there exists an AC-reduction order $>$ that satisfies $l > r$ for all $l \to r \in R$.

One of the most popular reduction orders is the recursive path order[7)]. The recursive path order $>_{rpo}$ is generated by a precedence $\rhd$, which is a strict order on Σ. To extend the reduction order to an AC-reduction order, flatting terms were introduced[6)]. The flatting term of a term t, denoted by $\bar{t}$, is the normal form of t for the rules $f(\vec{x}, f(\vec{y}), \vec{z}) \to f(\vec{x}, \vec{y}, \vec{z})$ for each AC-symbol f. Using flatting terms and the recursive path order $>_{rpo}$, we define $s >_{rpo}^{flat} t$ by $\bar{s} >_{rpo} \bar{t}$. However, this order $>_{rpo}^{flat}$ is not always monotonic. Therefore, we need a suitable restriction on the precedence, as shown by the following proposition.

Proposition 2.2 [6)] If all AC-symbols are minimal in a precedence $\rhd$, then the order $>_{rpo}^{flat}$ is an AC-reduction order.

For recent result, see 16).

3. AC-Dependency Pair

In this section, we review the AC-dependency pair, the weak AC-reduction order and the weak AC-reduction pair.

3.1 AC-Dependency Pair

On TRSs, Arts and Giesl introduced the notion of dependency pairs[1),3),4)], which can offer an effective method for analyzing an infinite reduction sequence. The dependency pair was extended to AC-TRSs by two different ways in 13) and 14). In this subsection, we introduce these AC-dependency pairs. In order to compare two methods in the same framework, the latter method is expressed with minor modification.

Definition 3.1 Let $\# : \Sigma \rightarrow \Sigma^{\#}$ be a marking function, where $\Sigma^{\#}$ is the set of fresh function symbols disjoint from $\mathcal{V} \cup \Sigma$. We denote $\#(f)$ by $f^{\#}$. We extend the marking function $\#$ over terms as follows:

$$\begin{cases} x^{\#} \equiv x \\ f(t_1, t_2)^{\#} \equiv f(t_1, t_2)^{\#f} \quad \text{if } f \in \Sigma_{AC} \\ f(t_1, \ldots, t_n)^{\#} \equiv f^{\#}(t_1, \ldots, t_n) \quad \text{if } f \notin \Sigma_{AC} \end{cases}$$

$$\begin{cases} x^{\#f} \equiv x \\ f(t_1, \ldots, t_n)^{\#f} \equiv f^{\#}(t_1^{\#f}, \ldots, t_n^{\#f}) \\ g(t_1, \ldots, t_m)^{\#f} \equiv g(t_1, \ldots, t_m) \quad \text{if } f \neq g \end{cases}$$

For each $f \in \Sigma_{AC}$, we regard $f^{\#}$ as an AC-symbol, i.e., $\Sigma_{AC}^{\#} = \Sigma_{AC} \cup \{f^{\#} \mid f \in \Sigma_{AC}\}$.

Definition 3.2 We define 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 define $s \underset{\#}{\rightarrow} t$ by $\exists t'.\ s \underset{R/AC}{\rightarrow} t' \wedge t'\downarrow_{\#} \equiv t$.

Definition 3.3 A head subterm relation $s \trianglerighteq_{hd} t$ is defined as $\exists C[\,]_p.\ [s \underset{AC}{\sim} C[t]_p \wedge \forall q \prec p.\ (C)_q = (t)_\varepsilon \in \Sigma_{AC}^{\#}]$.

Example 3.4 Let $\Sigma_{AC}^{\#} = \{add, add^{\#}\}$ and $t \equiv add(s(add(x,y)), add(z,w))$. Then, $t^{\#} \equiv add^{\#}(s(add(x,y)), add^{\#}(z,w))$, $t \trianglerighteq_{hd} add(z,w)$ and $t^{\#} \trianglerighteq_{hd} add^{\#}(z,w)$.

Definition 3.5 Let R be an AC-TRS. We define the AC-extended AC-TRS R^{AC}, the dependency pairs $DP^{\#}(R)$ and the AC-dependency pairs $DP_{AC}^{\#}(R)$ as follows:

$$R^{AC} = R \cup \{f(l,z) \rightarrow f(r,z) \mid l \rightarrow r \in R,\ (l)_\varepsilon = f \in \Sigma_{AC}\}$$

$$DP^{\#}(R) = \{\langle u^{\#}, v^{\#}\rangle \mid u \rightarrow C[v] \in R,\ (v)_\varepsilon \in DF(R)\}$$

$$DP_{AC}^{\#}(R) = DP^{\#}(R) \cup \{f(l,z)^{\#} \rightarrow f(r,z)^{\#} \mid l \rightarrow r \in R,\ (l)_\varepsilon = f \in \Sigma_{AC}\}$$

where z is a fresh variable.

The sets of unmarked dependency pairs and unmarked AC-dependency pairs of R, written by $DP(R)$ and $DP_{AC}(R)$, are obtained by erasing marks of symbols in $DP^{\#}(R)$ and $DP_{AC}^{\#}(R)$, respectively. The set $DP_{AC}^{\#}(R) \backslash DP^{\#}(R)$ is called extended dependency pairs of R.

Example 3.6 Consider the AC-TRS $R = \{+(x,0) \rightarrow x,\ +(x,s(y)) \rightarrow s(+(x,y))\}$ with $\Sigma_{AC} = \{+\}$. Then

$$R^{AC} = R \cup \begin{cases} +(+(x,0),z) \rightarrow +(x,z) \\ +(+(x,s(y)),z) \rightarrow +(s(+(x,y)),z) \end{cases}$$

$$DP^{\#}(R^{AC}) = \begin{cases} \langle +^{\#}(x,s(y)),\ +^{\#}(x,y)\rangle \\ \langle +^{\#}(+^{\#}(x,0),z),\ +^{\#}(x,z)\rangle \\ \langle +^{\#}(+^{\#}(x,s(y)),z),\ +^{\#}(s(+(x,y)),z)\rangle \\ \langle +^{\#}(+^{\#}(x,s(y)),z),\ +^{\#}(x,y)\rangle \end{cases}$$

$$DP_{AC}^{\#}(R) = \begin{cases} \langle +^{\#}(x,s(y)),\ +^{\#}(x,y)\rangle \\ \langle +^{\#}(+^{\#}(x,0),z),\ +^{\#}(x,z)\rangle \\ \langle +^{\#}(+^{\#}(x,s(y)),z),\ +^{\#}(s(+(x,y)),z)\rangle \end{cases}$$

The notion of AC-dependency pairs was introduced by two different ways. One introduced by us in 13) corresponds to $DP_{AC}^{\#}(R)$ and another one by Marché and Urbain in 14) corresponds to $DP^{\#}(R^{AC})$.

Proposition 3.7 [13)] An AC-TRS R is not AC-terminating iff there exist $\langle u_i^{\#}, v_i^{\#}\rangle \in DP_{AC}^{\#}(R)$ $(i = 0,1,2,\ldots)$ such that $(v_i\theta)^{\#} \overset{*}{\underset{\#}{\rightarrow}} \trianglerighteq_{hd} (u_{i+1}\theta)^{\#}$ for each i. Here, we assume that $Var(u_i^{\#}) \cap Var(u_j^{\#}) = \emptyset$ for distinct i and j without loss of generality.

Proposition 3.8 [14)] An AC-TRS R is not AC-terminating iff there exist $\langle u_i^{\#}, v_i^{\#}\rangle$

$\in DP^{\#}(R^{AC})$ $(i = 0, 1, 2, \ldots)$ such that $(v_i\theta)^{\#} \xrightarrow[\#]{*} \underset{AC}{\sim} (u_{i+1}\theta)^{\#}$ for each i. Here, we assume that $Var(u_i^{\#}) \cap Var(u_j^{\#}) = \emptyset$ for distinct i and j without loss of generality.

3.2 Weak AC-Reduction Order

In this subsection, we introduce the notion of weak AC-reduction order, which plays an important role on the method of AC-dependency pairs.

Definition 3.9 A quasi-order $\gtrsim$ is a weak AC-reduction order if $\gtrsim$ is AC-compatible ($s \underset{AC}{\sim} t \Rightarrow s \gtrsim t$), monotonic and stable, and its strict part $\gtrsim\!\!\!\!/$ is well-founded and stable. A weak AC-reduction order $\gtrsim$ has the AC-deletion property if $f(f(x,y),z) \gtrsim f(x,y)$ for all AC-symbols $f \in \Sigma_{AC}^{\#}$. A weak AC-reduction order $\gtrsim$ satisfies the AC-marked condition if for all $f \in \Sigma_{AC}$, $f^{\#}(f(x,y),z) \sim f^{\#}(f^{\#}(x,y),z)$, where $\sim$ is the equivalence part of $\gtrsim$.

An AC-reduction order $>$ can be easily extended to a weak AC-reduction order by $(> \cup \underset{AC}{\sim})^*$. On the other hand, an AC-reduction order cannot be directly obtained from a given weak AC-reduction order, because the strict part $\gtrsim\!\!\!\!/$ of a weak AC-reduction order $\gtrsim$ need not have the monotonicity property.

Proposition 3.10[13] Let R be an AC-TRS. If there exists a weak AC-reduction order $\gtrsim$ with AC-marked condition and AC-deletion property such that

- $l \gtrsim r$ for all $l \to r \in R$,
- $u^{\#} \gtrsim\!\!\!\!/\ v^{\#}$ for all $\langle u^{\#}, v^{\#} \rangle \in DP_{AC}^{\#}(R)$,

then R is AC-terminating.

Proposition 3.11 [14] Let R be an AC-TRS. If there exists a weak AC-reduction order $\gtrsim$ with AC-marked condition☆ such that

- $l \gtrsim r$ for all $l \to r \in R$,
- $u^{\#} \gtrsim\!\!\!\!/\ v^{\#}$ for all $\langle u^{\#}, v^{\#} \rangle \in DP^{\#}(R^{AC})$,

then R is AC-terminating.

Our AC-dependency pairs $DP_{AC}^{\#}(R)$ is smaller in number of pairs than Marché and Urbain's AC-dependency pairs $DP^{\#}(R^{AC})$. Hence, our method is more efficient than theirs. On the other hand, their method is more powerful than ours in theoretical, because our method requests the AC-deletion property.

3.3 Weak AC-Reduction Pair

In order to analyze transformation methods for proving termination, we extended the notion of weak reduction order to that of weak reduction pair[12]. In this subsection, we extend the notion to AC-TRSs.

Definition 3.12 A pair $(\gtrsim, >)$ of binary relations on terms is a weak AC-reduction pair if it satisfies the following conditions:

- $\gtrsim$ is AC-compatible ($s \underset{AC}{\sim} t \Rightarrow s \gtrsim t$).
- $\gtrsim$ is monotonic and stable.
- $>$ is stable and well-founded.
- $\gtrsim \cdot > \subseteq >$ or $> \cdot \gtrsim \subseteq >$.

A weak AC-reduction pair $(\gtrsim, >)$ has the AC-deletion property if for all $f \in \Sigma_{AC}^{\#}$

- $f(f(x,y),z) \gtrsim f(x,y)$ or $f(f(x,y),z) > f(x,y)$.

A weak AC-reduction pair $(\gtrsim, >)$ satisfies the AC-marked condition if for all $f \in \Sigma_{AC}$

- $f^{\#}(f(x,y),z) \sim f^{\#}(f^{\#}(x,y),z)$,

where $\sim$ is the equivalence part of $\gtrsim$.

In the above definition, we do not assume that $\gtrsim$ is a quasi-order or $>$ is a strict order. This simplifies the design of a weak AC-reduction pair. We should mention that this simplification does not lose the generality of our definition, because for a given weak AC-reduction pair $(\gtrsim, >)$ we can make the weak reduction pair $(\gtrsim^*, >^+)$ in which $\gtrsim^*$ is a quasi-order and $>^+$ is a strict order.

Theorem 3.13 For any AC-TRS R, the following properties are equivalent.

(1) AC-TRS R is AC-terminating.
(2) There exists a weak AC-reduction pair $(\gtrsim, >)$ with AC-deletion property such that $\forall l \to r \in R.\ l \gtrsim r$ and $\forall \langle u, v \rangle \in DP_{AC}(R).\ u > v$.
(3) There exists a weak AC-reduction pair $(\gtrsim, >)$ such that $\forall l \to r \in R.\ l \gtrsim r$ and $\forall \langle u, v \rangle \in DP(R^{AC}).\ u > v$.
(4) There exists a weak AC-reduction pair $(\gtrsim, >)$ with AC-marked condition and AC-deletion property such that $\forall l \to r \in$

☆ This AC-marked condition is slightly modified, because their original definition cannot handle collapsing rules, i.e., the rules whose right hand sides are variables.

R. $l \gtrsim r$ and $\forall \langle u^{\#}, v^{\#} \rangle \in DP^{\#}_{AC}(R)$. $u^{\#} > v^{\#}$.

(5) There exists a weak AC-reduction pair $(\gtrsim, >)$ with AC-marked condition such that $\forall l \to r \in R$. $l \gtrsim r$ and $\forall \langle u^{\#}, v^{\#} \rangle \in DP^{\#}(R^{AC})$. $u^{\#} > v^{\#}$.

Proof. For the cases $(1 \Rightarrow 2)$ and $(1 \Rightarrow 3)$, we define $\gtrsim$ by $(\underset{R/AC}{\to} \cup \underset{AC}{\sim})^*$, and $s > t$ by $s \underset{AC}{\not\sim} t$ and $s \gtrsim C[t]$ for some C. Then, it is easily shown that $(\gtrsim, >)$ is a weak AC-reduction pair with AC-deletion property such that $l \gtrsim r$ for all $l \to r \in R$ and $u > v$ for all $\langle u, v \rangle \in DP(R^{AC}) \supseteq DP_{AC}(R)$. For the cases $(2 \Rightarrow 4)$ and $(3 \Rightarrow 5)$, it is easily shown by identifying $f^{\#}$ with f. For the case $(4 \Rightarrow 1)$, it is easily proved from proposition 3.7. For the case $(5 \Rightarrow 1)$, it is easily proved from proposition 3.8. □

For a given AC-terminating AC-TRS, it is still open whether there exists a weak AC-reduction order satisfying one of the conditions in proposition 3.10 or 3.11. On the other hand, the above theorem guarantees the existence of a weak AC-reduction pair.

4. Argument Filtering Method

The original idea of the argument filtering method for TRSs without AC-function symbols was first proposed by Arts and Giesl[2),10)]. In this section, we extend the idea to AC-TRSs. Our extension designs a weak AC-reduction order and a weak AC-reduction pair from an arbitrary AC-reduction order. Moreover, we improve the method in two directions. One is the lexicographic argument filtering method, which lexicographically combines argument filtering functions to compare AC-dependency pairs. Another one is an extension on AC-multisets. These methods offer useful means to prove AC-termination of complicated AC-TRSs.

4.1 Argument Filtering Method

Definition 4.1 An argument filtering function is a function π such that for any $f \in \Sigma$, $\pi(f)$ is either an integer i or a list of integers $[i_1, \ldots, i_m]$ $(m \geq 0)$, where those integers $i, i_1, \ldots, i_m$ are positive and at most $arity(f)$. We can naturally extend π over terms as follows:

$$\begin{aligned} \pi(x) &= x \\ \pi(f(t_1, \ldots, t_n)) &= \pi(t_i) \quad \text{if } \pi(f) = i \\ \pi(f(t_1, \ldots, t_n)) &= f(\pi(t_{i_1}), \ldots, \pi(t_{i_m})) \\ & \qquad \text{if } \pi(f) = [i_1, \ldots, i_m] \end{aligned}$$

We denote by $\pi(\theta)$ the substitution defined as $\pi(\theta)(x) = \pi(\theta(x))$ for all $x \in \mathcal{V}$.
We hereafter assume that if $\pi(f)$ is not defined explicitly then it is intended to be $[1, \ldots, arity(f)]$.

Definition 4.2 An argument filtering function π satisfies the AC-condition if for all $f \in \Sigma^{\#}_{AC}$, $\pi(f)$ is either $[]$ or $[1, 2]$.

The above restriction is essential in AC-TRSs, because it guarantees that the image of the associative and commutative axioms for $f \in \Sigma_{AC}$ are either $f = f$ or themselves. We define AC-function symbols after argument filtering by $\Sigma^{\#}_{AC,\pi} = \{f \in \Sigma^{\#}_{AC} \mid \pi(f) = [1, 2]\}$. We also write by $\underset{AC}{\sim}$ the AC-equation generated by $\Sigma^{\#}_{AC,\pi}$. Then, it follows that $s \underset{AC}{\sim} t$ implies $\pi(s) \underset{AC}{\sim} \pi(t)$.

Definition 4.3 We define the AC-extension $\gtrsim_{AC}$ of a strict order $>$ by $\gtrsim_{AC} = (> \cup \underset{AC}{\sim})^*$. We define $s \gtrsim^{sub}_{AC} t$ by $s \gtrsim_{AC} C[t]$ for some C, and $\gnsim^{sub}_{AC}$ by its strict part.

Let $>$ be an AC-reduction order. Then $\gtrsim^{sub}_{AC}$ is a quasi-order and the strict part $\gnsim_{AC}$ of its AC-extension $\gtrsim_{AC}$ is also AC-reduction order.

Lemma 4.4 If a strict order $>$ is AC-compatible then $\gtrsim_{AC} = >^{=} \cdot \underset{AC}{\sim}$.

Proof. It is trivial. □

Lemma 4.5 Let $>$ be an AC-reduction order. Then $\gnsim^{sub}_{AC}$ is well-founded.

Proof. We assume that there exists an infinite decreasing sequence $t_0 \gnsim^{sub}_{AC} t_1 \gnsim^{sub}_{AC} t_2 \gnsim^{sub}_{AC} \cdots$. Then, there exist C_i $(i = 1, 2, \ldots)$ such that $t_i \gtrsim_{AC} C_{i+1}[t_{i+1}]$. Here, $\gtrsim_{AC}$ has the monotonicity, because $>$ and $\underset{AC}{\sim}$ have the monotonicity. Thus, $t_0 \gtrsim_{AC} C_1[t_1] \gtrsim_{AC} C_1[C_2[t_2]] \cdots$. From the well-foundedness of $>$ and lemma 4.4, there is some k such that $C'_k[t_k] \underset{AC}{\sim} C'_{k+1}[t_{k+1}] \underset{AC}{\sim} C'_{k+2}[t_{k+2}] \cdots$, where $C'_i \equiv C_1[\cdots C_i[] \cdots]$.

Since $\underset{AC}{\sim}$ preserves the size of terms, there is some m such that $C_m \equiv \Box$. Hence, it follows that $t_{m-1} \underset{AC}{\sim} t_m$. It is a contradiction to $t_{m-1} \gtrsim^{sub}_{AC} t_m$. □

Definition 4.6 Let $>$ be a strict order and π an argument filtering function. We define $s \gtrsim_\pi t$ by $\pi(s) \gtrsim_{AC} \pi(t)$, and $s >_\pi t$ by $\pi(s) \gtrsim^{sub}_{AC} \pi(t)$.

Lemma 4.7 Let $>$ be an AC-reduction order. Then the following properties hold:

- $s \gtrsim_\pi t \iff \pi(s) >^= \cdot \underset{AC}{\sim} \pi(t)$
- $s \gtrsim_\pi t \wedge t \gtrsim_\pi s \iff \pi(s) \underset{AC}{\sim} \pi(t)$
- $s >_\pi t \iff \exists C.\ \pi(s) > \cdot \underset{AC}{\sim} C[\pi(t)]$ or $\exists C \not\equiv \Box.\ \pi(s) \underset{AC}{\sim} C[\pi(t)]$.

Proof. It suffices to show implications from left to right.

The first property is a direct consequence of lemma 4.4.

Let $s \gtrsim_\pi t \wedge t \gtrsim_\pi s$. From lemma 4.4, $\pi(s) >^= \cdot \underset{AC}{\sim} \pi(t) >^= \cdot \underset{AC}{\sim} \pi(s)$. If $\pi(s) \underset{AC}{\not\sim} \pi(t)$ then $\pi(s) > \cdot \underset{AC}{\sim} \pi(s)$. It is a contradiction to the well-foundedness of $>$. Hence the second property holds.

Let $s >_\pi t$. Then $\pi(s) \gtrsim_{AC} C[\pi(t)]$. From lemma 4.4, $\pi(s) >^= \cdot \underset{AC}{\sim} C[\pi(t)]$. Hence, $\pi(s) \underset{AC}{\sim} C[\pi(t)]$ or $\pi(s) > \cdot \underset{AC}{\sim} C[\pi(t)]$. In the former case, if $C \equiv \Box$ then $\pi(s) \underset{AC}{\sim} \pi(t)$. It is a contradiction to $\pi(s) \gtrsim^{sub}_{AC} \pi(t)$. Hence $C \not\equiv \Box$. Therefore the third property holds. □

Lemma 4.8 $\pi(\theta)(\pi(t)) \equiv \pi(t\theta)$

Proof. We prove the claim by induction on t. The case $t \in \mathcal{V}$ is trivial. Suppose that $t \equiv f(t_1, \ldots, t_n)$. In the case $\pi(f) = j$, it follows that $\pi(\theta)(\pi(f(t_1, \ldots, t_n))) \equiv \pi(\theta)(\pi(t_j)) \equiv \pi(t_j\theta) \equiv \pi(f(t_1, \ldots, t_n)\theta)$. In the case $\pi(f) = [i_1, \ldots, i_m]$, the following relation holds;

$$\begin{aligned}
&\pi(\theta)(\pi(f(t_1, \ldots, t_n))) \\
&\quad \equiv \pi(\theta)(f(\pi(t_{i_1}), \ldots, \pi(t_{i_m}))) \\
&\quad \equiv f(\pi(\theta)(\pi(t_{i_1})), \ldots, \pi(\theta)(\pi(t_{i_m}))) \\
&\quad \equiv f(\pi(t_{i_1}\theta), \ldots, \pi(t_{i_m}\theta)) \\
&\quad \equiv \pi(f(t_1\theta, \ldots, t_n\theta)) \\
&\quad \equiv \pi(f(t_1, \ldots, t_n)\theta)
\end{aligned}$$

□

Theorem 4.9 If $>$ is an AC-reduction order and π is an argument filtering function with AC-condition then $(\gtrsim_\pi, >_\pi)$ is a weak AC-reduction pair with AC-deletion property.

Proof.

- (The AC-compatibility of $\gtrsim_\pi$): It is a direct consequence of the first property of lemma 4.7.
- (The monotonicity of $\gtrsim_\pi$): Let $s \gtrsim_\pi t$. In the case that $\Box$ does not occur in $\pi(C)$, it follows that $\pi(C[s]) \equiv \pi(C) \equiv \pi(C[t])$. Thus, $C[s] \gtrsim_\pi C[t]$ holds. In the other case, it follows that $\pi(s) \gtrsim_{AC} \pi(t) \Rightarrow \pi(C)[\pi(s)] \gtrsim_{AC} \pi(C)[\pi(t)] \Rightarrow \pi(C[s]) \gtrsim_{AC} \pi(C[t])$. Thus, $C[s] \gtrsim_\pi C[t]$ holds.
- (The stability of $\gtrsim_\pi$): From lemma 4.8, $s \gtrsim_\pi t \Rightarrow \pi(s) \gtrsim_{AC} \pi(t) \Rightarrow \pi(\theta)(\pi(s)) \gtrsim_{AC} \pi(\theta)(\pi(t)) \Rightarrow \pi(s\theta) \gtrsim_{AC} \pi(t\theta) \Rightarrow s\theta \gtrsim_\pi t\theta$.
- (The stability of $>_\pi$): Thanks to lemmas 4.7 and 4.8, if $\pi(s) \underset{AC}{\sim} C[\pi(t)]$ then $\pi(s\theta) \underset{AC}{\sim} C'[\pi(t\theta)]$ is trivial, where $C' \equiv \pi(\theta)(C)$. Suppose that $\pi(s) > \cdot \underset{AC}{\sim} C[\pi(t)]$.

$$\begin{aligned}
s >_\pi t &\Rightarrow \pi(s) > \cdot \underset{AC}{\sim} C[\pi(t)] \\
&\Rightarrow \pi(\theta)(\pi(s)) > \cdot \underset{AC}{\sim} \pi(\theta)(C[\pi(t)]) \\
&\Rightarrow \pi(\theta)(\pi(s)) > \cdot \underset{AC}{\sim} C'[\pi(\theta)(\pi(t))] \\
&\quad \text{where } C' \equiv \pi(\theta)(C) \\
&\Rightarrow \pi(s\theta) > \cdot \underset{AC}{\sim} C'[\pi(t\theta)] \\
&\Rightarrow s\theta >_\pi t\theta
\end{aligned}$$

- (The well-foundedness of $>_\pi$): Assuming that the existence of an infinite decreasing sequence $t_0 >_\pi t_1 >_\pi t_2 >_\pi \cdots$, it follows $\pi(t_0) \gtrsim^{sub}_{AC} \pi(t_1) \gtrsim^{sub}_{AC} \pi(t_2) \gtrsim^{sub}_{AC} \cdots$. It is a contradiction to lemma 4.5.
- ($\gtrsim_\pi \cdot >_\pi \subseteq >_\pi$): Let $t_0 \gtrsim_\pi t_1 >_\pi t_2$. From lemma 4.7, either $\pi(t_0) >^= \cdot \underset{AC}{\sim} \pi(t_1) > \cdot \underset{AC}{\sim} C[\pi(t_2)]$ or $\pi(t_0) >^= \cdot \underset{AC}{\sim} \pi(t_1) \underset{AC}{\sim} C[\pi(t_2)] \wedge C \not\equiv \Box$ holds. In the former case, $\pi(t_0) > \cdot \underset{AC}{\sim} C[\pi(t_2)]$ from the AC-compatibility and the transitivity of $>$. Thus, it follows that $t_0 >_\pi t_2$ from lemma 4.7. In the latter case, $\pi(t_0) >^= \cdot \underset{AC}{\sim} C[\pi(t_2)]$ and $C \not\equiv \Box$. Thus, it follows that $t_0 >_\pi t_2$ from lemma 4.7.
- (The AC-deletion property): Suppose that $f \in \Sigma^\#_{AC}$. If $\pi(f) = []$ then $\pi(f(f(x, y), z)) \equiv f \equiv \pi(f(x, y))$. Hence, it follows that $f(f(x, y), z) \gtrsim_\pi f(x, y)$. If $\pi(f) = [1, 2]$ then $\pi(f(f(x, y), z)) \equiv f(f(x, y), z) \equiv$

$C[\pi(f(x,y))]$ for $C \equiv f(\Box, z)$. Hence, it follows that $f(f(x,y),z) >_\pi f(x,y)$. □

In order to show the usefulness of the argument filtering method, we prove the AC-termination of AC-TRSs to which traditional techniques cannot be applied.

Example 4.10 As an AC-reduction order $>$, we use the order $>^{flat}_{rpo}$. We also suppose that for any AC-symbols f, $f^\#$ is identified to f or $\pi(f^\#) = []$. Hence, $\gtrsim_\pi$ trivially satisfies the AC-marked condition. The AC-termination of each R_i is proved by using theorem 3.13 ($4 \Rightarrow 1$) and theorem 4.9.

- Consider the following AC-TRS R_1 with $\Sigma^\#_{AC} = \{g\}$.

$$R_1 = \{\ f(f(x)) \ \rightarrow\ f(g(f(x), f(x)))$$

$$DP^\#_{AC}(R_1) = \left\{ \begin{array}{l} \langle f^\#(f(x)), f^\#(x)\rangle \\ \langle f^\#(f(x)), f^\#(g(f(x), f(x)))\rangle \end{array} \right.$$

Let $\pi(g) = []$ and $f \rhd g$. Then, $l \gtrsim_\pi r$ for all $l \rightarrow r \in R_1$, and $u^\# >_\pi v^\#$ for all $\langle u^\#, v^\#\rangle \in DP^\#_{AC}(R_1)$. Therefore R_1 is AC-terminating.

- Consider the following AC-TRS R_2 with $\Sigma^\#_{AC} = \{h\}$.

$$R_2 = \left\{ \begin{array}{lcl} f(f(x)) & \rightarrow & f(g(x)) \\ g(x) & \rightarrow & h(f(x), f(x)) \end{array} \right.$$

$$DP^\#_{AC}(R_2) = \left\{ \begin{array}{l} \langle f^\#(f(x)), f^\#(g(x))\rangle \\ \langle f^\#(f(x)), g^\#(x)\rangle \\ \langle g^\#(x), f^\#(x)\rangle \end{array} \right.$$

Let $\pi(h) = []$, $f \rhd g \rhd h$ and $f \rhd g^\# \rhd f^\#$. Then, $l \gtrsim_\pi r$ for all $l \rightarrow r \in R_2$, and $u^\# >_\pi v^\#$ for all $\langle u^\#, v^\#\rangle \in DP^\#_{AC}(R_2)$. Therefore R_2 is AC-terminating.

- Consider the following AC-TRS R_3 with $\Sigma^\#_{AC} = \{g, h, h^\#\}$.

$$R_3 = \left\{ \begin{array}{lcl} f(a) & \rightarrow & f(b) \\ b & \rightarrow & g(h(a,a), a) \\ h(x,x) & \rightarrow & x \end{array} \right.$$

$$DP^\#_{AC}(R_3) = \left\{ \begin{array}{l} \langle f^\#(a), f^\#(b)\rangle \\ \langle f^\#(a), b^\#\rangle \\ \langle b^\#, h^\#(a,a)\rangle \\ \langle h^\#(h^\#(x,x), z), h^\#(x,z)\rangle \end{array} \right.$$

Let $\pi(g) = []$, $b^\# \rhd a \rhd b \rhd g$ and $f^\# \rhd b^\# \rhd h^\#$. Then, $l \gtrsim_\pi r$ for all $l \rightarrow r \in R_3$, and $u^\# >_\pi v^\#$ for all $\langle u^\#, v^\#\rangle \in DP^\#_{AC}(R_3)$. Therefore R_3 is AC-terminating.

- Consider the following AC-TRS R_4 with $\Sigma^\#_{AC} = \{f, f^\#, h\}$.

$$R_4 = \left\{ \begin{array}{lcl} f(a,x) & \rightarrow & f(b,x) \\ b & \rightarrow & h(a,a) \end{array} \right.$$

$$DP^\#_{AC}(R_4) = \left\{ \begin{array}{l} \langle f^\#(a,x), f^\#(b,x)\rangle \\ \langle f^\#(a,x), b^\#\rangle \\ \langle f^\#(f^\#(a,x), z), f^\#(f^\#(b,x), z)\rangle \end{array} \right.$$

Let $\pi(h) = []$, $a \rhd b \rhd h$ and $a \rhd b^\#$. Then, $l \gtrsim_\pi r$ for all $l \rightarrow r \in R_4$, and $u^\# >_\pi v^\#$ for all $\langle u^\#, v^\#\rangle \in DP^\#_{AC}(R_4)$. Therefore R_4 is AC-terminating.

As in the proof of theorem 4.9, it can be proved that $\gtrsim_\pi$ is a weak AC-reduction order for any given AC-reduction order $>$. For designing a weak AC-reduction order, the argument filtering method is essentially a special form of recursive program schema (RPS). Indeed, Marché and Urbain proved a similar result in a general framework of AC-RPS[14].

4.2 Lexicographic Argument Filtering Method

By combining several argument filtering functions, we can strengthen the power of the argument filtering method. In this subsection, we propose the lexicographic argument filtering method, which lexicographically combines argument filtering functions to compare AC-dependency pairs. The method presented here offers useful means to prove AC-termination of complicated AC-TRSs on which a single argument filtering function does not work.

In this subsection, we suppose that $f^\#$ is identified to f or $\pi(f^\#) = []$ for any AC-symbol f. This restriction guarantees the AC-marked condition of $\gtrsim_\pi$ and $\pi((t\theta)^\#) \equiv \pi(t^\#\theta)$ if t is not a variable. The same restriction was supposed in example 4.10, because theorem 3.13 requests the AC-marked condition.

Theorem 4.11 Let R be an AC-TRS, $>$ an AC-reduction order and π an argument filtering function with AC-condition. Suppose that $\forall l \rightarrow r \in R.\ l \gtrsim_\pi r$ and $\forall \langle u^\#, v^\#\rangle \in DP^\#_{AC}(R).\ u^\# \gtrsim_\pi v^\# \ \vee\ u^\# >_\pi v^\#$. Then, R is not AC-terminating if and only if there exist $\langle u^\#_i, v^\#_i\rangle \in DP^\#_{AC}(R)$ ($i = 0, 1, 2, \ldots$) and a substitution θ, such that $\forall i.\ (v_i\theta)^\# \xrightarrow[\#]{*} \unrhd_{hd} (u_{i+1}\theta)^\#$ and $\{\pi(u^\#_0), \pi(v^\#_0), \pi(u^\#_1), \pi(v^\#_1), \ldots\}$ is AC-unifiable by $\pi(\theta)$.

Proof. ($\Leftarrow$) It is trivial from proposition 3.7.

($\Rightarrow$) From proposition 3.7, there exist $\langle u_i^\#, v_i^\# \rangle \in DP_{AC}^\#(R)$ $(i = 0, 1, 2, \ldots)$ and a substitution θ such that $\forall i.\ (v_i\theta)^\# \underset{\#}{\overset{*}{\rightarrow}} \trianglerighteq_{hd} (u_{i+1}\theta)^\#$. From the assumption and the marked condition, $\forall i.\ (u_i\theta)^\# \gtrsim_\pi (v_i\theta)^\# \vee (u_i\theta)^\# >_\pi (v_i\theta)^\#$. From the assumption and the AC-deletion property, $\forall i.\ (v_i\theta)^\# \gtrsim_\pi (u_{i+1}\theta)^\# \vee (v_i\theta)^\# >_\pi (u_{i+1}\theta)^\#$. From the well-foundedness and lemma 4.7, there is some number k such that all $\pi((u_i\theta)^\#)$ and $\pi((v_i\theta)^\#)$ $(i \geq k)$ are AC-equivalent. The assumption $f = f^\#$ or $\pi(f^\#) = []$ for any AC-symbol f yields AC-equivalence among $\pi(u_i^\#\theta)$ and $\pi(v_i^\#\theta)$ for all $i \geq k$. From lemma 4.8, all $\pi(u_i^\#)\pi(\theta)$ and $\pi(v_i^\#)\pi(\theta)$ $(i \geq k)$ are AC-equivalent. Therefore, $\{\pi(u_k^\#), \pi(v_k^\#), \pi(u_{k+1}^\#), \pi(v_{k+1}^\#), \ldots\}$ is AC-unifiable by $\pi(\theta)$. □

The following theorem gives a sufficient condition under which the lexicographic argument filtering method works well. In order to simplify the discussion, we treat only two argument filtering functions, though the following discussion can be easily extended to the finite number of argument filtering functions.

Theorem 4.12 Let R be an AC-TRS. If there exist AC-reduction orders $>^1$ and $>^2$ and argument filtering functions π_1 and π_2 with AC-condition such that

- $l \gtrsim^1_{\pi_1} r \wedge l \gtrsim^2_{\pi_2} r$ for all $l \rightarrow r \in R$,
- $u^\# \gtrsim^1_{\pi_1} v^\# \vee u^\# >^1_{\pi_1} v^\#$ for all $\langle u^\#, v^\# \rangle \in DP_{AC}^\#(R)$, and
- $u^\# >^2_{\pi_2} v^\#$ for all $\langle u^\#, v^\# \rangle \in DP_{AC}^\#(R)$ such that $\pi_1(u^\#)$ and $\pi_1(v^\#)$ are AC-unifiable,

then R is AC-terminating.

Proof. We assume that R is not AC-terminating. From theorem 4.11, there exist $\langle u_i^\#, v_i^\# \rangle \in DP_{AC}^\#(R)$ $(i = 0, 1, 2, \ldots)$ and a substitution θ, such that $\forall i.\ (v_i\theta)^\# \underset{\#}{\overset{*}{\rightarrow}} \trianglerighteq_{hd} (u_{i+1}\theta)^\#$ and $\{\pi_1(u_0^\#), \pi_1(v_0^\#), \pi_1(u_1^\#), \pi_1(v_1^\#), \ldots\}$ is AC-unifiable. From $l \gtrsim^2_{\pi_2} r$ for all $l \rightarrow r \in R$ and AC-deletion property of $(\gtrsim^2_{\pi_2}, >^2_{\pi_2})$, it follows that $(v_i\theta)^\# \gtrsim^2_{\pi_2} (u_{i+1}\theta)^\#$ or $(v_i\theta)^\# >^2_{\pi_2} (u_{i+1}\theta)^\#$ for any i. From $u^\# >^2_{\pi_2} v^\#$ for all $\langle u^\#, v^\# \rangle \in DP_{AC}^\#(R)$ such that $\pi_1(u^\#)$ and $\pi_1(v^\#)$ are AC-unifiable, it follows that $(u_i\theta)^\# >^2_{\pi_2} (v_i\theta)^\#$ for any i. It is a contradiction to the well-foundedness of $>^2_{\pi_2}$. □

In order to show the usefulness of the lexicographic argument filtering method, we prove the AC-termination of an AC-TRS to which not only traditional techniques but also single argument filtering function cannot be applied.

Example 4.13 As an AC-reduction order $>$, we use the order $>^{flat}_{rpo}$. Consider the following AC-TRS R_5 with $\Sigma_{AC}^\# = \{g, g^\#\}$.

$$R_5 = \left\{ \begin{array}{rcl} f(x, 0) & \rightarrow & s(0) \\ f(s(x), s(y)) & \rightarrow & s(f(x, y)) \\ g(0, x) & \rightarrow & g(f(x, x), x) \end{array} \right.$$

$$DP_{AC}^\#(R_5) = \left\{ \begin{array}{l} \langle f^\#(s(x), s(y)), f^\#(x, y) \rangle \\ \langle g^\#(0, x), g^\#(f(x, x), x) \rangle \\ \langle g^\#(0, x), f^\#(x, x) \rangle \\ \langle g^\#(g^\#(0, x), z), g^\#(g^\#(f(x, x), x), z) \rangle \end{array} \right.$$

Let $\pi_1(s) = \pi_1(f) = \pi_1(f^\#) = []$ and $0 \rhd^1 f = f^\# \rhd^1 s$. Then, $l \gtrsim^1_{\pi_1} r$ for all $l \rightarrow r \in R_5$, and $u^\# \gtrsim^1_{\pi_1} v^\#$ or $u^\# >^1_{\pi_1} v^\#$ for all $\langle u^\#, v^\# \rangle \in DP_{AC}^\#(R_5)$. Let $\pi_2(g) = \pi_2(g^\#) = []$ and $f \rhd^2 s$. Then $l \gtrsim^2_{\pi_2} r$ for all $l \rightarrow r \in R_5$ and $f^\#(s(x), s(y)) >^2_{\pi_2} f^\#(x, y)$, which is an only AC-unifiable AC-dependency pair after argument filtering by π_1. From theorem 4.12, R_5 is AC-terminating.

It should be mentioned that traditional proof techniques by simplification orders cannot be directly applied to R_5 even if there exist no AC-function symbols, i.e., $\Sigma_{AC} = \emptyset$. However, the above lexicographic argument filtering method also proves termination of TRS R_5.

Corollary 4.14 Let R be an AC-TRS. If for any $i = 1, 2, \ldots, n$ there exist AC-reduction orders $>^i$ and argument filtering functions π_i with AC-condition such that

- $l \gtrsim^i_{\pi_i} r$ for all i and $l \rightarrow r \in R$,
- $u^\# \gtrsim^i_{\pi_i} v^\# \vee u^\# >^i_{\pi_i} v^\#$ for all i and $\langle u^\#, v^\# \rangle \in DP_{AC}^\#(R)$ such that for any $j < i$, $\pi_j(u^\#)$ and $\pi_j(v^\#)$ are AC-unifiable,
- $u^\# >^n_{\pi_n} v^\#$ for all $\langle u^\#, v^\# \rangle \in DP_{AC}^\#(R)$ such that for all $j < n$, $\pi_j(u^\#)$ and $\pi_j(v^\#)$ are AC-unifiable,

then R is AC-terminating.

4.3 Extension by AC-Multisets

In this subsection, we design a new argument filtering method by using AC-multisets. We first explain the notions of AC-multiset and AC-multiset extension.

A multiset is a set of terms in which elements may have multiple occurrences. We use standard set notation like $\{s, s, t\}$. For any multiset $M = \{t_1, t_2, \ldots, t_n\}$, we define the AC-multiset $M_{AC} = \{[\![t_1]\!], [\![t_2]\!], \ldots, [\![t_n]\!]\}$, where $[\![t_i]\!]$ is the equivalence class of t_i modulo $\underset{AC}{\sim}$. We define $S =_{AC} T$ by $S_{AC} = T_{AC}$. For example, $\{f(0,1), f(f(0,1),2)\} =_{AC} \{f(1,0), f(0, f(1,2))\}$ for AC-symbol f. Let $\gtrsim$ be a quasi-order whose equivalence part is equal to $\underset{AC}{\sim}$. We define its AC-multiset extension $\ggcurly$ by $S \ggcurly T$ iff $\forall [\![t]\!] \in T_{AC} - S_{AC}.\ \exists [\![s]\!] \in S_{AC} - T_{AC}.\ s \gtrsim t$. Note that for any $s' \in [\![s]\!]$ and $t' \in [\![t]\!]$, $s \gtrsim t$ implies $s' \gtrsim t'$, because the equivalence part of the quasi-order $\gtrsim$ is equal to $\underset{AC}{\sim}$.

Proposition 4.15 [9] For any quasi-order $\gtrsim$ whose equivalence part is equal to $\underset{AC}{\sim}$, its AC-multiset extension $\ggcurly$ is a quasi-order. Moreover, if the strict part of $\gtrsim$ is well-founded then so is the strict part of $\ggcurly$.

An argument filtering function π cannot preserve the AC-equivalence (i.e., $s \underset{AC}{\sim} t \Rightarrow \pi(s) \underset{AC}{\sim} \pi(t)$) without the AC-condition. Hence, we cannot treat an argument filtering function π if $\pi(f) = 1$ or $\pi(f) = 2$ for some $f \in \Sigma_{AC}$, because $\pi(f(x,y)) = \pi(f(y,x))$ makes $x = y$ or $y = x$ for the axiom of commutative law $f(x,y) =_C f(y,x)$. This problem can be avoided by defining $\hat{\pi}(f(x,y)) = \{x, y\}$. In this subsection, in order to treat such π, we introduce the extension $\hat{\pi}$ of argument filtering function π over multisets by permitting $\pi(f) = 0$ as an exception for any $f \in \Sigma^{\#}$.

Definition 4.16 We define the argument filtering function $\hat{\pi}$ from terms to multisets as follows:

$$\hat{\pi}(x) = \{x\}$$
$$\hat{\pi}(f(\vec{t_i})) = \hat{\pi}(t_j) \qquad \text{if } \pi(f) = j\ (\neq 0)$$
$$\hat{\pi}(f(\vec{t_i})) = \cup_i \hat{\pi}(t_i) \qquad \text{if } \pi(f) = 0$$
$$\hat{\pi}(f(\vec{t_i})) = \{f(\vec{t'}_{i_j}) \mid t'_{i_j} \in \hat{\pi}(t_{i_j})\} \qquad \text{if } \pi(f) = [i_1, \ldots, i_m]$$

We also define the substitution $\hat{\pi}(\theta)$ from terms to multisets as follows:

$$\hat{\pi}(\theta)(x) = \hat{\pi}(\theta(x))$$
$$\hat{\pi}(\theta)(f(\vec{t_i})) = \{f(\vec{t'_i}) \mid t'_i \in \hat{\pi}(\theta)(t_i)\}$$

We extend $\hat{\pi}(\theta)$ over multisets as follows:

$$\hat{\pi}(\theta)(T) = \{t \mid t' \in T,\ t \in \hat{\pi}(\theta)(t')\}$$

For example, let $\pi(f) = 0$ and $\theta(x) = f(a,b)$. Then, it follows that $\hat{\pi}(f(a,b)) = \{a, b\}$ and $\hat{\pi}(\theta)(g(x,x)) = \hat{\pi}(g(f(a,b), f(a,b))) = \{g(a,a), g(a,b), g(b,a), g(b,b)\}$.

Lemma 4.17 $s \underset{AC}{\sim} t \Rightarrow \hat{\pi}(s) =_{AC} \hat{\pi}(t)$

Proof. See Appendix A.1. □

Lemma 4.18 $\hat{\pi}(\theta)(\hat{\pi}(t)) = \hat{\pi}(t\theta)$

Proof. See Appendix A.2. □

Definition 4.19 Let $>$ be an AC-reduction order. We define $\ggcurly_{AC}$ by the AC-multiset extension of $\gtrsim_{AC}$, $\ggcurly^{sub}_{AC}$ by the AC-multiset extension of $\gtrsim^{sub}_{AC}$, and $\gg^{sub}_{AC}$ by the strict part of $\ggcurly^{sub}_{AC}$.

Note that for any AC-reduction order $>$ both the equivalence parts of $\gtrsim_{AC}$ and $\gtrsim^{sub}_{AC}$ are equal to $\underset{AC}{\sim}$, and both the equivalence parts of $\ggcurly_{AC}$ and $\ggcurly^{sub}_{AC}$ are equal to $=_{AC}$.

Definition 4.20 We define $s \gtrsim^{mul}_{\pi} t$ by $\hat{\pi}(s) \ggcurly_A C\hat{\pi}(t)$, and $s >^{mul}_{\pi} t$ by $\hat{\pi}(s) \gg^{sub}_{AC} \hat{\pi}(t)$.

Theorem 4.21 If $>$ is an AC-reduction order and π is an argument filtering function with AC-condition then $(\gtrsim^{mul}_{\pi}, >^{mul}_{\pi})$ satisfies the conditions of the weak AC-reduction pair except for the stability.

Proof.
- (The AC-compatibility of $\gtrsim^{mul}_{\pi}$): Let $s \underset{AC}{\sim} t$. From lemma 4.17, $\hat{\pi}(s) =_{AC} \hat{\pi}(t)$. Hence it follows that $s \gtrsim^{mul}_{\pi} t$.
- (The monotonicity of $\gtrsim^{mul}_{\pi}$): Let $s \gtrsim^{mul}_{\pi} t$. We prove the claim by induction on C. It suffices to show the case $C \equiv f(\ldots, t_{i-1}, \Box, t_{i+1}, \ldots)$. In the case $\pi(f) = j\ (\neq 0)$, if $j \neq i$ then $\hat{\pi}(C[s]) \ggcurly_{AC} \hat{\pi}(C[t])$ is trivial, otherwise $\hat{\pi}(C[s]) = \hat{\pi}(s) \ggcurly_{AC} \hat{\pi}(t) = \hat{\pi}(C[t])$.

In the case $\pi(f)=0$, $\hat{\pi}(C[s]) = \bigcup_{i\neq j}\hat{\pi}(t_i)\cup\hat{\pi}(s) \geqq_{AC} \bigcup_{i\neq j}\hat{\pi}(t_i)\cup\hat{\pi}(t) = \hat{\pi}(C[t])$. In the case $\pi(f)=[i_1,\ldots,i_m]$, if $i\notin\pi(f)$ then it is trivial. Suppose that $i\in[i_1,\ldots,i_m]=\pi(f)$. We denote $\hat{\pi}(C)=\{\hat{C}_1,\ldots,\hat{C}_p\}$, $\hat{\pi}(s)=\{\hat{s}_1,\ldots,\hat{s}_q\}$ and $\hat{\pi}(t)=\{\hat{t}_1,\ldots,\hat{t}_r\}$. Then it is obvious that $\hat{\pi}(C[s])=\{\hat{C}_i[\hat{s}_j]\mid 1\le i\le p,\ 1\le j\le q\}$ and $\hat{\pi}(C[t])=\{\hat{C}_i[\hat{t}_j]\mid 1\le i\le p,\ 1\le j\le r\}$. For any i, from $\{\hat{s}_1,\ldots,\hat{s}_q\}\geqq_{AC}\{\hat{t}_1,\ldots,\hat{t}_r\}$, it follows that $\{\hat{C}_i[\hat{s}_1],\ldots,\hat{C}_i[\hat{s}_q]\}\geqq_{AC}\{\hat{C}_i[\hat{t}_1],\ldots,\hat{C}_i[\hat{t}_r]\}$, because both $\underset{AC}{\sim}$ and $\gtrsim_{AC}$ $(= > \cdot \underset{AC}{\sim})$ have the monotonicity. Hence, it follows that $\hat{\pi}(C[s])\geqq_{AC}\hat{\pi}(C[t])$.

- (The well-foundedness of $>^{mul}_{\pi}$):

From lemma 4.5, $\gtrsim^{sub}_{AC}$ is well-founded. Hence, $\gg^{sub}_{AC}$ is well-founded by proposition 4.15. Therefore, $>^{mul}_{\pi}$ is well-founded.

- ($\gtrsim^{mul}_{\pi}\cdot>^{mul}_{\pi}\subseteq>^{mul}_{\pi}$):

Let $\hat{\pi}(t_0)\geqq_{AC}\hat{\pi}(t_1)\gg^{sub}_{AC}\hat{\pi}(t_2)$. In the case $\hat{\pi}(t_0)=_{AC}\hat{\pi}(t_1)$ it is trivial that $\hat{\pi}(t_0)\gg^{sub}_{AC}\hat{\pi}(t_2)$. Suppose that $\hat{\pi}(t_0)\neq_{AC}\hat{\pi}(t_1)$. From $\gtrsim_{AC}\subseteq\gtrsim^{sub}_{AC}$, it follows that $\hat{\pi}(t_0)\gg^{sub}_{AC}\hat{\pi}(t_1)$. Since $\gg^{sub}_{AC}$ is transitive by proposition 4.15, it follows that $\hat{\pi}(t_0)\gg^{sub}_{AC}\hat{\pi}(t_2)$. □

Unfortunately, both $\gtrsim^{mul}_{\pi}$ and $>^{mul}_{\pi}$ are not stable. For example, let $s\equiv h(x)$, $t\equiv g(x,x)$, $\pi(f)=0$ and $\theta=\{x:=f(y,z)\}$. Using the order $>^{flat}_{rpo}$ with the precedence $h\rhd g$ as an AC-reduction order, we trivially obtain $s>^{mul}_{\pi}t$. However, since $\hat{\pi}(s\theta)=\{h(y),h(z)\}$ and $\hat{\pi}(t\theta)=\{g(y,y),g(y,z),g(z,y),g(z,z)\}$, it follows that $s\theta\not>^{mul}_{\pi}t\theta$. Hence, we need a suitable restriction to assure the stability of $\gtrsim^{mul}_{\pi}$ and $>^{mul}_{\pi}$.

On the other hand, in general, for any t, $\hat{t}\in\hat{\pi}(t)$ and θ, we have

$$\hat{\pi}(\theta)(\hat{t})\supseteq\{\hat{t}\theta_1,\ldots,\hat{t}\theta_n\}$$

for $\theta_1,\ldots,\theta_n$ such that $\forall x\in Var(t).\ x\theta_i\in\hat{\pi}(x\theta)$ and $\forall x\notin Var(s).\ x\theta_i\equiv x$. Moreover, if t is linear then the equivalence holds, i.e.,

$$\hat{\pi}(\theta)(\hat{t})=\{\hat{t}\theta_1,\ldots,\hat{t}\theta_n\}.$$

In the previous example, letting $\hat{s}\equiv h(x)\in\hat{\pi}(s)$ and $\hat{t}\equiv g(x,x)\in\hat{\pi}(t)$, it follows that

$$\hat{\pi}(\theta)(\hat{s})=\{\hat{s}\theta_1,\hat{s}\theta_2\}\text{ and }\hat{\pi}(\theta)(\hat{t})\supseteq\{\hat{t}\theta_1,\hat{t}\theta_2\}$$

where $\theta_1=\{x:=y\}$ and $\theta_2=\{x:=z\}$. Using this fact we prove the following lemma.

Lemma 4.22 Let s and t be terms such that $\hat{t}$ is linear for all $[\![\hat{t}]\!]\in\hat{\pi}(t)_{AC}-\hat{\pi}(s)_{AC}$. Then $s\gtrsim^{mul}_{\pi}t\Rightarrow s\theta\gtrsim^{mul}_{\pi}t\theta$ and $s>^{mul}_{\pi}t\Rightarrow s\theta>^{mul}_{\pi}t\theta$.

Proof. ($\gtrsim^{mul}_{\pi}$): It suffices to show that $\hat{\pi}(\theta)(\hat{s})\geqq_{AC}\cup_i\hat{\pi}(\theta)(\hat{t}_i)$ for any $[\![\hat{s}]\!]\in\hat{\pi}(s)_{AC}-\hat{\pi}(t)_{AC}$ and $[\![\hat{t}_i]\!]\in\hat{\pi}(t)_{AC}-\hat{\pi}(s)_{AC}$ $(1\le i\le n)$ such that $\hat{s}\gtrsim_{AC}\hat{t}_i$. We suppose $\{\theta_1,\ldots,\theta_m\}$ constructed by each substitution θ_i satisfying $\forall x\in Var(s).\ x\theta_i\in\hat{\pi}(x\theta)$ and $\forall x\notin Var(s).\ x\theta_i\equiv x$. Then the following inclusion holds:

$$\hat{\pi}(\theta)(\hat{s})\supseteq\{\hat{s}\theta_j\mid 1\le j\le m\}.$$

We suppose $\{\theta^i_1,\ldots,\theta^i_{m_i}\}$ constructed by each substitution θ^i_j satisfying $\forall x\in Var(t_i).x\theta^i_j\in\hat{\pi}(x\theta)$ and $\forall x\notin Var(t_i).x\theta_i\equiv x$. From the linearity of $\hat{t}_i$ the following equation holds:

$$\hat{\pi}(\theta)(\hat{t}_i)=\{\hat{t}_i\theta^i_j\mid 1\le j\le m_i\}.$$

Since $\hat{s}\gtrsim_{AC}\hat{t}_i$, it follows that $Var(\hat{s})\supseteq Var(\hat{t}_i)$. Thus, $\{\theta^i_1,\ldots,\theta^i_{m_i}\}\subseteq\{\theta_1,\ldots,\theta_m\}$. Hence, the following inclusion holds:

$$\cup_i\hat{\pi}(\theta)(\hat{t}_i)\subseteq\{\hat{t}_i\theta_j\mid 1\le i\le n,\ 1\le j\le m\}.$$

From the stability of $\gtrsim_{AC}$, it follows that $\hat{s}\theta_j\gtrsim_{AC}\hat{t}_i\theta_j$ for any j. Therefore, it follows that $\hat{\pi}(\theta)(\hat{s})\geqq_{AC}\{\hat{s}\theta_j\mid 1\le j\le m\}\geqq_{AC}\{\hat{t}_i\theta_j\mid 1\le i\le n,\ 1\le j\le m\}\geqq_{AC}\cup_i\hat{\pi}(\theta)(\hat{t}_i)$.

($>^{mul}_{\pi}$): It suffices to show that $\hat{\pi}(\theta)(\hat{s})\gg^{sub}_{AC}\cup_i\hat{\pi}(\theta)(\hat{t}_i)$ for any $[\![\hat{s}]\!]\in\hat{\pi}(s)_{AC}-\hat{\pi}(t)_{AC}$ and $[\![\hat{t}_i]\!]\in\hat{\pi}(t)_{AC}-\hat{\pi}(s)_{AC}$ such that $\hat{s}\gtrsim^{sub}_{AC}\hat{t}_i$. Thanks to the stability of $\gtrsim^{sub}_{AC}$, as similar to the proof for $\gtrsim^{mul}_{\pi}$, it follows that $\hat{\pi}(\theta)(\hat{s})\gg^{sub}_{AC}\cup_i\hat{\pi}(\theta)(\hat{t}_i)$. □

Lemma 4.23 $\trianglerighteq_{hd}\subseteq\gtrsim^{mul}_{\pi}\cup>^{mul}_{\pi}$.

Proof. Let $s\trianglerighteq_{hd}t$. From the definition, there is a context C such that $s\underset{AC}{\sim}C[t]_p$ and $(C)_q=(t)_\varepsilon\in\Sigma^{\#}_{AC}$ for all $q\prec p$. In the case $C\equiv\square$, $\hat{\pi}(s)=_{AC}\hat{\pi}(t)$ by lemma 4.17. Thus $s\gtrsim^{mul}_{\pi}t$. Suppose that $C\not\equiv\square$. From the definition, there is some term t' and AC-symbol f such that $s\underset{AC}{\sim}C[t]\underset{AC}{\sim}f(t',t)$ and $f=(t)_\varepsilon=(C)_\varepsilon$. From lemma 4.17, $\hat{\pi}(s)=_{AC}\hat{\pi}(f(t',t))$. If $\pi(f)=[]$ then $\hat{\pi}(f(t',t))=\{f\}=\hat{\pi}(t)$. Hence, it follows that $f(t',t)\gtrsim^{mul}_{\pi}t$. If $\pi(f)=[1,2]$ then $\hat{\pi}(s)=_{AC}\hat{\pi}(f(t',t))=\{f(v',v)\mid v'\in\hat{\pi}(t'),\ v\in\hat{\pi}(t)\}\gg^{sub}_{AC}\{v\mid v\in\hat{\pi}(t)\}=\hat{\pi}(t)$. Hence, it follows that $s>^{mul}_{\pi}t$. If $\pi(f)=0$ then $\hat{\pi}(s)=_{AC}\hat{\pi}(f(t',t))=\hat{\pi}(t')\cup\hat{\pi}(t)\geqq_{AC}\hat{\pi}(t)$. Hence, it follows that $s\gtrsim^{mul}_{\pi}t$. □

Theorem 4.24 Let R be an AC-TRS. If there exists an AC-reduction order $>$ and an argument filtering function π with the AC-condition such that

- $f^{\#}$ is identified to f or $\pi(f^{\#})=[]$ for all AC-symbols f,
- $\hat{r}$ is linear

for all $l \to r \in R$ and $[\![\hat{r}]\!] \in \hat{\pi}(r)_{AC} - \hat{\pi}(l)_{AC}$,

- $l \gtrsim_{\pi}^{mul} r$ for all $l \to r \in R$, and
- $u^{\#} >_{\pi}^{mul} v^{\#}$ for all $\langle u^{\#}, v^{\#} \rangle \in DP_{AC}^{\#}(R)$,

then R is AC-terminating.

Proof. Same as the proof of the case ($4 \Rightarrow 1$) in theorem 3.13, use of theorem 4.21 and lemmas 4.22 and 4.23. □

Note that in similar to the proof of theorem 4.21, it can be proved that for any given AC-reduction order $>$, $\gtrsim_{\pi}^{mul}$ is a weak AC-reduction order except for the stability. Under the condition of lemma 4.22, the strict part $\gtrsim_{\pi}^{mul}$ of $\gtrsim_{\pi}^{mul}$ is stable.

In order to show the usefulness of AC-multiset extension, we prove the AC-termination of an AC-TRS to which not only traditional techniques but also single argument filtering function and lexicographic argument filtering method cannot be applied.

Example 4.25 As an AC-reduction order $>$, we use the order $>_{rpo}^{flat}$. Consider the following AC-TRS R_6 with $\Sigma_{AC}^{\#} = \{f\}$.

$$R_6 = \begin{cases} g(0, f(x,x)) & \to & x \\ g(x, s(y)) & \to & g(f(x,y), 0) \\ g(s(x), y) & \to & g(f(x,y), 0) \\ g(f(x,y), 0) & \to & f(g(x,0), g(y,0)) \end{cases}$$

$$DP_{AC}^{\#}(R_6) = \begin{cases} \langle g^{\#}(x, s(y)), g^{\#}(f(x,y), 0) \rangle \\ \langle g^{\#}(s(x), y), g^{\#}(f(x,y), 0) \rangle \\ \langle g^{\#}(f(x,y), 0), g^{\#}(x, 0) \rangle \\ \langle g^{\#}(f(x,y), 0), g^{\#}(y, 0) \rangle \end{cases}$$

Let $\pi(f) = 0$ and $s \rhd 0$. Then, $l \gtrsim_{\pi}^{mul} r$ for all $l \to r \in R_6$, and $u^{\#} >_{\pi}^{mul} v^{\#}$ for all $\langle u^{\#}, v^{\#} \rangle \in DP_{AC}^{\#}(R_6)$. From theorem 4.24, R_6 is AC-terminating.

Note that the above argument filtering method by AC-multisets also proves termination of TRS R_6 without AC-function symbols, i.e., $\Sigma_{AC} = \emptyset$.

5. Distribution Elimination

Transformation methods, which transform a given TRS into a TRS whose termination is easier to prove than the original one, have been proposed. Most famous methods are so-called elimination methods. In 12), we showed that the argument filtering method combined with dependency pair technique gives a uniform framework why various elimination methods work well; however, the only exception was the distribution elimination transformation. In this section, using AC-multiset extension of the argument filtering method, we analyze the distribution elimination transformation.

Definition 5.1 (Distribution Elimination) Let e be a symbol, called eliminated symbol. A rule $l \to r$ is a distribution rule for e if $l \equiv C[e(x_1, \ldots, x_n)]$ and $r \equiv e(C[x_1], \ldots, C[x_n])$ for some non-empty context C in which e does not occur and pairwise different variables $x_1, \ldots, x_n$. The distribution elimination (DIS_e) is defined as follows:

$$E_e(t) = \begin{cases} \{t\} \quad \text{if} \quad t \in \mathcal{V} \\ \bigcup_{i=1}^{n} E_e(t_i) \quad \text{if} \quad t \equiv e(t_1, \ldots, t_n) \\ \{f(s_1, \ldots, s_n) \mid s_i \in E_e(t_i)\} \\ \qquad \text{if} \quad t \equiv f(t_1, \ldots, t_n) \ \text{with} \ f \neq e \end{cases}$$

$DIS_e(R) = \{l \to r' \mid l \to r \in R$ is not a distribution rule for e, $r' \in E_e(r)\}$

For example, let $t \equiv e(0, g(1, e(2,3)))$, then $E_e(t) = \{0,\ g(1,2),\ g(1,3)\}$.

Proposition 5.2 [17] Suppose that each rule $l \to r \in R$ is a distribution rule or a rule in which the eliminated symbol e does not occur in l. If $DIS_e(R)$ is terminating and right-linear then R is terminating.

Here, $DIS_e(R)$ is right-linear means that r is a linear term for any $l \to r \in DIS_e(R)$.

Moreover, this result was extended to AC-TRSs.

Proposition 5.3 [15] Suppose that each rule $l \to r \in R$ is a distribution rule or a rule in which the eliminated symbol e does not occur in l. If $DIS_e(R)$ is terminating, right-linear and e is only AC-symbol (i.e. $\Sigma_{AC} = \{e\}$) then R is AC-terminating.

Finally, we prove the following theorem that includes propositions 5.2 and 5.3 as special cases, i.e., $\Sigma_{AC} = \emptyset$ and $\Sigma_{AC} = \{e\}$, respectively.

Theorem 5.4 Suppose that each rule $l \to r \in R$ is a distribution rule or a rule in which the eliminated symbol e does not occur in l. If $DIS_e(R)$ is AC-terminating and right-linear then R is AC-terminating.

Proof. Let R_D be the AC-TRS constructed by all distribution rules in R, and $R_0 = R \backslash R_D$. Let $>$ be $\xrightarrow[DIS_e(R)/AC]{+}$. Since $DIS_e(R)$ is AC-terminating, $>$ is an AC-reduction order. We denote DP_0 dependency pairs constructed from R_0, DP_0^{ex} extended dependency pairs constructed from R_0, DP_D dependency pairs constructed from R_D, and DP_D^{ex} extended dependency pairs constructed from R_D.

- $arity(e) = 1$: We choose $\pi(e) = 1$.

It is trivial that $\pi(R_0) = DIS_e(R)$ and $\pi(l) \equiv \pi(r)$ for all $l \to r \in R_D$. Thus, if R is not AC-terminating then there exists an infinite reduction $t_1 \xrightarrow[R_D]{} t_2 \xrightarrow[R_D]{} t_3 \xrightarrow[R_D]{} \cdots$. However, R_D is trivially AC-terminating. It is a contradiction.

- $arity(e) > 1$: We choose $\pi(e) = 0$.

First, we investigate non distribution rules. It is obvious that $\hat{\pi}(l) = \{l\}$ and $l \to r' \in DIS_e(R)$ for any $l \to r \in R_0$ and $r' \in \hat{\pi}(r)$. Thus, $l \gtrsim_\pi^{mul} r$ and $l >_\pi^{mul} r$ for any rule $l \to r \in R_0$. It follows that $s \xrightarrow[R_0]{} t$ implies $s \gtrsim_\pi^{mul} t$, because $\gtrsim_\pi^{mul}$ is monotonic and stable by the right-linearity of $DIS_e(R)$. Moreover, it follows that $u\theta >_\pi^{mul} v\theta$ for all θ and $\langle u, v \rangle \in DP_0^{ex}$.

On the other hand, it is obvious that for any $\langle u, v \rangle \in DP_0$ and $v' \in \hat{\pi}(v)$, $\hat{\pi}(u) = \{u\}$ and $u \to C[v'] \in DIS_e(R)$ for some C. Thus, $u >_\pi^{mul} v$ for any $\langle u, v \rangle \in DP_0$. Since $DIS_e(R)$ is right-linear, it follows that $u\theta >_\pi^{mul} v\theta$ for all θ and $\langle u, v \rangle \in DP_0$.

Next, we focus on the distribution rules. For any distribution rule $C[e(x_1, \ldots, x_n)] \to e(C[x_1], \ldots, C[x_n]) \in R_D$, $\hat{\pi}(C[e(x_1, \ldots, x_n)]) = \{C[x_1], \ldots, C[x_n]\} = \hat{\pi}(e(C[x_1], \ldots, C[x_n]))$. Thus, $s \xrightarrow[R_D]{*} t$ implies $s \gtrsim_\pi^{mul} t$. Moreover, $f(l, z)\theta \gtrsim_\pi^{mul} f(r, z)\theta$ for any $\langle f(l, z), f(r, z) \rangle \in DP_D^{ex}$ and θ. The dependency pair in DP_D can be denoted by $\langle C[e(x_1, \ldots, x_n)], v \rangle$ such that $C'[v] \equiv C[x_i]$ for some i and C'. Since $\hat{\pi}(C[e(x_1, \ldots, x_n)]) = \{C[x_1], \ldots, C[x_n]\} \gg_{AC} \{C[x_i]\} = \{C'[v]\}$, it follows that $C[e(x_1, \ldots, x_n)]\theta >_\pi^{mul} v\theta$ for any θ.

Finally, we prove this theorem based on the above properties. Assume that R is not AC-terminating. From proposition 3.7, there exists an infinite AC-dependency chain $\langle u_0^\#, v_0^\# \rangle \langle u_1^\#, v_1^\# \rangle \cdots$ and θ such that $(v_i\theta)^\# \xrightarrow[\#]{*} t_i^\# \trianglerighteq_{hd} (u_{i+1}\theta)^\#$ for some t_i ($i = 1, 2, \ldots$). By supposing that $f^\#$ is identified to f, $v_i\theta \xrightarrow[R/AC]{*} t_i \trianglerighteq_{hd} u_{i+1}\theta$ for all i. We have already proved that $v_i\theta \gtrsim_\pi^{mul} t_i$, $u_i\theta \gtrsim_\pi^{mul} v_i\theta$ for all $\langle u_i, v_i \rangle \in DP_D^{ex}$, and $u_i\theta >_\pi^{mul} v_i\theta$ for all $\langle u_i, v_i \rangle \in DP_{AC}(R) \backslash DP_D^{ex}$. From lemma 4.23 and $\gtrsim_\pi^{mul} \subseteq >_\pi^{mul}$, it follows that $t_i \underset{AC}{\sim} u_{i+1}\theta$ or $t_i >_\pi^{mul} u_{i+1}\theta$. Hence there exists a number m such that $\langle u_i, v_i \rangle \in DP_D^{ex}$ for all $i \geq m$. Moreover, there exists an infinite AC-reduction sequence $f(l_0, z_0)\theta \xrightarrow[R_D]{} f(r_0, z_0)\theta \xrightarrow[R_D]{*} f(l_1, z_1) \xrightarrow[R_D]{} f(r_1, z_1)\theta \cdots$ for some $l_i \to r_i \in R_D$ ($i = 0, 1, \ldots$). However, R_D is trivially AC-terminating. It is a contradiction. □

Acknowledgments We would like to thank Masaki Nakamura for his useful discussions, Taro Suzuki and anonymous referees for their useful comments.

References

1) Arts, T.: *Automatically Proving Termination and Innermost Normalization of Term Rewriting Systems*, PhD Thesis, Univ. of Utrecht (1997).
2) Arts, T. and Giesl, J.: Termination of Term Rewriting Using Dependency Pairs, to appear in Theoretical Computer Science.
3) Arts, T. and Giesl, J.: Automatically Proving Termination Where Simplification Orderings Fail, *TAPSOFT'97*, LNCS, Vol. 1214, pp. 261–272 (1997).
4) Arts, T. and Giesl, J.: Proving Innermost Normalization Automatically, *RTA'97*, LNCS, Vol. 1232, pp. 157–171 (1997).
5) Baader, F. and Nipkow, T.: *Term Rewriting and All That*, Cambridge University Press (1998).
6) Bachmair, T. and Plaisted, D. A.: Termination Orderings for Associative-Commutative Rewriting Systems, *J. Symbolic Computation*, Vol. 1, pp. 329–349 (1985).
7) Dershowitz, N.: Orderings for Term-rewriting Systems, *Theoretical Computer Science*, Vol.17, pp. 279–301 (1982).
8) Dershowitz, N. and Jouannaud, J.-P.: Rewrite Systems, *Handbook of Theoretical Computer Science, Vol. B* (van Leeuwen, J.(ed.)), North-Holland, pp. 243–320 (1990).
9) Ferreira, M.: *Termination of Term Rewriting, Well-foundedness, Totality and Transformations*, PhD Thesis, Utrecht University (1995).
10) Giesl, J. and Ohlebusch, E.: Pushing the Frontiers of Combining Rewrite Systems Farther Outwards, In Proceedings of the Second International Workshop on Frontiers of Combining Systems, FroCos'98, Applied Logic Series, Amsterdam,The Netherlands (1998).

11) Klop, J. W.: Term Rewriting Systems, *Handbook of Logic in Computer Science II*, Oxford University Press, pp. 1–112 (1992).
12) Kusakari, K., Nakamura, M. and Toyama, Y.: Argument Filtering Transformation, *PPDP'99*, LNCS, Vol. 1702, pp. 47–61 (1999).
13) Kusakari, K. and Toyama, Y.: On Proving AC-Termination by AC-Dependency Pairs, Research Report IS-RR-98-0026F, School of Information Science, JAIST (1998).
14) Marché, C. and Urbain, X.: Termination of Associative-Commutative Rewriting by Dependency Pairs, *RTA'98*, LNCS, Vol.1379, pp.241–255 (1998).
15) Ohsaki, H., Middeldorp, A. and Giesl, J.: Equational Termination by Semantic Labeling, Technical Report TR-98-29, Electrotechnical Laboratory (1998).
16) Rubio, A.: A Fully Syntactic AC-RPO, *RTA'99*, LNCS, Vol. 1631, pp. 133–147 (1999).
17) Zantema, H.: Termination of Term Rewriting, Interpretation and Type Elimination, *Journal of Symbolic Computation*, Vol. 17, pp. 23–50 (1994).

Appendix

A.1 Proof of lemma 4.17

Lemma 4.17 $s \underset{AC}{\sim} t \Rightarrow \hat{\pi}(s) =_{AC} \hat{\pi}(t)$

Proof. We prove the claim by induction on s. The case $s \equiv x \in \mathcal{V}$ is trivial, because $t \equiv x \equiv s$ by $s \underset{AC}{\sim} t$. Suppose that $s \equiv f(s_1, \ldots, s_n)$. Because of $s \underset{AC}{\sim} t$, the root symbol of t is f. Thus we denote $t \equiv f(t_1, \ldots, t_n)$. If $s_i \underset{AC}{\sim} t_i$ for all i then $\hat{\pi}(s_i) =_{AC} \hat{\pi}(t_i)$ for all i by induction hypothesis. Hence it follows that $\hat{\pi}(s) =_{AC} \hat{\pi}(t)$. On the other hand, since $=_{AC}$ is an equivalence relation, it suffices to show $\hat{\pi}(s) =_{AC} \hat{\pi}(t)$ for the cases $f \in \Sigma_{AC}$ and either $s \equiv f(s_1, s_2) \wedge t \equiv f(s_2, s_1)$ or $s \equiv f(f(s_{11}, s_{12}), s_2) \wedge t \equiv f(s_{11}, f(s_{12}, s_2))$. We have the following three cases.

- $\pi(f) = 0$:
$\hat{\pi}(f(s_1, s_2)) = \hat{\pi}(s_1) \cup \hat{\pi}(s_2) = \hat{\pi}(f(s_2, s_1))$, and $\hat{\pi}(f(f(s_{11}, s_{12}), s_2)) = \hat{\pi}(f(s_{11}, s_{12})) \cup \hat{\pi}(s_2) = \hat{\pi}(s_{11}) \cup \hat{\pi}(s_{12}) \cup \hat{\pi}(s_2) = \hat{\pi}(s_{11}) \cup \hat{\pi}(f(s_{12}, s_2)) = \hat{\pi}(f(s_{11}, f(s_{12}, s_2)))$.
- $\pi(f) = []$:
$\hat{\pi}(f(s_1, s_2)) = \{f\} = \hat{\pi}(f(s_2, s_1))$, and $\hat{\pi}(f(f(s_{11}, s_{12}), s_2)) = \{f\} = \hat{\pi}(f(s_{11}, f(s_{12}, s_2)))$.
- $\pi(f) = [1, 2]$:
$\hat{\pi}(f(s_1, s_2)) = \{f(\hat{s}_1, \hat{s}_2) \mid \hat{s}_i \in \hat{\pi}(s_i)\} =_{AC} \{f(\hat{s}_2, \hat{s}_1) \mid \hat{s}_i \in \hat{\pi}(s_i)\} = \hat{\pi}(f(s_2, s_1))$, and $\hat{\pi}(f(f(s_{11}, s_{12}), s_2)) = \{f(\hat{s}_3, \hat{s}_2) \mid \hat{s}_3 \in \hat{\pi}(f(s_{11}, s_{12})),\ \hat{s}_2 \in \hat{\pi}(s_2)\} = \{f(f(\hat{s}_{11}, \hat{s}_{12}), \hat{s}_2) \mid \hat{s}_{11} \in \hat{\pi}(s_{11}),\ \hat{s}_{12} \in \hat{\pi}(s_{12}),\ \hat{s}_2 \in \hat{\pi}(s_2)\} =_{AC} \{f(\hat{s}_{11}, f(\hat{s}_{12}, \hat{s}_2)) \mid \hat{s}_{11} \in \hat{\pi}(s_{11}),\ \hat{s}_{12} \in \hat{\pi}(s_{12}),\ \hat{s}_2 \in \hat{\pi}(s_2)\} = \{f(\hat{s}_{11}, \hat{s}_4) \mid \hat{s}_{11} \in \hat{\pi}(s_{11}),\ \hat{s}_4 \in \hat{\pi}(f(s_{12}, s_2))\} = \hat{\pi}(f(s_{11}, f(s_{12}, s_2)))$. □

A.2 Proof of Lemma 4.18

Lemma 4.18 $\hat{\pi}(\theta)(\hat{\pi}(t)) = \hat{\pi}(t\theta)$

Proof. We prove the claim by induction on t. In the case $t \equiv x \in \mathcal{V}$, $\hat{\pi}(\theta)(\hat{\pi}(x)) = \hat{\pi}(\theta)(\{x\}) = \hat{\pi}(x\theta)$. Suppose that $t \equiv f(t_1, \ldots, t_n)$. We have the following three cases.

- $\pi(f) = j\ (\neq 0)$:

$$\begin{aligned}
&\hat{\pi}(\theta)(\hat{\pi}(f(t_1, \ldots, t_n)))\\
&= \hat{\pi}(\theta)(\hat{\pi}(t_j))\\
&= \hat{\pi}(t_j\theta)\\
&= \hat{\pi}(f(t_1\theta, \ldots, t_j\theta, \ldots, t_n\theta))\\
&= \hat{\pi}(f(t_1, \ldots, t_j, \ldots, t_n)\theta)
\end{aligned}$$

- $\pi(f) = 0$:

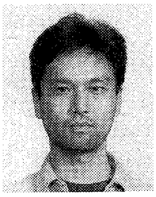
$$\begin{aligned}
&\hat{\pi}(\theta)(\hat{\pi}(f(t_1, \ldots, t_n)))\\
&= \hat{\pi}(\theta)(\hat{\pi}(t_1) \cup \cdots \cup \hat{\pi}(t_n))\\
&= \hat{\pi}(\theta)(\hat{\pi}(t_1)) \cup \cdots \cup \hat{\pi}(\theta)(\hat{\pi}(t_n))\\
&= \hat{\pi}(t_1\theta) \cup \cdots \cup \hat{\pi}(t_n\theta)\\
&= \hat{\pi}(f(t_1\theta, \ldots, t_n\theta))\\
&= \hat{\pi}(f(t_1, \ldots, t_n)\theta)
\end{aligned}$$

- $\pi(f) = [i_1, \ldots, i_m]$:

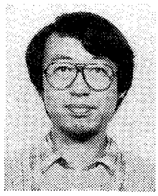
$$\begin{aligned}
&\hat{\pi}(\theta)(\hat{\pi}(f(\vec{t_i})))\\
&= \hat{\pi}(\theta)(\{f(\vec{t'_{i_j}}) \mid t'_{i_j} \in \hat{\pi}(t_{i_j})\})\\
&= \{t'' \mid t' \in T',\ t'' \in \hat{\pi}(\theta)(t')\}\\
&\quad \text{where } T' = \{f(\vec{t'_{i_j}}) \mid t'_{i_j} \in \hat{\pi}(t_{i_j})\}\\
&= \{t'' \mid t'_{i_j} \in \hat{\pi}(t_{i_j}),\ t'' \in T''\}\\
&\quad \text{where } T'' = \{f(\vec{t''_{i_j}}) \mid t''_{i_j} \in \hat{\pi}(\theta)(t'_{i_j})\}\\
&= \{f(\vec{t''_{i_j}}) \mid t'_{i_j} \in \hat{\pi}(t_{i_j}),\ t''_{i_j} \in \hat{\pi}(\theta)(t'_{i_j})\}\\
&= \{f(\vec{t''_{i_j}}) \mid t''_{i_j} \in T''\}\\
&\quad \text{where}\\
&\quad T'' = \{t'''_{i_j} \mid t'_{i_j} \in \hat{\pi}(t_{i_j}),\ t'''_{i_j} \in \hat{\pi}(\theta)(t'_{i_j})\}\\
&= \{f(\vec{t''_{i_j}}) \mid t''_{i_j} \in \hat{\pi}(\theta)(\hat{\pi}(t_{i_j}))\}\\
&= \{f(\vec{t''_{i_j}}) \mid t''_{i_j} \in \hat{\pi}(t_{i_j}\theta)\}\\
&= \hat{\pi}(f(t_1\theta, \ldots, t_n\theta))\\
&= \hat{\pi}(f(t_1, \ldots, t_n)\theta)
\end{aligned}$$

□

(Received October 15, 1999)
(Accepted February 22, 2000)



Keiichirou Kusakari was born in 1969. He received B.E. Tokyo Institute of Technology in 1994, M.E. from Japan Advanced Institute of Science and Technology (JAIST) in 1996. He is now a doctoral student of the School of Information Science in JAIST. His research interests include term rewriting systems, program theory, and automated theorem proving. He is a member of JSSST.



Yoshihito Toyama was born in 1952. He received B.E. from Niigata University in 1975, M.E. and D.E. from Tohoku University in 1977 and 1990, respectively. From 1977 to 1993, he worked at NTT Laboratories. Since 1993, he is a professor of School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). His research interests include term rewriting systems, program theory, and automated theorem proving. He is a member of IEICE, IPSJ, JSSST, ACM, and EATCS.