Proving Termination of AC-rewriting without Extended Rules

Cristina Borralleras[1] and Albert Rubio[2]

[1] Universitat de Vic, Spain
Email: cristina.borralleras@uvic.es
[2] Universitat Politècnica de Catalunya, Barcelona, SPAIN
Email: rubio@lsi.upc.es

Abstract. Polynomial interpretations and RPO-like orderings allow one to prove termination of Associative and Commutative (AC-)rewriting by only checking the rules of the given rewrite system without considering the so-called extended rules. However, these methods have important limitations as termination proving tools.

To overcome these limitations, more powerful methods like the dependency pairs method have been extended to the AC-case. Unfortunately, in order to ensure AC-termination, the extended rules, which, in general, are hard to prove, must be added to the rewrite system.

In this paper we present a new automatable ordering-based termination proving method for AC-rewriting which does not need to consider extended rules. Due to this, we can easily prove several non-trivial examples appearing in the literature that, to our knowledge, can be handled by no other automatic method.

1 Introduction

In programming, as well as in theorem proving, it is very common to have binary operators which satisfy the associative and commutative properties. However, such axioms cannot be treated as additional rules in the rewrite system (e.g., commutativity cannot be oriented by any well-founded ordering), and hence they require a special treatment. The most common approach to deal with AC symbols is rewriting *modulo* AC, that is, rewriting using matching modulo associativity and commutativity to detect the applicability of the rules. The following example (taken from [MU98]) describes addition ($+$) and multiplication ($*$) for natural numbers in binary notation defined by a constant $\#$, denoting the empty sequence of digits, and two unary postfixed functions $(_)0$ and $(_)1$, to add 0's and 1's to the right. For instance, 5 is written as $(((\#)1)0)1$.

Example 1. In this example both $+$ and $*$ are AC-symbols.

$$\begin{aligned}(\#)0 &\to \#\\ x + \# &\to x\\ x * \# &\to \#\end{aligned}$$

$$\begin{aligned}(x)0 + (y)0 &\to (x + y)0\\ (x)0 + (y)1 &\to (x + y)1\\ (x)1 + (y)1 &\to (x + y + (\#)1)0\end{aligned}$$

$$\begin{aligned}x * (y)0 &\to (x * y)0\\ x * (y)1 &\to x + (x * y)0\end{aligned}$$

Due to the fact that rewriting is performed modulo the AC-axioms, proving termination becomes a more difficult task. In particular, the applied method must be AC-compatible, which roughly means that all terms in the same AC-equivalent class are treated in the same way. When using an ordering-based termination proof method, AC-compatibility means that if some term s is greater than some term t, then any term that is AC-equivalent to s is greater than any term that is AC-equivalent to t.

Many efforts have been made [BCL87,BP85,KSZ95,DP93,RN95,KS00,Rub02] in order to obtain AC-compatible simplification orderings (i.e., monotonic orderings including the subterm relation) by extending the methods used for standard rewriting.

These methods are, in general, well suited for automation. However, as for standard rewriting, such AC-compatible simplification orderings have important limitations as termination proving tools. For instance, there are many term rewrite systems (TRSs) that are terminating but are not contained in any simplification ordering, i.e., they are not *simply terminating*.

In [MU98,KT01,Urb01] the dependency pairs method [AG00], which can prove termination of TRSs that are not simply terminating, is adapted to deal with AC-rewriting, and in [GK01] it is adapted for rewriting modulo more general equational theories. All these AC-versions of the dependency pairs method need to consider the so-called *extended rules*. For the AC-case this means to add to the set of rules one extended rule $f(l, x) \to f(r, x)$, where x is a new variable, for every rule $l \to r$ with l headed by an AC-symbol f. In the previous example we have to add

$$\begin{aligned}x + \# + z &\to x + z\\ x * \# * z &\to \# * z\end{aligned}$$

$$\begin{aligned}(x)0 + (y)0 + z &\to (x + y)0 + z\\ (x)0 + (y)1 + z &\to (x + y)1 + z\\ (x)1 + (y)1 + z &\to (x + y + (\#)1)0 + z\end{aligned}$$

$$\begin{aligned}x * (y)0 * z &\to (x * y)0 * z\\ x * (y)1 * z &\to x + (x * y)0 * z\end{aligned}$$

These extended rules are needed to ensure monotonicity with respect to the AC-symbols. But adding them to the system usually makes the termination proof harder.

An alternative to the dependency pair method is the use of the monotonic semantic path ordering (MSPO; [BFR00]), which is a monotonic version of the *semantic path ordering* (SPO; [KL80]). The SPO generalizes path orderings like Dershowit's *recursive path ordering* (RPO; [Der82]) by replacing the use of a precedence by the use of any measure, defined by an *underlaying quasi-ordering*, involving the whole term and not only the head symbol.

The aim of this paper is to adapt the MSPO to deal with AC symbols, obtaining a fully monotonic AC-ordering which allows us to avoid the use of extended rules. Following the ideas used to adapt RPO to the AC-case in [Rub02], SPO has been adapted to obtain an associative and commutative SPO (ACSPO). After that, a monotonic version of ACSPO, called ACMSPO, is obtained. Even though the ACSPO and the ACRPO share the same structure, finding the appropriate formulation of the ordering was not an easy task. In order to ensure the properties of the ACSPO and the ACMSPO, some new conditions on the underlying quasi-ordering used inside SPO are needed. This conditions have to be tight, otherwise the ordering becomes too weak. Additionally, to ensure stability under substitutions it is necessary to *linearize* some variables of the terms, but, fortunately, we have shown that this is only required in a single case.

With ACMSPO we have been able to obtain a simple termination proof for the *ternary integral arithmetic* example used in [CMR97] for computing addition and multiplication of integers in balanced ternary notation, where there are three unary postfixed functions $(x)0$, $(x)1$ and $(x)j$ representing respectively $3x$, $3x+1$ and $3x-1$ (see also [Knu97], pages 207 and 208 for further details).

Example 2. Both $+$ and $*$ are AC-symbols.

$$
\begin{array}{rl}
(\#)0 & \to \# \\
x + \# & \to x \\
(x)0 + (y)0 & \to (x+y)0 \\
(x)0 + (y)1 & \to (x+y)1 \\
(x)0 + (y)j & \to (x+y)j \\
(x)1 + (y)j & \to (x+y)0 \\
(x)1 + (y)1 & \to (x+y+(\#)1)j \\
(x)j + (y)j & \to (x+y+(\#)j)1
\end{array}
\qquad
\begin{array}{rl}
x - y & \to x + opp(y) \\
opp(\#) & \to \# \\
opp((x)0) & \to (opp(x))0 \\
opp((x)1) & \to (opp(x))j \\
opp((x)j) & \to (opp(x))1 \\
x * \# & \to \# \\
x * (y)0 & \to (x*y)0 \\
x * (y)1 & \to x + (x*y)0 \\
x * (y)j & \to (x*y)0 + opp(x)
\end{array}
$$

Proving termination of this example is not trivial at all. In [CMR97] there is a quite complex ad-hoc hand-tailored proof of termination based on a lexicographic combination of three interpretations on the non-negative integers. In [MU98] an AC-version of the *dependency pairs method* [AG00] was used for proving termination of this example (together with its extended rules), but the use of some AC-marked symbols, which was necessary to deal with the example, turned out to be unsound.

As shown for MSPO [BR02], the ACMSPO method can be fully automated in a constraint solving based system (see Section 6 for details).

The paper is organized as follows. The next section is devoted to preliminaries on orderings and AC-rewriting. In Section 3, the AC semantic path ordering (ACSPO), on which we will build our method, is introduced. A monotonic version of the ACSPO is given in Section 4. In Section 5, ACMSPO is used to check the termination of several examples and in Section 6 it is shown how ACMSPO can be implemented as a fully automated termination proving tool, using a constraint framework similar to the one of the dependency pairs method. Conclusions and further work are discussed in Section 7. Finally, the proofs of the properties of the orderings are given in Section 8.

2 AC-rewriting and termination

In the following we consider that $\mathcal{F}$ is a set of function symbols, $\mathcal{F}_{AC}$ the subset containing all the AC symbols, $\mathcal{X}$ a set of variables and $\mathcal{T}(\mathcal{F},\mathcal{X})$ is the set of terms built from $\mathcal{F}$ and $\mathcal{X}$.

Rewriting modulo AC, or AC-rewriting, consists of rewriting over the equivalence classes defined by the associativity and commutativity axioms. Therefore, instead of syntactic equality, rewriting modulo AC considers $=_{AC}$, the congruence generated on $\mathcal{T}(\mathcal{F},\mathcal{X})$ by the associativity and commutativity axioms for the symbols in $\mathcal{F}_{AC}$.

Given a TRS $\mathcal{R}$, a term $s \in \mathcal{T}(\mathcal{F},\mathcal{X})$ rewrites to t with $\mathcal{R}$ modulo AC, denoted by $s \to_{\mathcal{R}/AC} t$ if $s =_{AC} s'$, $s'|_p = l\sigma$ for some rule $l \to r \in \mathcal{R}$, term s', position p and substitution σ, and $t =_{AC} s'[r\sigma]_p$.

Let s and t be arbitrary terms in $\mathcal{T}(\mathcal{F},\mathcal{X})$, let f be a function symbol in $\mathcal{F}$ and let σ be a substitution. A quasi-ordering $\succeq$ is a transitive reflexive relation. Its *strict part* $\succ$ is the strict ordering $\succeq \setminus \preceq$ (i.e, $s \succ t$ iff $s \succeq t$ and $s \not\preceq t$). Its *equivalence* $\sim$ is $\succeq \cap \preceq$. Note that $\succeq$ is the disjoint union of $\succ$ and $\sim$, and that if $=$ denotes syntactic equality then $\succ \cup =$ is a quasi-ordering whose strict part is $\succ$. A quasi-ordering $\succeq$ is *monotonic* if $s \succeq t$ implies $f(\dots s \dots) \succeq f(\dots t \dots)$, and *stable under substitutions* if $s \succeq t$ implies $s\sigma \succeq t\sigma$.

A (strict partial) ordering $\succ$ is a transitive irreflexive relation. It is *monotonic* if $s \succ t$ implies $f(\dots s \dots) \succ f(\dots t \dots)$, and *stable under substitutions* if $s \succ t$ implies $s\sigma \succ t\sigma$. Monotonic orderings that are stable under substitutions are called *rewrite orderings*. A *reduction ordering* is a rewrite ordering that is *well-founded*: there are no infinite sequences $t_1 \succ t_2 \succ \dots$

As the rewrite relation is defined over AC-equivalence classes, any ordering $\succ$ used to prove termination of AC-rewriting has to be *AC-compatible*.

Definition 1. *A relation is said to be AC-compatible if it is compatible with the* $=_{AC}$ *relation. In particular* $\succ$ *is* AC-compatible *if* $s' =_{AC} s \succ t =_{AC} t'$ *implies* $s' \succ t'$*; and* $\succeq$ *is* AC-compatible *if* $s' =_{AC} s \succeq t =_{AC} t'$ *implies* $s' \succeq t'$.

Theorem 1. *Let* $\succ$ *be an AC-compatible reduction ordering and let* $\mathcal{R}$ *be a TRS. If* $l \succ r$ *for all rules* $l \to r \in \mathcal{R}$ *then* $\mathcal{R}$ *is terminating for rewriting modulo AC.*

The AC-multiset extension of an AC-compatible ordering is defined in the standard way but using AC-equality instead of syntactic equality.

Definition 2. *The AC-multiset extension of an AC-compatible ordering $\succ$, denoted by $\gg$, is defined by:*

$M \cup S \gg M' \cup T$ *if and only if* $S \neq \emptyset$, $M ==_{AC} M'$ *and* $\forall t_i \in T\ \exists s_j \in S$ *such that* $s_j \succ t_i$;

where

$==_{AC}$ *is the standard extension of* $=_{AC}$ *to multisets, i.e.* $M ==_{AC} M'$ *iff either* $M = M' = \emptyset$ *or* $M \setminus \{s_i\} ==_{AC} M' \setminus \{t_j\}$ *and* $s_i =_{AC} t_j$.

From this definition it follows that for comparing two multisets, once the AC-equal terms $s =_{AC} t$ have been deleted, it suffices to check whether for every element in the second multiset T, there is a greater element in the first multiset S.

A very common way to easily obtain AC-compatibility is to consider the terms flattened wrt. the AC-symbols. The flat form of t, denoted by $\overline{t}$, is the normal form of t wrt. the (infinite set of) rules

$f(x_1 \ldots x_{i-1}, f(y_1 \ldots y_k), x_{i+1} \ldots x_n) \rightarrow f(x_1 \ldots x_{i-1}, y_1 \ldots y_k, x_{i+1} \ldots x_n)$

for each $f \in \mathcal{F}_{AC}$. E.g., let $t = +(g(+(a,+(b,c))),+(d,+(e,f)))$ then $\overline{t} = +(g(+(a,b,c)),d,e,f)$. Note that due to the flattening we need to consider variable arities (greater than or equal to 2) for symbols in $\mathcal{F}_{AC}$.

It follows from an easy induction that $s =_{AC} t$ iff $\overline{s}$ and $\overline{t}$ are equal up to permutation of arguments for the AC-symbols. We also denote by $=_{AC}$ this equality up to permutation of arguments for the AC symbols. From now on, all terms are assumed to be in their flattened form.

However, when considering flattened terms the orderings frequently loses monotonicity. For instance, with the *recursive path ordering* (RPO) if a symbol f is greater than a symbol g and f is an AC-symbol, we have that $f(a,a) \succ_{rpo} g(a)$ but $f(f(a,a),b) \not\succ_{rpo} f(g(a),b)$ as the flattened form of $f(f(a,a),b)$ is $f(a,a,b)$ and $f(g(a),b) \succ_{rpo} f(a,a,b)$. The usual way to recover monotonicity consists of giving a special treatment to the small symbols like g when they occur below a big AC-symbol like f.

Remark that if we have a term s in flattened form, when adding a context $f(\ldots,s,\ldots)$, the flattening rules need to be applied at top position of the arguments only. Thus, we define the top-flattening of a term s wrt. an AC-symbol f, denoted by $tf_f(s)$, as $tf_f(f(s_1,\ldots,s_n)) = s_1,\ldots,s_n$ and $tf_f(s) = s$ if $top(s) \neq f$. Then $f(\ldots,tf_f(s),\ldots)$ is the flattened form of $f(\ldots,s,\ldots)$ for any given flattened term s and flattened context $f(...[]...)$.

Finally, let us give some definitions, which only concerns the AC-symbols. A strict ordering $\succ$ is AC-monotonic if $s \succ t$ implies $f(\ldots,tf_f(s),\ldots) \succ f(\ldots,tf_f(t),\ldots)$ for any $f \in \mathcal{F}_{AC}$. A quasi-ordering $\succeq$ is AC-monotonic if $s \succeq t$ implies $f(\ldots,tf_f(s),\ldots) \succeq f(\ldots,tf_f(t),\ldots)$ for any $f \in \mathcal{F}_{AC}$ and it fulfils the AC-deletion property if for all $f \in \mathcal{F}_{AC}$, if $n > m$ and $1 \leq i_1 < \ldots < i_m \leq n$ then $f(s_1,\ldots,s_n) \succeq f(s_{i_1},\ldots,s_{i_m})$.

2.1 ACRPO

We give a restricted version of the ACRPO in [Rub02], where the *precedence*, i.e., the well-founded quasi-ordering on the set of function symbols, is total and the arguments are always compared as multisets.

In the definition of ACRPO the main difference with RPO appears when comparing two terms headed by some AC-symbol f.

In this case, in order to ensure monotonicity, we have to give a different treatment to the arguments headed by symbols bigger than f and the arguments headed by symbols smaller that f. For this reason we need to define the following (multi)sets of terms.

Definition 3. *Let s be a term of the form $f(s_1, \ldots, s_n)$ with $f \in \mathcal{F}_{AC}$.*

- *The multiset of arguments of s headed by a symbol bigger than f, denoted by $BigHead(s)$, is defined as $\{s_i \mid 1 \leq i \leq n \wedge top(s_i) \succ_{\mathcal{F}} f\}$*
- *The multiset of arguments of s headed by a symbol not smaller than f, denoted by $NoSmallHead(s)$, is defined as $\{s_i \mid 1 \leq i \leq n \wedge f \not\succ_{\mathcal{F}} top(s_i)\}$*
- *The set of terms embedded in s through an argument headed by a small symbol, denoted by $EmbSmall(s)$, is defined as $\{f(s_1, \ldots, tf_f(v_j), \ldots, s_n) \mid s_i = h(v_1, \ldots, v_r) \wedge f \succ_{\mathcal{F}} h \wedge j \in \{1, \ldots, r\}\}$*

Note that the difference between $NoSmallHead(s)$ and $BigHead(s)$ is that the former includes the variables which are arguments of s and the latter does not. For AC-equivalent terms all these three (multi)sets coincide (modulo AC).

For instance, let $t = f(g(g(x,a),a),x,h(a),a)$ with $f \in \mathcal{F}_{AC}$, $h \succ_{\mathcal{F}} a \succ_{\mathcal{F}} f \succ_{\mathcal{F}} g$. Then $BigHead(t) = \{h(a), a\}$, $NoSmallHead(t) = \{x, h(a), a\}$ and $EmbSmall(t) = \{f(g(x,a),x,h(a),a), f(a,x,h(a),a)\}$.

Additionally, we need to count the number of arguments of a term t headed by an AC-symbol, but in order to preserve stability under substitutions we have to describe the counting by means of an expression with variables.

Definition 4. *Let s be a term of the form $f(s_1, \ldots, s_n)$ with $f \in \mathcal{F}_{AC}$. Then $\#(s)$ is an expression with variables on the positive integers, defined as $\#(f(s_1, \ldots, s_n)) = \#_v(s_1) + \ldots + \#_v(s_n)$, where $\#_v(x) = x$ and $\#_v(t) = 1$ if t is not a variable.*

For example, we have $\#(f(x,y,g(x))) = x+y+1$, which means the arguments coming from x (when applying a substitution and flattening), the ones coming from y plus 1. Then can compare the amount of arguments of two terms by checking $\#(f(x,y,g(x))) = x+y+1 > x+y = \#(f(x,y))$, which is necessary to achieve stability under substitution. For AC-equivalent terms the counting coincides.

Definition 5. *Let $\succeq_{\mathcal{F}}$ be a well-founded total precedence and let s and t be terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Then $s = f(s_1 \ldots s_n) \succ_{ACRPO} t$ if and only if*

1. *$s_i \succeq_{ACRPO} t$ for some $i \in \{1 \ldots n\}$ or*

2. $t = g(t_1...t_m)$, $f \succ_{\mathcal{F}} g$ *and* $s \succ_{ACRPO} t_i$ *for all* $i \in \{1...m\}$ *or*
3. $t = g(t_1...t_m)$, $f = g \notin \mathcal{F}_{AC}$ *and* $\{s_1, ..., s_n\} \gg_{ACRPO} \{t_1, ..., t_n\}$ *or*
4. $t = g(t_1...t_m)$, $f = g \in \mathcal{F}_{AC}$ *and* $s' \succeq_{ACRPO} t$ *for some* $s' \in EmbSmall(s)$ *or*
5. $t = g(t_1...t_m)$, $f = g \in \mathcal{F}_{AC}$, $s \succ_{ACRPO} t'$ *for all* $t' \in EmbSmall(t)$, $NoSmallHead(s) \underline{\gg}_{ACRPO} NoSmallHead(t)$ *and either*
   (a) $BigHead(s) \gg_{ACRPO} BigHead(t)$ *or*
   (b) $\#(s) > \#(t)$ *or*
   (c) $\#(s) \geq \#(t)$ *and* $\{s_1, ..., s_n\} \gg_{ACRPO} \{t_1, ..., t_m\}$

where $\succeq_{ACRPO}$ *is* $\succ_{ACRPO} \cup =_{AC}$ *and* $\gg_{ACRPO}$ *is the AC-multiset extension of* $\succ_{ACRPO}$.

The condition $NoSmallHead(s) \underline{\gg} NoSmallHead(t)$ ensures that every variable in t is taken care of by a variable in s or by an argument of s headed by a big symbol. Then, if by instantiation, some variable becomes a term headed by a big symbol, we know that some argument of the (instantiation) of s headed by a big symbol takes care of it.

The following examples show the behavior of the ordering when comparing terms headed by the same AC-symbol.

Example 3. Let $f \in \mathcal{F}_{AC}$ and $h, g, a, b \in \mathcal{F} \setminus \mathcal{F}_{AC}$, and take the precedence $h \succ_{\mathcal{F}} f \succ_{\mathcal{F}} g \succ_{\mathcal{F}} a \succ_{\mathcal{F}} b$. Then we have

1. $s = f(g(f(h(a), a)), a) \succ_{ACRPO} f(h(a), a, a) = t$ by case 4 since $f \succ_{\mathcal{F}} g$ implies $f(tf_f(f(h(a), a)), a) = f(h(a), a, a) \in EmbSmall(s)$.
2. $s = f(h(a), g(a)) \succ_{ACRPO} f(g(h(a)), a) = t$ by case 5a, since we have that $s \succ_{\mathcal{F}} f(h(a), a) \in EmbSmall(t)$ by case 4, $NoSmallHead(s) = \{h(a)\} \succeq \succeq_{ACRPO} \emptyset = NoSmallHead(t)$ and $BigHead(s) = \{h(a)\} \gg_{ACRPO} \emptyset = BigHead(t)$.
3. $s = f(g(h(a)), a, a, a) \succ_{ACRPO} f(g(f(h(a), a)), a) = t$ by case 5b, since we have $\#(s) = 4 > 2 = \#(t)$, $NoSmallHead(s) = \emptyset = NoSmallHead(t)$, $BigHead(s) = \emptyset = BigHead(t)$ and, finally, $s \succ_{ACRPO} f(h(a), a, a) = t' \in EmbSmall(t)$ by applying first case 4 and then $s' = f(h(a), a, a, a) \succ_{ACRPO} f(h(a), a, a) = t'$ by case 5b, since $NoSmallHead(s') = BigHead(s') = \{h(a)\} = BigHead(t') = NoSmallHead(t')$, $EmbSmall(t') = \emptyset$ and $\#(s') = 4 > 3 = \#(t')$.
4. $s = f(h(a), a) \succ_{ACRPO} f(h(a), b) = t$, by case 5c, as $EmbSmall(t) = \emptyset$, $NoSmallHead(s) = BigHead(s) = \{h(a)\}$ and $NoSmallHead(t) = BigHead(t) = \{h(a)\}$, $\#(s) = 2 = \#(t)$ and $\{h(a), a\} \gg_{ACRPO} \{h(a), b\}$.
5. $s = f(h(x, y), a, y) \succ_{ACRPO} f(g(h(x, y)), y) = t$ by case 5a, since we have that $NoSmallHead(s) = \{h(x, y), y\} \underline{\gg}_{ACRPO} \{y\} = NoSmallHead(t)$, $BigHead(s) = \{h(x, y)\} \gg_{ACRPO} \emptyset = BigHead(t)$ and finally, $s \succ_{ACRPO} f(h(x, y), y) = t' \in EmbSmall(t)$ by case 5b, since $NoSmallHead(s) = \{h(x, y), y\} = NoSmallHead(t')$, $BigHead(s) = \{h(x, y)\} = BigHead(t')$, $EmbSmall(t') = \emptyset$ and $\#(s) = y + 2 > y + 1 = \#(t')$.

2.2 AC-compatible polynomial interpretations

Polynomial interpretations can be used to prove AC-termination as well. To achieve AC-compatibility we need to define polynomial interpretations such that all AC-equivalent terms have the same interpretation. In this paper we will use polynomial interpretations as ingredient in our termination proofs.

Then, we can define a quasi-ordering and a strict ordering using this interpretations as follows:

- $s \succeq_\mu t$ iff $\mu(s) \geq \mu(t)$
- $s \succ_\mu t$ iff $\mu(s) > \mu(t)$

To ensure monotonicity of $\succeq_\mu$, we will only consider polynomial interpretations with non-negative integer coefficients.

Note that $\succ_\mu$ is included in $\succeq_\mu$ but does not coincide with the strict part of $\succeq_\mu$. The reason for making this distinction is that the given $\succ_\mu$ is stable under substitutions and the strict part of $\succeq_\mu$ is not, as shown in the example: if $\mu(f(x,y)) = x+y$, $\mu(g(x)) = x+1$ and $\mu(h(x)) = 0$ then $\mu(g(f(z_1,z_2))) = z_1 + z_2 + 1$ and $\mu(g(z_1)) = z_1 + 1$, and hence, $\mu(g(f(z_1,z_2))) \geq \mu(g(z_1))$ and $\mu(g(z_1)) \not\geq \mu(g(f(z_1,z_2)))$, but replacing z_2 by $h(z_3)$ we have $\mu(g(f(z_1,h(z_3)))) = z_1 + 1 = \mu(g(z_1))$. Hence before the instantiation, $g(f(z_1,z_2))$ and $g(z_1)$ compare in the strict part of $\succeq_\mu$ and, after the instantiation, they do not. Finally, the defined strict ordering $\succ_\mu$ does not have this stability problem since $g(f(z_1,z_2)) \not\succ_\mu g(z_1)$.

The following lemma gives a sufficient condition to ensure AC-compatibility.

Lemma 1. *[BCL87] Let μ be a polynomial interpretation. If for all AC-symbol f we have that $\mu(f(x,y))$ is of the form $axy + b(x+y) + c$ with $b^2 = b + ac$ then $\succeq_\mu$ and $\succ_\mu$ are AC-compatible.*

For instance, if f and g are AC-symbols, the interpretation μ defined by $\mu(f(x,y)) = x + y + 1$ or by $\mu(f(x,y)) = 2 * x * y + x + y$ fulfils the aforemention conditions.

2.3 The SPO

To conclude the preliminaries we will recall briefly the definition of the semantic path ordering [KL80]. The SPO is a generalization of RPO where the comparison with the precedence is replaced by a comparison using an underlying quasi-ordering involving the whole term.

Here we will present a slightly modified version of SPO where instead of using only an underlying quasi-ordering we will use a compatible ordering pair which includes a quasi-ordering $\succeq_Q$ and a strict ordering $\succ_q$ which is compatible with $\succeq_Q$. This is done in order to use polynomial interpretations as underlying quasi-ordering, since as seen, the strict polynomial ordering we have defined does not coincide with the strict part of the given polynomial quasi-ordering, since the latter is not stable under substitutions (see Section 2.2).

Definition 6. *Let $\succeq_Q$ be a quasi-ordering and let $\succ_q$ be an ordering. Then $\langle \succeq_Q, \succ_q \rangle$ is a* compatible ordering pair *if*

- *$\succ_q$ is compatible with $\succeq_Q$, i.e., $s' \succeq_Q s \succ_q t \succeq_Q t'$ implies $s' \succ_q t'$.*
- *$\succ_q$ is well-founded.*
- *$\succ_q$ and $\succeq_Q$ are stable under substitutions.*

These compatible ordering pairs can be defined using polynomial interpretations or using standard term interpretations over some term ordering. Another possibility is to combine (lexicographically) a precedence with (quasi-)orderings obtained by using interpretations.

From a given compatible ordering pair we can build the SPO in a recursive way following a path ordering scheme.

Definition 7. *(The SPO). Let $\langle \succeq_Q, \succ_q \rangle$ be a compatible ordering pair.*
Let s and t be terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Then $s = f(s_1 ... s_n) \succ_{spo} t$ if and only if

1. *$s_i \succeq_{spo} t$ for some $i \in \{1...n\}$ or*
2. *$t = g(t_1, \ldots, t_n)$, $s \succ_q t$ and $s \succ_{spo} t_i$ for all $i \in \{1...m\}$ and*
3. *$t = g(t_1, \ldots, t_n)$, $s \succeq_Q t$ and $\{s_1, ..., s_n\} \gg_{spo} \{t_1, ..., t_m\}$*

where $\succeq_{spo}$ is $\succ_{spo} \cup =$ and $\gg_{spo}$ is the multiset extension of $\succ_{spo}$.

The semantic path ordering is a well-founded ordering which is stable under substitutions. Unfortunately, it is, in general, not monotonic.

3 The AC-Semantic Path Ordering

In this section we will define an AC-compatible version of SPO, based on the definition of ACRPO. In this case, to be able to adapt the ACRPO scheme, we need to restrict the family of underlying orderings used inside SPO, to those using first a precedence and later (lexicographically) some quasi-ordering comparing the whole terms. Additionally, in order to ensure stability under substitutions we need to impose some extra conditions on this quasi-ordering.

Definition 8. *Let $\succeq_Q$ be a quasi-ordering and let $\succ_q$ be an ordering. Then $\langle \succeq_Q, \succ_q \rangle$ is an* AC-compatible ordering pair *if*

- *$\langle \succeq_Q, \succ_q \rangle$ is a compatible ordering pair.*
- *$\succ_q$ and $\succeq_Q$ are AC-compatible.*
- *$\succeq_Q$ fulfils the AC-deletion property.*

In addition, we say that $\langle \succeq_Q, \succ_q \rangle$ is AC-monotonic if both $\succeq_Q$ and $\succ_q$ are AC-monotonic.

Finally we have two more conditions relating the precedence and $\succeq_Q$.

(R1) $\forall g \in \mathcal{F}$ such that $f \succ_{\mathcal{F}} g$ for some $f \in \mathcal{F}_{AC}$,
$g(x_1, \ldots, x_n) \succeq_Q x_i$ for all $i = 1 \ldots n$

(R2) $f(s_1,\ldots,x,\ldots,s_n) \succeq_Q g(t_1,\ldots,x,\ldots,t_m)$ implies
$f(s_1,\ldots,y,\ldots,s_n) \succeq_Q g(t_1,\ldots,y,\ldots,t_m)$ for all $f =_{\mathcal{F}} g \in \mathcal{F}_{AC}$.

Due to the unavoidable condition of being AC-compatible, all other conditions except (R1) are easily satisfied. Condition (R1) has an impact on those symbols that are smaller, in the precedence, than an AC-symbol.

However, the allowed classes of quasi-orderings turn out to be powerful enough to obtain simple proofs of termination for the non-trivial examples given in the introduction.

Additionally, again due to the stability under substitutions, we need to *linearize* some variables when comparing two terms headed by equivalent AC-symbols.

Definition 9. *Given two terms s and t such that $top(s) =_{\mathcal{F}} top(t) \in \mathcal{F}_{AC}$ and a set of variables L, the* linearization *of s and t wrt. L is defined as*

$$lin(s,t,L) = \begin{cases} < s,t > & \text{if } \nexists\ t_i = s_j \in L \\ lin(s',t',L) & \text{if } t_i = s_j = v \in L \text{ where} \\ & w = new(v),\ s' = s[w]_j,\ t' = t[w]_i \end{cases}$$

Example 4. Let $f \in \mathcal{F}_{AC}$ and $h \succ_{\mathcal{F}} f \succ_{\mathcal{F}} g$. Given $s = f(h(x,y), g(z), z, y, z)$, $t = f(g(x), z, x, y, z)$ and $L = \{z\}$ we have that
$\langle \hat{s}, \hat{t} \rangle = lin(s,t,L) = \langle f(h(x,y), g(z), z_1, y, z_2), f(g(x), z_1, x, y, z_2) \rangle$.

We now give the definition of the ACSPO. For simplicity reasons, here we have only considered total precedences, but like for ACRPO this condition can be removed. On the other hand, since we consider $\succeq_{\mathcal{F}}$ as $\succ_{\mathcal{F}} \cup =_{\mathcal{F}}$ instead of $\succ_{\mathcal{F}}$ union syntactic equality then the flat form $\overline{t}$ of t has to be defined as the normal form of t wrt. the rules
$f(x_1 \ldots x_{i-1}, g(y_1 \ldots y_k), x_{i+1} \ldots x_n) \to f(x_1 \ldots x_{i-1}, y_1 \ldots y_k, x_{i+1} \ldots x_n)$
for each $f \in \mathcal{F}_{AC}$ and $g =_{\mathcal{F}} f$.

Definition 10. *(The ACSPO). Let $\succeq_{\mathcal{F}}$ be a total well-founded precedence such that for all $f, g \in \mathcal{F}$, $f =_{\mathcal{F}} g$ implies that either $f, g \in \mathcal{F} \setminus \mathcal{F}_{AC}$ or $f, g \in \mathcal{F}_{AC}$, and let $\langle \succeq_Q, \succ_q \rangle$ be an AC-monotonic and AC-compatible ordering pair, with $\succeq_Q$ fulfilling the above conditions (R1) and (R2).*

Let s, t be terms in $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Then $s = f(s_1 ... s_n) \succ_{acspo} t$ if and only if

1. *$s_i \succeq_{acspo} t$ for some $i \in \{1...n\}$ or*
2. *$f \in \mathcal{F}_{AC}$ and $s' \succeq_{acspo} t$ for some $s' \in EmbSmall(s)$ or*
3. *$t = g(t_1 ... t_m)$ and either $f \succ_{\mathcal{F}} g$ or ($f =_{\mathcal{F}} g$ and $s \succ_q t$), and $s \succ_{acspo} t_i$ for all $i \in \{1...m\}$, or*
4. *$t = g(t_1 ... t_m)$, $f =_{\mathcal{F}} g \notin \mathcal{F}_{AC}$, $s \succeq_Q t$ and $\{s_1, ..., s_n\} \succcurlyeq_{acspo} \{t_1, ..., t_n\}$ or*
5. *$t = g(t_1 ... t_m)$, $f =_{\mathcal{F}} g \in \mathcal{F}_{AC}$, $s \succeq_Q t$, $\hat{s} \succ_{acspo} t'$ for all $t' \in EmbSmall(\hat{t})$, $NoSmallHead(s) \succcurlyeq_{acspo} NoSmallHead(t)$ and either*
   (a) $BigHead(s) \succcurlyeq_{acspo} BigHead(t)$ or
   (b) $\#(s) > \#(t)$ or

(c) $\#(s) \geq \#(t)$ *and* $\{s_1, ..., s_n\} \gg_{acspo} \{t_1, ..., t_m\}$

where $\succeq_{acspo}$ *is* $\succ_{acspo} \cup =_{AC}$, $\gg_{acspo}$ *is the AC-multiset extension of* $\succ_{acspo}$ *and* $< \hat{s}, \hat{t} >= lin(s, t, L_{<s,t>})$ *where* $L_{<s,t>}$ *is the set of variables* $\{x \mid \exists\ t_i = x \in \mathcal{X}$ *and* $BigHead(s) \not\gg_{acspo} BigHead(t) \cup \{x\}\}$

The definition for *NoSmallHead*, *BigHead* and *EmbSmall* are given in Section 2. Note that linearization is only needed in case 5, when checking recursively the terms in *EmbSmall*. Remark that in this case, we have also required $NoSmallHead(s) \gtrdot_{acspo} NoSmallHead(t)$, which ensures that for any variable x occurring as argument of t such that $BigHead(s) \not\gg_{acspo} BigHead(t) \cup \{x\}$ there are, at least, as many occurrences of x as argument of s than as argument of t. Hence, after linearization all occurrences of variables in $L_{<s,t>}$ that are arguments in t have been renamed.

Theorem 2. $\succ_{acspo}$ *is a well-founded AC-compatible ordering which is stable under substitutions.*

4 The Monotonic AC-Semantic Path Ordering

Now we will obtain a monotonic version by using the same technique as for the MSPO in [BFR00] but considering AC-reduction triplets of the form $\langle \succeq_I, \succeq_Q, \succ_q \rangle$.

Definition 11. *A quasi-ordering* $\succeq_Q$ *is monotonic wrt.* $\succeq_I$ *if* $s \succeq_I t$ *implies* $f(\ldots s \ldots) \succeq_Q f(\ldots t \ldots)$ *for every* $f \in \mathcal{F}$. *Note that if* $f \in \mathcal{F}_{AC}$ *then we may need to apply some flattening.*

Then $\langle \succeq_I, \succeq_Q, \succ_q \rangle$ *is an AC-reduction triplet if* $\langle \succeq_Q, \succ_q \rangle$ *is an AC-monotonic AC-compatible ordering pair and* $\succeq_I$ *is an AC-compatible monotonic and stable under substitutions quasi-ordering and* $\succeq_Q$ *is monotonic wrt.* $\succeq_I$.

Definition 12. *(The ACMSPO). Let* $\succeq_{\mathcal{F}}$ *be a precedence and* $\langle \succeq_I, \succeq_Q, \succ_q \rangle$ *be an AC-reduction triplet, where* $\succeq_Q$ *satisfies conditions (R1) and (R2).*

$$s \succ_{acmspo} t \quad iff \quad s \succeq_I t \text{ and } s \succ_{acspo} t$$

Note that $\succeq_{\mathcal{F}}$ must fulfil the same condition required in the definition of ACSPO.

Theorem 3. $\succ_{acmspo}$ *is an AC-compatible reduction ordering.*

This result implies that ACMSPO is a suitable method for proving AC-termination without the need of considering any extension rule. However, in order to use this method in practice we need to build actual adequate triplets $\langle \succeq_I, \succeq_Q, \succ_q \rangle$ fulfilling all requirements.

In most of the cases it is enough to define $\succeq_Q$ (respectively $\succ_q$) as the application of $\succeq_I$ (respectively $\succ_i$) on terms after applying a renaming of the non-AC head symbols (as in the dependency pairs method). Hence we have $s \succeq_Q t$ iff

$N(s) \succeq_I N(t)$ (respectively $s \succ_q t$ iff $N(s) \succ_i N(t)$) for some renaming mapping applied only at top positions for non-AC symbols. In this case, the monotonicity of $\succeq_Q$ wrt. $\succeq_I$ is implied by the monotonicity of $\succeq_I$.

In all examples considered in this paper, even the renaming is not needed, and hence $\langle \succeq_Q, \succ_q \rangle$ and $\langle \succeq_I, \succ_i \rangle$ coincide.

In this paper, the pair $\langle \succeq_I, \succ_i \rangle$ is defined by a polynomial interpretation μ as seen in Section 2.2. The pair $\langle \succeq_I, \succ_i \rangle$ can also be obtained by using standard term interpretations (with straightforward conditions to ensure AC-compatibility and the other properties) over an AC-compatible reduction ordering on terms like ACRPO (in a similar way as done for the dependency pairs method and MSPO).

Furthermore, since $\langle \succeq_Q, \succ_q \rangle$ and $\langle \succeq_I, \succ_i \rangle$ coincide all conditions we have imposed on $\succeq_Q$ and $\succ_q$ should also be satisfied by $\succeq_I$ and $\succ_i$. Hence, we have to check that $\langle \succeq_I, \succ_i \rangle$ is an AC-monotonic AC-compatible ordering pair and $\succeq_I$, apart from monotonicity (which was already required), satisfies also R1 and R2.

The next lemma provides sufficient conditions for a polynomial interpretation to satisfy all required properties. A polynomial interpretations I is said to be *strictly positive* if no symbol is interpreted to 0 in I. The interpretation of a symbol f is said to be *fully argument dependent* if $I(f(x_1, \ldots, x_n))$ depends on all $x_1, \ldots, x_n$.

Lemma 2. *Given a precedence $\succeq_\mathcal{F}$ and a polynomial interpretation I with non-negative integer coefficients. A pair $\langle \succeq_I, \succ_i \rangle$ is an AC-monotonic AC-compatible ordering pair and $\succeq_I$ is monotonic and satisfies R1 and R2 if*

- *for all AC-symbol f we have that $I(f(x,y)) = axy + b(x+y) + c$ with*
  1. *$b^2 = b + ac$ and*
  2. *either I is strictly positive and $a \neq 0$ or $b \neq 0$.*
- *for all symbol g s.t. $f \succ_\mathcal{F} g$ for some AC-symbol f we have that either*
  1. *I is strictly positive and the interpretation of g is fully argument dependent, or*
  2. *$I(g(x_1, \ldots, x_n)) = P + b_1 x_1 + \ldots + b_n x_n$ where P is a polynomial and all $b_i \neq 0$.*

Similarly, we can impose some simple condition on term interpretations in order to satisfy all required properties.

5 Examples

In this section we will present several examples together with the ingredients, i.e., a precedence and a polynomial interpretation, required to show its termination by ACMSPO. For these examples, we have checked that all rules are included in ACMSPO. In this section, we assume that the ingredients are provided by the user and, hence, we only need to check the rules following the definition of ACMSPO (as usually done for path orderings). In the following section a way to automatically generate these ingredients is studied.

The first three examples come with the complete proof showing that all rules are included in ACMSPO with the given ingredients (i.e., a precedence and a polynomial interpretation).

Let us start with a very simple example.

Example 5. Both $+$ and $*$ are AC-symbols.

$$\begin{aligned} x + 0 &\to x \\ x + s(y) &\to s(x + y) \\ x * 0 &\to 0 \\ x * s(y) &\to x * y + x \end{aligned}$$

$$\begin{aligned} minus(x, 0) &\to x \\ minus(s(x), s(y)) &\to minus(x, y) \\ quot(0, s(y)) &\to 0 \\ quot(s(x), s(y)) &\to s(quot(minus(x, y), s(y))) \end{aligned}$$

Take as precedence $quot \succ_{\mathcal{F}} minus \succ_{\mathcal{F}} * \succ_{\mathcal{F}} + \succ_{\mathcal{F}} s$ and as $\langle \succeq_I, \succ_i \rangle = \langle \succeq_Q, \succ_q \rangle$ the polynomial interpretation $I(quot(x, y)) = x$, $I(minus(x, y)) = x$, $I(s(x)) = x + 1$, $I(0) = 1$, $I(x + y) = x + y$ and $I(x * y) = x * y + x + y$.

Now, we will show how the definition of ACMSPO is used for checking the rules. We have to check that all rules $l \to r$ are included in ACMSPO, which, by definition, means that $l \succeq_I r$ and $l \succ_{acspo} r$.

◇ First we prove $l \succeq_I r$ for all rule $l \to r$:

$$\begin{aligned} I(x + 0) = x + 1 &\geq x = I(x) \\ I(x + s(y)) = x + y + 1 &\geq x + y + 1 = I(s(x + y)) \\ I(x * 0) = x * 1 + x + 1 &\geq 1 = I(0) \\ I(x * s(y)) = x * (y + 1) + x + y + 1 &\geq x * y + x + y + x = I(x * y + x) \\ I(minus(x, 0)) = x &\geq x = I(x) \\ I(minus(s(x), x(y))) = x + 1 &\geq x = I(minus(x, y)) \\ I(quot(0, s(y))) = 1 &\geq 1 = I(0) \\ I(quot(s(x) + s(y))) = x + 1 &\geq x + 1 = I(s(quot(minus(x, y), s(y)))) \end{aligned}$$

◇ Now we show $l \succ_{acspo} r$ for all rule $l \to r$.

(1) $x + 0 \succ_{acspo} x$ by case 1

(2) $x + s(y) \succ_{acspo} s(x + y)$ by case 3:
$+ \succ_{\mathcal{F}} s$ and
$x + s(y) \succ_{acspo} x + y$ by case 2 since $x + y \in EmbSmall(x + s(y))$

(3) $x * 0 \succ_{acspo} 0$ by case 1.

(4) $x * s(y) \succ_{acspo} x * y + x$ by case 3:
$* \succ_{\mathcal{F}} +$ and
$x * s(y) \succ_{acspo} x * y$ by case 2 since $x * y \in EmbSmall(x * s(y))$
$x * s(y) \succ_{acspo} x$ by case 1.

(5) $minus(x, 0) \succ_{acspo} x$ by case 1.

(6) $minus(s(x), s(y)) \succ_{acspo} minus(x, y)$ by case 3:
$minus = minus$, $I(minus(s(x), s(y))) = x + 1 > x = I(minus(x, y))$
$minus(s(x), s(y)) \succ_{acspo} x$ by case 1 and
$minus(s(x), s(y)) \succ_{acspo} y$ by case 1.

(7) $quot(0, s(y)) \succ_{acspo} 0$ by case 1.
(8) $quot(s(x), s(y)) \succ_{acspo} s(quot(minus(x, y), s(y)))$ by case 3:
$quot \succ_{\mathcal{F}} s$ and
$quot(s(x), s(y)) \succ_{acspo} quot(minus(x, y), s(y))$ by case 3:
$I(quot(s(x), s(y))) = x + 1 > x = I(quot(minus(x, y), s(y)))$ and
$quot(s(x), s(y)) \succ_{acspo} minus(x, y)$ by case 3:
$quot \succ_{\mathcal{F}} minus$ and
$quot(s(x), s(y)) \succ_{acspo} x$ by case 1 and
$quot(s(x), s(y)) \succ_{acspo} y$ by case 1.
$quot(s(x), s(y)) \succ_{acspo} s(y)$ by case 1.

Example 6. Binary arithmetic (given in the introduction). Let us recall the rules ($+$ and $*$ are the only AC-symbols).

$$\begin{aligned} (\#)0 &\to \# \\ x + \# &\to x \\ x * \# &\to \# \end{aligned}$$

$$\begin{aligned} (x)0 + (y)0 &\to (x + y)0 \\ (x)0 + (y)1 &\to (x + y)1 \\ (x)1 + (y)1 &\to (x + y + (\#)1)0 \end{aligned}$$

$$\begin{aligned} x * (y)0 &\to (x * y)0 \\ x * (y)1 &\to x + (x * y)0 \end{aligned}$$

Take as precedence $* \succ_{\mathcal{F}} + \succ_{\mathcal{F}} 1 \succ_{\mathcal{F}} 0 \succ_{\mathcal{F}} \#$ and as $\langle \succeq_I, \succ_i \rangle = \langle \succeq_Q, \succ_q \rangle$ the polynomial interpretation $I(\#) = 1, I((x)0) = x, I((x)1) = x+2, I(x+y) = x+y$ and $I(x * y) = x * y + x + y$.

Now, we will show how the definition of ACMSPO is used for checking the rules. We have to check that all rules $l \to r$ are included in ACMSPO, which, by definition, means that $l \succeq_I r$ and $l \succ_{acspo} r$.

$\diamond$ First we prove $l \succeq_I r$ for all rule $l \to r$ in the TRS:

$$\begin{aligned} I((\#)0) = 1 &\geq 1 = I(\#) \\ I(x + \#) = x + 1 &\geq x = I(x) \\ I(x * \#) = x * 1 + x + 1 &\geq 1 = I(\#) \\ I((x)0 + (y)0) = x + y &\geq x + y = I((x + y)0) \\ I((x)0 + (y)1) = x + y + 2 &\geq x + y + 2 = I((x + y)1) \\ I((x)1 + (y)1) = x + 2 + y + 2 &\geq x + y + 1 + 2 = I((x + y + (\#)1)0) \\ I(x * (y)0) = x * y + x + y &\geq x * y + x + y = I((x * y)0) \\ I(x * (y)1) = x * (y + 2) + x + y + 2 &\geq x + x * y + x + y = I(x + (x * y)0) \end{aligned}$$

$\diamond$ Now we show $l \succ_{acspo} r$ for all rule $l \to r$ in the TRS.
(1) $(\#)0 \succ_{acspo} \#$ by case 1,
(2) $x + \# \succ_{acspo} x$ by case 1,
(3) $x * \# \succ_{acspo} \#$ by case 1,

(4) $(x)0 + (y)0 \succ_{acspo} (x + y)0$ by case 3
$+ \succ_{\mathcal{F}} 0$ and $(x)0 + (y)0 \succ_{acspo} x + y$ by applying case 2 twice since
$(x)0 + y \in EmbSmall((x)0 + (y)0)$ and
$x + y \in EmbSmall((x)0 + y)$

(5) $(x)0 + (y)1 \succ_{acspo} (x + y)1$ by case 3
$+ \succ_{\mathcal{F}} 1$ and $(x)0 + (y)1 \succ_{acspo} x + y$ by applying case 2 twice since
$(x)0 + y \in EmbSmall((x)0 + (y)1)$ and
$x + y \in EmbSmall((x)0 + y)$

(6) $(x)1 + (y)1 \succ_{acspo} (x + y + (\#)1)0$ by case 3
$+ \succ_{\mathcal{F}} 0$ and $(x)1 + (y)1 \succ_{acspo} x + y + (\#)1$ by case 3:
$+ =_{\mathcal{F}} +$ and $I((x)1 + (y)1) = x + y + 4 > x + y + 3 = I(x + y + (\#)1)$
$(x)1 + (y)1 \succ_{acspo} x$ by applying case 1 twice
$(x)1 + (y)1 \succ_{acspo} y$ by applying case 1 twice
$(x)1 + (y)1 \succ_{acspo} (\#)1$ by case 3 :
$+ \succ_{\mathcal{F}} 1$ and $(x)1 + (y)1 \succ_{acspo} \#$ by case 3 since $+ \succ_{\mathcal{F}} \#$

(7) $x * (y)0 \succ_{acspo} (x * y)0$ by case 3
$* \succ_{\mathcal{F}} 0$ and $x * (y)0 \succ_{acspo} x * y$ by case 2 since
$x * y \in EmbSmall(x * (y)0)$

(8) $x * (y)1 \succ_{acspo} x + (x * y)0$ by case 3
$* \succ_{\mathcal{F}} +$
$x * (y)1 \succ_{acspo} x$ by case 1 and
$x * (y)1 \succ_{acspo} (x * y)0$ by case 3
$* \succ_{\mathcal{F}} 0$ and $x * (y)1 \succ_{acspo} x * y$ by case 2 since
$x * y \in EmbSmall(x * (y)1)$

We come back two the second example of the introduction. Note that the polynomial interpretation we are using is very simple, which shows that the use of a precedence first is crucial.

Example 7. Ternary integral arithmetic (given in the introduction in Example 2). Let us recall the rules ($+$ and $*$ are the only AC-symbols).

$$
\begin{array}{rl}
(\#)0 & \to \# \\
x + \# & \to x \\
(x)0 + (y)0 & \to (x + y)0 \\
(x)0 + (y)1 & \to (x + y)1 \\
(x)0 + (y)j & \to (x + y)j \\
(x)1 + (y)j & \to (x + y)0 \\
(x)1 + (y)1 & \to (x + y + (\#)1)j \\
(x)j + (y)j & \to (x + y + (\#)j)1
\end{array}
\qquad
\begin{array}{rl}
x - y & \to x + opp(y) \\
opp(\#) & \to \# \\
opp((x)0) & \to (opp(x))0 \\
opp((x)1) & \to (opp(x))j \\
opp((x)j) & \to (opp(x))1 \\
x * \# & \to \# \\
x * (y)0 & \to (x * y)0 \\
x * (y)1 & \to x + (x * y)0 \\
x * (y)j & \to (x * y)0 + opp(x)
\end{array}
$$

Take as precedence $* \succ_{\mathcal{F}} - \succ_{\mathcal{F}} + \succ_{\mathcal{F}} opp \succ_{\mathcal{F}} 1 =_{\mathcal{F}} j \succ_{\mathcal{F}} 0 \succ_{\mathcal{F}} \#$ and as $\langle \succeq_I, \succ_i \rangle = \langle \succeq_Q, \succ_q \rangle$ the polynomial interpretation $I(\#) = 0$, $I(opp(x)) = x$, $I((x)0) = x$, $I((x)1) = I((x)j) = x + 1$, $I(x - y) = I(x + y) = x + y$ and $I(x * y) = x * y + x + y$.

Now, we will show how the definition of ACMSPO is used for checking the rules. We have to check that all rules $l \to r$ are included in ACMSPO, which, by definition, means that $l \succeq_I r$ and $l \succ_{acspo} r$.

$\diamond$ First we prove $l \succeq_I r$ for all rule $l \to r$ in the TRS:

$$
\begin{aligned}
I((\#)0) = 0 &\geq 0 = I(\#)\\
I(x + \#) = x + 0 &\geq x = I(x)\\
I((x)0 + (y)0) = x + y &\geq x + y = I((x + y)0)\\
I((x)0 + (y)1) = x + y + 1 &\geq x + y + 1 = I((x + y)1)\\
I((x)0 + (y)j) = x + y + 1 &\geq x + y + 1 = I((x + y)j)\\
I((x)1 + (y)j) = x + y + 2 &\geq x + y = I((x + y)0)\\
I((x)1 + (y)1) = x + y + 2 &\geq x + y + 2 = I((x + y + (\#)1)j)\\
I((x)j + (y)j) = x + y + 2 &\geq x + y + 2 = I((x + y + (\#)j)1)\\
I(x - y) = x + y &\geq x + y = I(x + opp(y))\\
I(opp(\#)) = 0 &\geq 0 = I(\#)\\
I(opp((x)0)) = x &\geq x = I((opp(x))0)\\
I(opp((x)1)) = x + 1 &\geq x + 1 = I((opp(x))j)\\
I(opp((x)j)) = x + 1 &\geq x + 1 = I((opp(x))1)\\
I(x * \#) = x * 0 + x + 0 &\geq 0 = I(\#)\\
I(x * (y)0) = x * y + x + y &\geq x * y + x + y = I((x * y)0)\\
I(x * (y)1) = x * (y + 1) + x + y + 1 &\geq x + x * y + x + y = I(x + (x * y)0)\\
I(x * (y)j) = x * (y + 1) + x + y + 1 &\geq x * y + x + y + x = I((x * y)0 + opp(x))
\end{aligned}
$$

$\diamond$ Now we show $l \succ_{acspo} r$ for all rule $l \to r$ in the TRS.

(1) $(\#)0 \succ_{acspo} \#$ by case 1,

(2) $x + \# \succ_{acspo} x$ by case 1,

(3) $(x)0 + (y)0 \succ_{acspo} (x + y)0$ by case 3
$+ \succ_{\mathcal{F}} 0$ and $(x)0 + (y)0 \succ_{acspo} x + y$ by applying case 2 twice since
$(x)0 + y \in EmbSmall((x)0 + (y)0)$ and
$x + y \in EmbSmall((x)0 + y)$

(4) $(x)0 + (y)1 \succ_{acspo} (x + y)1$ by case 3
$+ \succ_{\mathcal{F}} 1$ and $(x)0 + (y)1 \succ_{acspo} x + y$ by applying case 2 twice since
$(x)0 + y \in EmbSmall((x)0 + (y)1)$ and
$x + y \in EmbSmall((x)0 + y)$

(5) $(x)0 + (y)j \succ_{acspo} (x + y)j$ by case 3
$+ \succ_{\mathcal{F}} j$ and $(x)0 + (y)j \succ_{acspo} x + y$ by applying case 2 twice since
$(x)0 + y \in EmbSmall((x)0 + (y)j)$ and
$x + y \in EmbSmall((x)0 + y)$

(6) $(x)1 + (y)j \succ_{acspo} (x + y)0$ by case 3
$+ \succ_{\mathcal{F}} 0$ and $(x)1 + (y)j \succ_{acspo} x + y$ by applying case 2 twice since
$(x)1 + y \in EmbSmall((x)1 + (y)j)$ and
$x + y \in EmbSmall((x)1 + y)$

(7) $(x)1 + (y)1 \succ_{acspo} (x + y + (\#)1)j$ by case 3
$+ \succ_{\mathcal{F}} j$ and $(x)1 + (y)1 \succ_{acspo} x + y + (\#)1$ by case 3 since
$+ =_{\mathcal{F}} +$ and $I((x)1 + (y)1) = x + y + 2 > x + y + 1 = I(x + y + (\#)1)$
$(x)1 + (y)1 \succ_{acspo} x$ by applying case 1 twice
$(x)1 + (y)1 \succ_{acspo} y$ by applying case 1 twice
$(x)1 + (y)1 \succ_{acspo} (\#)1$ by case 3 since
$+ \succ_{\mathcal{F}} 1$ and $(x)1 + (y)1 \succ_{acspo} \#$ by case 3 since $+ \succ_{\mathcal{F}} \#$

(8) $(x)j + (y)j \succ_{acspo} (x + y + (\#)j)1$ by case 3
$+ \succ_{\mathcal{F}} 1$ and $(x)j + (y)j \succ_{acspo} x + y + (\#)j$ by case 3:
$+ =_{\mathcal{F}} +$ and $I((x)j + (y)j) = x + y + 2 > x + y + 1 = I(x + y + (\#)j)$
$(x)j + (y)j \succ_{acspo} x$ by applying case 1 twice
$(x)j + (y)j \succ_{acspo} y$ by applying case 1 twice
$(x)j + (y)j \succ_{acspo} (\#)j$ by case 3 since
$+ \succ_{\mathcal{F}} j$ and $(x)j + (y)j \succ_{acspo} \#$ by case 3 since $+ \succ_{\mathcal{F}} \#$

(9) $x - y \succ_{acspo} x + opp(y)$ by case 3
$- \succ_{\mathcal{F}} +$ and
$x - y \succ_{acspo} x$ by applying case 1 and
$x - y \succ_{acspo} opp(y)$ by case 3 since
$- \succ_{\mathcal{F}} opp$ and $x - y \succ_{acspo} y$ by case 1

(10) $opp(\#) \succ_{acspo} \#$ by case 1,

(11) $opp((x)0) \succ_{acspo} (opp(x))0$ by case 3
$opp \succ_{\mathcal{F}} 0$ and $opp((x)0) \succ_{acspo} opp(x)$ by case 4 since
$opp = opp$, $I(opp((x)0)) = x = I(opp(x))$ and
$\{(x)0\} \succcurlyeq_{acspo} \{x\}$ since $(x)0 \succ_{acspo} x$ by case 1

(12) $opp((x)1) \succ_{acspo} (opp(x))j$ by case 3
$opp \succ_{\mathcal{F}} j$ and $opp((x)1) \succ_{acspo} opp(x)$ by case 3 since
$opp = opp$, $I(opp((x)1)) = x + 1 > x = I(opp(x))$ and
$opp((x)1) \succ_{acspo} x$ by case 1

(13) $opp((x)j) \succ_{acspo} (opp(x))1$ by case 3
$opp \succ_{\mathcal{F}} 1$ and $opp((x)j) \succ_{acspo} opp(x)$ by case 3 since
$opp = opp$, $I(opp((x)j)) = x + 1 > x = I(opp(x))$ and
$opp((x)j) \succ_{acspo} x$ by case 1

(14) $x * \# \succ_{acspo} \#$ by case 1,

(15) $x * (y)0 \succ_{acspo} (x * y)0$ by case 3
$* \succ_{\mathcal{F}} 0$ and $x * (y)0 \succ_{acspo} x * y$ by case 2 since
$x * y \in EmbSmall(x * (y)0)$

(16) $x*(y)1 \succ_{acspo} x+(x*y)0$ by case 3
$*\succ_{\mathcal{F}} +$
$x*(y)1 \succ_{acspo} x$ by case 1
$x*(y)1 \succ_{acspo} (x*y)0$ by case 3
$*\succ_{\mathcal{F}} 0$ and
$x*(y)1 \succ_{acspo} x*y$ by case 2 since $x*y \in EmbSmall(x*(y)1)$
(17) $x*(y)j \succ_{acspo} (x*y)0 + opp(x)$ by case 3
$*\succ_{\mathcal{F}} +$
$x*(y)j \succ_{acspo} (x*y)0$ by case 3
$*\succ_{\mathcal{F}} 0$ and
$x*(y)j \succ_{acspo} x*y$ by case 2 since $x*y \in EmbSmall(x*(y)j)$
$x*(y)j \succ_{acspo} opp(x)$ by case 3 since
$*\succ_{\mathcal{F}} opp$ and $x*(y)j \succ_{acspo} x$ by case 1

The last example does not include any AC-symbol, but it is proved terminating in combination with the rules of the ternary integral arithmetic of the previous example. These two TRS are used in [CMR97] to later describe the rational numbers.

Example 8. Ternary integral arithmetic (cont.)

$$\begin{aligned}
sign(x) &\to sign?(x,\#)\\
sign?(\#,x) &\to x\\
sign?((x)0,y) &\to sign?(x,y)\\
sign?((x)1,y) &\to sign?(x,(\#)1)\\
sign?((x)j,y) &\to sign?(x,(\#)j)
\end{aligned}
\qquad
\begin{aligned}
abs(x) &\to abs?(x,x,\#)\\
abs?((x)0,y,z) &\to abs?(x,y,z)\\
abs?((x)1,y,z) &\to abs?(x,y,(\#)1)\\
abs?((x)j,y,z) &\to abs?(x,y,(\#)j)\\
abs?(\#,x,\#) &\to \#\\
abs?(\#,x,(\#)1) &\to x\\
abs?(\#,x,(\#)j) &\to opp(x)
\end{aligned}$$

$$\begin{aligned}
min(x,y) &\to min?(abs(y)-abs(x),x,y,\#)\\
min'(x,y) &\to min?(abs((y)1)-abs((x)1),x,y,\#)\\
min''(x,y) &\to min?(abs((y)j)-abs((x)j),x,y,\#)\\
min?((x)0,y,z,u) &\to min?(x,y,z,u)\\
min?((x)1,y,z,u) &\to min?(x,y,z,(\#)1)\\
min?((x)j,y,z,u) &\to min?(x,y,z,(\#)j)\\
min?(\#,x,y,\#) &\to x\\
min?(\#,x,y,(\#)1) &\to x\\
min?(\#,x,y,(\#)j) &\to y
\end{aligned}$$

Let R_1 be the rules for the Example 7 and R_2 the rules given above. Then $R_1 \cup R_2$ is proved included in ACMSPO by taking as precedence $min =_{\mathcal{F}} min' =_{\mathcal{F}} min'' \succ_{\mathcal{F}} min? \succ_{\mathcal{F}} abs \succ_{\mathcal{F}} abs? \succ_{\mathcal{F}} * \succ_{\mathcal{F}} - \succ_{\mathcal{F}} + \succ_{\mathcal{F}} opp \succ_{\mathcal{F}} 1 =_{\mathcal{F}} j \succ_{\mathcal{F}} 0 \succ_{\mathcal{F}} \#$ and as $\langle \succeq_I, \succ_i \rangle = \langle \succeq_Q, \succ_q \rangle$ the polynomial interpretation $I(\#) = 0$, $I(opp(x)) = x, I((x)0) = x, I((x)1) = I((x)j) = x+1, I(x-y) = I(x+y) = x+y$, $I(x*y) = x*y+x+y, I(sign(x)) = 2x, I(sign?(x,y)) = 2x+y, I(abs(x)) = 2x$, $I(abs?(x,y,z)) = x+y, I(min(x,y)) = 3x+3y, I(min'(x,y)) = I(min''(x,y)) = 3x+3y+4$ and $I(min?(x,y,z,u)) = x+y+z$.

6 Constraints

Proving termination with ACMSPO can be translated into a constraint solving problem in the same way as it is done for MSPO in [BFR00] and implemented in a fully automated system called *Termptation* [BR02]. The idea is to extract a set of constraints from the application of the definition of ACMSPO to the set of rules. Hence, if we can solve the constraints then there is a proof of termination of the given TRS using ACMSPO.

First, we obtain set of constraints on $\succeq_I$ from the first condition of the definition of ACMSPO applied to all rules. Then, we have to automatically generate a precedence to be used in the application of ACSPO. Note that, although theoretically there are exponentially many possibilities, in practice it can be done efficiently (as for the non-AC case). After that, we obtain constraints on $\succeq_I$ and $\succ_i$ by applying the definition of ACSPO. For instance, for the Example 6, after guessing the precedence $* \succ_{\mathcal{F}} + \succ_{\mathcal{F}} 1 \succ_{\mathcal{F}} 0 \succ_{\mathcal{F}} \#$ and assuming, for simplicity reasons, that $\succeq_I$ (resp. $\succ_i$) and $\succeq_Q$ (resp. $\succ_q$) coincide, we have the following constraints.

The first set is coming from the first condition of the definition of ACMSPO:

$$
\begin{array}{lll}
(\#)0 \succeq_I \# & (x)0 + (y)0 \succeq_I (x+y)0 & x * (y)0 \succeq_I (x * y)0 \\
x + \# \succeq_I x & (x)0 + (y)1 \succeq_I (x+y)1 & x * (y)1 \succeq_I x + (x * y)0 \\
x * \# \succeq_I \# & (x)1 + (y)1 \succeq_I (x+y+(\#)1)0 &
\end{array}
$$

The second set is coming from the recursive application of the definition of ACSPO, by adding all conditions on $\succeq_I$ and $\succ_i$ (in fact, on $\succeq_Q$ and $\succ_q$) that we find in the chosen path of the definition of ACSPO (we have taken the same path as in the proof given in the example). In this example, since most of the comparisons are solved by subterm (case 1), AC-embedding (case 2) or precedence (case 3), we have only one constraint in this second set:

$$(x)1 + (y)1 \succ_i x + y + (\#)1$$

These constraints can be handled and solved by finding an appropriate polynomial interpretation (as done, for instance, in CiME system [CMMU00]). We are currently working on the implementation of this method inside the *Termptation* system.

7 Further work and conclusions

The ACMSPO is the first method not being a simplification ordering that can prove AC-termination without considering extended rules. This allow us to prove in a simple way several non-trivial examples of the literature.

In order to ensure monotonicity and stability under substitutions we have needed to impose some additional conditions on the underlying orderings used in ACSPO. These conditions can be relaxed by adding more checking in the ACSPO and some more linearizations, which restrict the applicability of the method. This

trade off between the conditions imposed on the underlying orderings and the conditions required in the checking of ACSPO has to be further analyzed.

Another nice property of our method is that we strongly believe that it can be easily combined with methods based on extended rules. In fact, we can obtain a much simpler AC-compatible version of SPO, called EACSPO, which needs to consider extended rules. In this case, we only need to flatten terms and require the underlying quasi-ordering to be AC-compatible. The resulting method is very similar to the existing AC-versions of the dependency pairs method.

We are currently working on the combination both AC-extensions of SPO, namely ACSPO and EACSPO, with the aim of obtaining a method such that for every AC-symbol one can choose whether to add the extended rules or not. Then an AC-symbol is treated like in the ACSPO if it has no extended rules included, otherwise is treated like in EACSPO, which is almost like the other symbols. In this way one can get the best of both methods.

8 Proofs

In this section we provide all proofs of the properties of the orderings, in particular, Theorem 2 given in Section 3, and Theorem 3 and Lemma 2 both given in Section 4.

First we give the proof of Lemma 2 which provides sufficient conditions for a polynomial interpretation to satisfy all the properties required to be used as underlying triplet of the $\succ_{acmspo}$.

Lemma 2. *Given a precedence $\succeq_{\mathcal{F}}$ and a polynomial interpretation I (with non-negative integers coefficients). The pair $\langle \succeq_I, \succ_i \rangle$ is an AC-monotonic AC-compatible ordering pair and $\succeq_I$ is monotonic and satisfies R1 and R2 if*

1. *$I(f(x,y)) = axy + b(x+y) + c$ for all AC-symbol f and*
   (a) *$b^2 = b + ac$ and*
   (b) *either I is strictly positive and $a \neq 0$ or $b \neq 0$.*
2. *for all symbol g s.t. $f \succ_{\mathcal{F}} g$ for some AC-symbol f we have that either*
   (a) *I is strictly positive and the interpretation of g is fully argument dependent, or*
   (b) *$I(g(x_1, \ldots, x_n)) = P + b_1x_1 + \ldots + b_nx_n$ where P is a polynomial and all $b_i \neq 0$.*

Proof. For $\langle \succeq_I, \succ_i \rangle$ being an AC-monotonic AC-compatible ordering pair we have to prove that $\succ_i$ is well-founded, both $\succeq_I$ and $\succ_i$ are AC-compatible AC-monotonic and stable under substitutions, $\succ_i$ is compatible with $\succeq_I$ and that $\succeq_I$ fulfils the AC-deletion property. Additionally we have to prove that $\succeq_I$ is monotonic and satisfies R1 and R2.

By definition of $\succeq_I$ and $\succ_i$ we have that $\succ_i$ is well-founded, both $\succeq_I$ and $\succ_i$ are stable under substitutions and $\succ_i$ is compatible with $\succeq_I$. Since we only allow non-negative integer coefficients it is trivial to see that $\succeq_I$ is monotonic.

Condition $I(f(x,y)) = axy+b(x+y)+c$ for all AC-symbol f with $b^2 = b+ac$, by Lemma 1, ensures the AC-compatibility of $\succeq_I$ and $\succ_i$.

Now we will prove that for the given interpretations $\succ_i$ is AC-monotonic and $\succeq_I$ fulfils the AC-deletion property and conditions $R1$ and $R2$.

First we prove that $\succ_i$ is AC-monotonic, i.e. $s \succ_i t$ implies $f(s,u) \succ_i f(t,u)$ for all $f \in \mathcal{F}_{AC}$ and for all term u. Thus, we have $I(s) > I(t)$ and we have to prove $I(f(s,u)) > I(f(t,u))$. We distinguish whether $b = 0$ or $b \neq 0$.

- If $b = 0$ then, by condition 1b, we have that I is strictly positive and $a \neq 0$. Then, since $f \in \mathcal{F}_{AC}$ we have that $I(f(s,u)) = aI(s)I(u)+c$ and $I(f(t,u)) = aI(t)I(u)+c$. Since I is strictly positive we have that $I(s), I(t), I(u) > 0$ and since $a \neq 0$ and $I(s) > I(t)$ then conclude that $I(f(s,u)) > I(f(t,u))$.
- If $b \neq 0$ then $I(f(s,u)) = aI(s)I(u) + b(I(s) + I(u)) + c$ and $I(f(t,u)) = aI(t)I(u) + b(I(t) + I(u)) + c$. Since $I(s) > I(t)$ then we have that $I(s) > 0$, $aI(s)I(u) \geq aI(t)I(u)$ and $b(I(s) + I(u)) > b(I(t) + I(u))$. Therefore $I(f(s,u)) > I(f(t,u))$.

Now we prove that $\succeq_I$ fulfils the AC-deletion property, i.e. $f(s,(f(t,u))) \succeq_I f(t,u)$ for all terms s,t and u and AC-symbol f. Thus, we have to prove that $I(f(s,(f(t,u)))) \geq I(f(t,u))$ and we distinguish whether $b = 0$ or $b \neq 0$.

- If $b = 0$ then, by condition 1b, we have that I is strictly positive and $a \neq 0$. Then, since $f \in \mathcal{F}_{AC}$ we have that $I(f(s,f(t,u))) = aI(s)I(f(t,u)) + c$. Since I is strictly positive we have that $I(s) > 0$, $I(f(t,u)) > 0$ and since $a \neq 0$ then conclude that $I(f(s,f(t,u))) \geq I(f(t,u))$.
- If $b \neq 0$ then $I(f(s,f(t,u))) = aI(s)I(f(t,u)) + b(I(s) + I(f(t,u))) + c$. Since $b > 0$, $I(s) \geq 0$, $I(f(t,u)) \geq 0$ and $aI(s)I(f(t,u)) + c \geq 0$ then we have that $I(f(s,f(t,u))) = aI(s)I(f(t,u)) + b(I(s) + I(f(t,u))) + c \geq I(f(t,u))$.

Now we prove that $\succeq_I$ fulfils condition R1, i.e. $\forall g \in \mathcal{F}$ such that $f \succ_{\mathcal{F}} g$ for some $f \in \mathcal{F}_{AC}$, $g(x_1,\ldots,x_n) \succeq_Q x_i$ for all $i = 1 \ldots n$. If I is strictly positive and the interpretation of g is fully argument dependent then it holds trivially. Otherwise, by condition 2, $I(g(x_1,\ldots,x_n)) = P + b_1x_1 + \ldots + b_nx_n$ where P is a polynomial and all $b_j \neq 0$ and hence $I(g(x_1,\ldots,x_n)) \geq b_ix_i \geq x_i$.

Finally we prove $\succeq_I$ fulfils condition R2, i.e. $f(x,f(s_1,t_1)) \succeq_I g(x,g(s_2,t_2))$ implies $f(y,f(s_1,t_1)) \succeq_I g(y,g(s_2,t_2))$ for all $f =_{\mathcal{F}} g \in \mathcal{F}_{AC}$. Hence, we have to prove that $I(f(x,f(s_1,t_1))) \geq I(g(x,g(s_2,t_2)))$ implies $I(f(y,f(s_1,t_1))) \geq I(g(y,g(s_2,t_2)))$. Since $f,g \in \mathcal{F}_{AC}$, by condition 1, we have $I(f(x,f(s_1,t_1))) = a_1xI(f(s_1,t_1))+b_1(x+I(f(s_1,t_1)))+c_1 = x(a_1I(f(s_1,t_1))+b_1)+b_1I(f(s_1,t_1))+c_1 \geq x(a_2I(g(s_2,t_2)) + b_2) + b_2I(f(s_2,t_2)) + c_2 = a_2xI(g(s_2,t_2)) + b_2(x + I(f(s_2,t_2))) + c_2 = I(g(x,g(s_2,t_2)))$. Therefore, we have that $a_1I(f(s_1,t_1)) + b_1 \geq a_2I(g(s_2,t_2)) + b_2$ and $b_1I(f(s_1,t_1)) + c_1 \geq b_2I(f(s_2,t_2)) + c_2$ and since $I(f(y,f(s_1,t_1))) = a_1yI(f(s_1,t_1))+b_1(y+I(f(s_1,t_1)))+c_1 = y(a_1I(f(s_1,t_1))+b_1) + b_1I(f(s_1,t_1)) + c_1$ and $I(g(y,g(s_2,t_2))) = a_2yI(g(s_2,t_2)) + b_2(y + I(f(s_2,t_2))) + c_2 = y(a_2I(g(s_2,t_2)) + b_2) + b_2I(f(s_2,t_2)) + c_2$ then we conclude that $I(f(y,f(s_1,t_1))) \geq I(g(y,g(s_2,t_2)))$. □

In what follows we will provide all proofs of the properties of the orderings.

First we give a property, concerning to the AC-equality and the multisets $BigHead$, $NoSmallHead$ and $EmbSmall$ defined in Section 2.1, which was proved in [Rub02]:

Property 1. If $s =_{AC} t$ then $top(s) =_{\mathcal{F}} top(t)$, $BigHead(s) ==_{AC} BigHead(t)$, $NoSmallHead(s) ==_{AC} NoSmallHead(t)$ and $\#(s) = \#(t)$.

After flattening AC-equal terms are equal up to permutation of arguments of AC-symbols. Therefore, $\forall s_i\ \exists t_j$ s.t. $s_i =_{AC} t_j$ and $\forall s' \in EmbSmall(s)\ \exists t' \in EmbSmall(t)$ s.t. $s' =_{AC} t'$.

The next lemma establishes that the AC-equality is preserved after linearizing pairs of terms.

Lemma 3. *If $s =_{AC} s'$ and $t =_{AC} t'$ then $lin(s,t,L) =_{AC} lin(s',t',L)$ (up to renaming of variables) for any set of variables L.*

Proof. From Property 1 it follows that after flattening AC-equal terms are equal up to permutation of arguments of AC-symbols. Thus, any variable occurring as argument of a term u also occurs as argument in any AC-equal term of u. Hence, it trivially holds that $lin(s,t,L) =_{AC} lin(s',t',L)$ (up to renaming of variables). □

Now we prove the AC-compatibility of the ordering.

Lemma 4. *$\succ_{acspo}$ is AC-compatible.*

Proof. We have to prove the compatibility of $\succ_{acspo}$ wrt. permutative equality of arguments of AC-symbols, which we also call $=_{AC}$, i.e. $s' =_{AC} s \succ_{acspo} t =_{AC} t'$ implies $s' \succ_{acspo} t'$. We proceed by induction on $|s| + |t|$ and we distinguish several cases according to the proof of $s \succ_{acspo} t$. Let $s = f(s_1, \ldots, s_n)$, $s' = f(s'_1, \ldots, s'_n)$, $t = g(t_1, \ldots, t_m)$ and $t' = g(t'_1, \ldots, t'_m)$ (note that, by Property 1, $s =_{AC} s'$ implies $top(s) =_{\mathcal{F}} top(s')$, and $t =_{AC} t'$ implies $top(t) =_{\mathcal{F}} top(t')$).

1. If $s \succ_{acspo} t$ by case 1 then $s_i \succeq_{acspo} t$. By Property 1, $s =_{AC} s'$ implies that there is some $s'_j =_{AC} s_i$, and hence, by induction hypothesis (if $s_i \succ_{acspo} t$) and by transitivity of $=_{AC}$ (if $s_i =_{AC} t$), we obtain $s'_j \succeq_{acspo} t'$. Therefore we have $s' \succ_{acspo} t'$ by case 1.
2. If $s \succ_{acspo} t$ by case 2 then $u \succeq_{acspo} t$ for some $u \in EmbSmall(s)$. By Property 1, $s =_{AC} s'$ implies that there exists $u' \in EmbSmall(s')$ such that $u =_{AC} u'$, and from $u' =_{AC} u \succeq_{acspo} t =_{AC} t'$, by induction hypothesis (and by transitivity of $=_{AC}$), we obtain $u' \succeq_{acspo} t'$. Therefore $s' \succ_{acspo} t'$ by case 2.
3. If $s \succ_{acspo} t$ by case 3 then $s \succ_{acspo} t_i$ for all $i \in \{1...m\}$ and either $f \succ_{\mathcal{F}} g$ or ($f =_{\mathcal{F}} g$ and $s \succ_q t$). By Property 1, $t =_{AC} t'$ implies that for all t'_j there exists t_i such that $t_i =_{AC} t'_j$ and hence, by induction hypothesis, $s' =_{AC} s \succ_{acspo} t_i =_{AC} t'_j$ implies $s' \succ_{acspo} t'_j$.
   On the other hand, by AC-compatibility of $\succ_q$, $s' =_{AC} s \succ_q t =_{AC} t'$ implies $s' \succ_q t'$, and hence, we have that either $f \succ_{\mathcal{F}} g$ or ($f =_{\mathcal{F}} g$ and $s' \succ_q t'$). Thus, we conclude $s' \succ_{acspo} t'$ by case 3.

4. If $s \succ_{acspo} t$ by case 4 then $f =_{\mathcal{F}} g \notin \mathcal{F}_{AC}$, $s \succeq_Q t$ and $\{s_1, ..., s_n\} \succ \succ_{acspo} \{t_1, ..., t_m\}$. By AC-compatibility of $\succeq_Q$, $s' =_{AC} s \succeq_Q t =_{AC} t'$ implies $s' \succeq_Q t'$. Moreover, we have that $\{s'_1...s'_n\} ==_{AC} \{s_1...s_n\} \gg_{acspo} \{t_1...t_m\} ==_{AC} \{t'_1...t'_m\}$ and hence, by induction hypothesis and by AC-compatibility of the multiset extension of an AC-compatible relation we have $\{s'_1...s'_n\} \gg_{acspo} \{t'_1...t'_m\}$. Therefore $s' \succ_{acspo} t'$ by case 4.
5. If case $s \succ_{acspo} t$ by 5 then $f =_{\mathcal{F}} g \in \mathcal{F}_{AC}$, $s \succeq_Q t$, $\hat{s} \succ_{acspo} u$ for all $u \in EmbSmall(\hat{t})$ and $NoSmallHead(s) \underline{\gg}_{acspo} NoSmallHead(t)$. By AC-compatibility of $\succeq_Q$ we have $s' \succeq_Q t'$. By Property 1, we have that $NoSmallHead(s) ==_{AC} NoSmallHead(s')$ and $NoSmallHead(t) ==_{AC} NoSmallHead(t')$. Hence, by induction hypothesis and by AC-compatibility of the multiset extension of an AC-compatible relation, $NoSmallHead(s') \succeq \underline{\succeq}_{acspo} NoSmallHead(t')$.
   Now, we have to prove $\hat{s'} \succ_{acspo} u'$ for all $u' \in EmbSmall(\hat{t'})$. By Lemma 3, $s =_{AC} s'$ and $t =_{AC} t'$ implies $\hat{s} =_{AC} \hat{s'}$ and $\hat{t} =_{AC} \hat{t'}$, and by Property 1 $\hat{t} =_{AC} \hat{t'}$ implies that $\forall u' \in EmbSmall(\hat{t'})\ \exists u \in EmbSmall(\hat{t})$ such that $u =_{AC} u'$. Thus, we have $\hat{s'} =_{AC} \hat{s} \succ_{acspo} u =_{AC} u'$ that, by induction hypothesis, implies $\hat{s'} \succ_{acspo} u'$.
   Finally, we consider three cases.
   (a) If $s \succ_{acspo} t$ by case 5a then we have $BigHead(s') ==_{AC} BigHead(s) \succ \succ_{acspo} BigHead(t) ==_{AC} BigHead(t')$ that, by induction hypothesis and by AC-compatibility of the multiset extension of an AC-compatible relation, implies $BigHead(s') \gg_{acspo} BigHead(t')$, and hence $s' \succ_{acspo} t'$ by case 5a.
   (b) If $s \succ_{acspo} t$ by case 5b then $\#(s) > \#(t)$ and since, by Property 1, $\#(s) = \#(s')$ and $\#(t) = \#(t')$, we conclude $s' \succ_{acspo} t'$ by case 5b.
   (c) Otherwise $s \succ_{acspo} t$ by case 5c, that is, $\#(s) \geq \#(t)$ and $\{s_1, ..., s_n\} \succ \succ_{acspo} \{t_1, ..., t_m\}$. On the one hand, since, by Property 1, $\#(s) = \#(s')$ and $\#(t) = \#(t')$, we have that $\#(s') \geq \#(t')$. On the other hand, we have $\{s'_1...s'_n\} ==_{AC} \{s_1...s_n\} \gg_{acspo} \{t_1...t_m\} ==_{AC} \{t'_1...t'_m\}$ that, by induction hypothesis and by AC-compatibility of the multiset extension of an AC-compatible relation, implies $\{s'_1...s'_n\} \gg_{acspo} \{t'_1...t'_m\}$. Therefore $s' \succ_{acspo} t'$ by case 5c. □

In order to prove the transitivity of the ordering, first we need to prove some lemmas. The following lemma and the next property are necessary to prove properties concerning linearizations.

Lemma 5. *If $s = f(s_1...s_n) \succ_{acspo} g(t_1...t_m) = t$ by case (i) and $f \in \mathcal{F}_{AC}$ then $s\gamma \succ_{acspo} t\gamma$ by case (i) for all $\gamma = \{x_1 \mapsto z_1...x_k \mapsto z_k\}$ s.t. $\{z_1...z_k\}$ are variables and x_i are arguments which appear at most once in s and once in t and if they appear in t then they also appear in s.*

Proof. By induction on the sum of sizes of s and t.

- If $s \succ_{acspo} t$ by case 1 then $s_i \succeq_{acspo} t$ for some $i = 1...n$. $t \notin \mathcal{X}$ implies $s_i \notin \mathcal{X}$, and hence s_i does not contain any variable in $Dom(\gamma)$. Thus, we

also have that t does not contain any variable in $Dom(\gamma)$ either, and hence $s_i = s_i\gamma \succ_{acspo} t\gamma = t$. Therefore, $s\gamma \succ_{acspo} t\gamma$ by case 1.

- If $s \succ_{acspo} t$ by case 2 then $s' \succeq_{acspo} t$ for some $s' = f(\ldots, tf_f(v), \ldots) \in EmbSmall(s)$. Since for all $x_i \in Dom(\gamma)$ x_i does not appear in any term of $tf_f(v)$ and if x_i is argument of s then it is also argument of s', by induction hypothesis, we obtain $s'\gamma \succeq_{acspo} t\gamma$. Therefore, $s\gamma \succ_{acspo} t\gamma$ by case 2.
- If $s = f(s_1, \ldots, s_n) \succ_{acspo} t = g(t_1, \ldots, t_m)$ by case 3 then $s \succ_{acspo} t_i$ for all $i = 1, \ldots, m$ and either $f \succ_{\mathcal{F}} g$ or ($f =_{\mathcal{F}} g$ and $s \succ_q t$). By stability of $\succ_q$, $s\gamma \succ_q t\gamma$ and, by induction hypothesis, $s\gamma \succ_{acspo} t_i\gamma$ for all $t_i \notin \mathcal{X}$. Let $t_i = x_j \in Dom(\gamma)$ then, by assumption, there exists $s_p = x_j$ argument of s. Hence, $s_p\gamma = t_i\gamma$ and hence $s\gamma \succ_{acspo} t_i\gamma$ by case 1. Otherwise $t_i = x \notin Dom(\gamma)$ and hence $t_i\gamma = t_i$. $s \succ_{acspo} x$ implies that there exists $s_j \succeq_{acspo} x$, and by assumption $s_j\gamma = s_j$. Thus, $s_j\gamma \succeq_{acspo} x\gamma$ and hence $s\gamma \succ_{acspo} x\gamma = t_i\gamma$ by case 1. Therefore, we conclude $s\gamma \succ_{acspo} t\gamma$ by case 3.
- If $s \succ_{acspo} t$ by case 5 then $top(s) =_{\mathcal{F}} top(t)$, $s \succeq_Q t$, $\hat{s} \succ_{acspo} t'$ for all $t' \in EmbSmall(\hat{t})$, $NoSmallHead(s) \gtrdot\!\!\!\succeq_{acspo} NoSmallHead(t)$ and either $BigHead(s) \gtrdot_{acspo} BigHead(t)$, or $\#(s) > \#(t)$, or ($\#(s) \geq \#(t)$ and $\{s_1, ..., s_n\} \gtrdot_{acspo} \{t_1, ..., t_m\}$). By stability of $\succeq_Q$ we have $s\gamma \succeq_Q t\gamma$. From $\hat{s} \succ_{acspo} t'$, by induction hypothesis we have $\hat{s}\gamma' \succ_{acspo} t'\gamma'$ for all substitution γ' fulfilling the assumption. We have to prove $\widehat{s\gamma} \succ_{acspo} t''$ for all $t'' \in EmbSmall(\widehat{t\gamma})$ and to achieve it we will prove that $\exists\gamma'$ fulfilling the assumption such that $\hat{s}\gamma' = \widehat{s\gamma}$, $\hat{t}\gamma' = \widehat{t\gamma}$ and $t'\gamma' = t''$.

  Let $\gamma_0 = \{u_i \mapsto x_i \mid x_i \in Dom(\gamma) \text{ and } t|_p = x_i, \hat{t}|_p = u_i\}$ (note that $s = \hat{s}\gamma_0$ and $t = \hat{t}\gamma_0$). If $(\hat{s}\gamma_0)\gamma \neq \widehat{s\gamma}$ (up to renaming of variables) then $\exists\{x_i \mapsto z\} \in \gamma$ (x_i is argument of both s and t and it occurs only once in s and once in t) and either $\exists s_j = z$ or $\exists\{x_j \mapsto z\} \in \gamma$ s.t. $i \neq j$. Let $\gamma^z = \{x_i \mapsto z \mid x_i \mapsto z \in \gamma\} \subseteq \gamma$ then $\gamma_1 = (\gamma \backslash \gamma^z) \cup \{x_{i_1} \mapsto z_1, x_{i_2} \mapsto z_2, ...\}$ ($x_{i_k} \in Dom(\gamma^z)$ and z_k new variables). After doing the same process for every $\{x_i \mapsto z\} \in \gamma$ which fulfils the above conditions we obtain $\gamma' = \gamma_0 \cup \gamma_1 \cup \gamma_2 ...$ fulfilling the conditions of the assumption, such that $\hat{s}\gamma' = \widehat{s\gamma}$, $\hat{t}\gamma' = \widehat{t\gamma}$ and $t'\gamma' = t''$.

  Therefore, we have proved $\widehat{s\gamma} \succ_{acspo} t''$ for all $t'' \in EmbSmall(\widehat{t\gamma})$.
  Furthermore, since $NoSmallHead(s) \gtrdot\!\!\!\succeq_{acspo} NoSmallHead(t)$, $NoSmallHead(s\gamma) = (NoSmallHead(s) \setminus \{x_1 ... x_k\}) \cup \{z_1 ... z_k\}$ and $NoSmallHead(t\gamma) = (NoSmallHead(t) \setminus \{x_1 ... x_k\}) \cup \{z_1 ... z_k\}$ then we have $NoSmallHead(s\gamma) \gtrdot\!\!\!\succeq_{acspo} NoSmallHead(t\gamma)$.
  Finally, if $BigHead(s) \gtrdot_{acspo} BigHead(t)$, as $BigHead(s\gamma) = BigHead(s)$ and $BigHead(t\gamma) = BigHead(t)$ then we obtain $s\gamma \succ_{acspo} t\gamma$ by case 5a.
  If $\#(s) > \#(t)$ then $\#(s\gamma) > \#(t\gamma)$ and hence $s\gamma \succ_{acspo} t\gamma$ by case 5b.
  Otherwise, $\#(s) \geq \#(t)$ and hence $\#(s\gamma) > \#(t\gamma)$, and $\{s_1 ... s_n\} \gtrdot_{acspo} \{t_1 ... t_m\}$ that implies $\{s_1\gamma ... s_n\gamma\} = (\{s_1 ... s_n\} \setminus \{x_1 ... x_k\}) \cup \{z_1 ... z_k\} \gtrdot_{acspo} (\{t_1 ... t_m\} \setminus \{x_1 ... x_k\}) \cup \{z_1 ... z_k\} = \{t_1\gamma ... t_m\gamma\}$ and hence, $s\gamma \succ_{acspo} t\gamma$ by case 5c. □

Property 2. Let s and t be terms and L be a set of variables. If $< \hat{s}, \hat{t} >= lin(s, t, L)$ then $\exists\gamma$ s.t. $\hat{s}\gamma = s$ and $\hat{t}\gamma = t$.

Proof. γ is the substitution which maps the new variables in $lin(s,t,L)$ to the original variables in s and t. That is, $\gamma = \{u \mapsto x \mid t'_i = u, t_i = x, u \neq x, i = 1,\ldots,m\}$ where $t = f(t_1,\ldots,t_m)$ and $\hat{t} = f(t'_1,\ldots,t'_m)$. □

Let us recall the definition of $L_{\langle s,t\rangle}$ given inside the definition 3. $L_{\langle s,t\rangle}$ is the set of variables $\{x \mid \exists\ t_i = x \in \mathcal{X}\ \ and\ \ BigHead(s) \not\gg_{acspo} BigHead(t) \cup \{x\}\}$

Lemma 6. *Let* $top(s) =_{\mathcal{F}} top(t) \in \mathcal{F}_{AC}$, $NoSmallHead(s) \underline{\gg}_{acspo} NoSmallHead(t)$ *and* $\langle \hat{s}, \hat{t}\rangle = lin(s,t,L_{\langle s,t\rangle})$*:*

$\hat{s} \succ_{acspo} \hat{t}$ *implies* $s \succ_{acspo} t$,
$\hat{s} \succ_{acspo} t' \in EmbSmall(\hat{t})$ *implies* $s \succ_{acspo} t' \in EmbSmall(t)$ *and*
$\hat{s} \succ_{acspo} t'_i$ *argument of* $\hat{t}$ *implies* $s \succ_{acspo} t_i$ *argument of* t.

Proof. By using Property 2 and Lemma 5. □

With the following lemma we prove that the ordering $\succ_{acspo}$ fulfils the AC-deletion property.

Lemma 7. *Let* f *be an AC-symbol. If* $n > m$ *and* $1 \leq i_1 < \ldots < i_m \leq n$ *then* $s = f(s_1,\ldots,s_n) \succ_{acspo} f(s_{i_1},\ldots,s_{i_m}) = s'$. *That is,* $\succ_{acspo}$ *fulfils the AC-deletion property.*

Proof. By induction on $|s|$. Since $f \in \mathcal{F}_{AC}$ and $\succeq_Q$ fulfils the AC-deletion property, we have $f(s_1,\ldots,s_n) \succeq_Q f(s_{i_1},\ldots,s_{i_m})$. If $f(s_1,...,s_n) \succ_q f(s_{i_1},...,s_{i_m})$, we conclude $f(s_1,...,s_n) \succ_{acspo} f(s_{i_1},...,s_{i_m})$ by case 3 since $f(s_1,...,s_n) \succ_{acspo} s_{i_k}$ for all $k = 1,\ldots,m$ by case 1. Otherwise $s = f(s_1,...,s_n) \succeq_Q f(s_{i_1},...,s_{i_m}) = s'$.

Let $\langle \hat{s}, \widehat{s'}\rangle = lin(s,s',L_{\langle s,s'\rangle})$. We have that for all $u' = f(w_{i_1}...tf_f(v)...w_{i_m}) \in EmbSmall(\widehat{s'})$ there is $u = f(w_1...tf_f(v)...w_n) \in EmbSmall(\hat{s})$. By induction hypothesis, $u \succ_{acspo} u'$ and hence, $\hat{s} \succ_{acspo} u'$ for all $u' \in EmbSmall(\widehat{s'})$ by case 2. Finally, it is easy to see that $NoSmallHead(f(s_1,\ldots,s_n)) \underline{\gg}_{acspo} NoSmallHead(f(s_{i_1},\ldots,s_{i_m}))$ and that if $BigHead(f(s_1,\ldots,s_n)) \gg_{acspo} BigHead(f(s_{i_1},\ldots,s_{i_m}))$ then $f(s_1,...,s_n) \succ_{acspo} f(s_{i_1},...,s_{i_m})$ by case 5a, otherwise $f(s_1,...,s_n) \succ_{acspo} f(s_{i_1},...,s_{i_m})$ by case 5b. □

With the following two lemmas we will prove that $\succeq_{acspo}$ fulfils the subterm property.

Lemma 8. $s \succeq_{acspo} f(t_1,\ldots,t_m) = t$ *with* $f \in \mathcal{F}_{AC}$ *implies* $s \succ_{acspo} f(t_{i_1},\ldots,t_{i_k}) = t'$ *with* $m > k$ *and* $1 \leq i_1 < \ldots < i_k \leq m$.

Proof. By induction on $|s| + |t|$. Consider first $s = f(s_1,\ldots,s_n) =_{AC} t$ then we also have that there exists $s' = (s_{j_1},\ldots,s_{j_k}) =_{AC} f(t_{i_1},\ldots,t_{i_k}) = t'$. Since by Lemma 7 $s \succ_{acspo} s'$ then, by AC-compatibility of $\succ_{acspo}$ we obtain $s \succ_{acspo} t'$. Otherwise we distinguish several cases according to the proof of $s \succ_{acspo} t$.

- If $s \succ_{acspo} t$ by case 1, i.e. $s_i \succeq_{acspo} t$ then, by induction hypothesis, we have $s_i \succ_{acspo} t'$ and hence $s \succ_{acspo} t'$ by case 1.

- If $s \succ_{acspo} t$ by case 2, i.e. $u \succeq_{acspo} t$ for some $u \in EmbSmall(s)$ then, by induction hypothesis, we have $u \succ_{acspo} t'$ and hence $s \succ_{acspo} t'$ by case 2.
- $s \succ_{acspo} t$ by case 3, i.e. $s \succ_{acspo} t_i$ for all $i = 1, \ldots, m$ and either $top(s) \succ_{\mathcal{F}} f$ or $(top(s) =_{\mathcal{F}} f$ and $s \succ_q t)$.
  On the one hand, $s \succ_{acspo} t_i$ for all $i = 1, \ldots, m$ implies $s \succ_{acspo} t_{i_j}$ for all $j = 1, \ldots, k$. On the other hand, by the AC-deletion property of $\succeq_Q$ we have $t \succeq_Q t'$, and hence, by compatibility of $\succ_q$ and $\succeq_Q$, we obtain $s \succ_q t'$. Thus, since either $top(s) \succ_{\mathcal{F}} f$ or $(top(s) =_{\mathcal{F}} f$ and $s \succ_q t')$ we conclude $s \succ_{acspo} t'$ by case 3.
- Otherwise $s = g(s_1, \ldots, s_n) \succ_{acspo} t$ by case 5, i.e. $g =_{\mathcal{F}} f$, $s \succeq_Q t$, $\hat{s} \succ_{acspo} u$ for all $u \in EmbSmall(\hat{t})$, $NoSmallHead(s) \succcurlyeq_{acspo} NoSmallHead(t)$ and either $BigHead(s) \succ\!\!\succ_{acspo} BigHead(t)$ or $\#(s) > \#(t)$ or $(\#(s) \geq \#(t)$ and $\{s_1, \ldots, s_n\} \succ\!\!\succ_{acspo} \{t_1, \ldots, t_m\})$. By the AC-deletion property of $\succeq_Q$, $t \succeq_Q t'$ and, by transitivity of $\succeq_Q$, $s \succeq_Q t'$. Let $\langle \hat{s}, \hat{t} \rangle = lin(s, t, L_{\langle s,t \rangle})$ and $\langle \check{s}, \check{t'} \rangle = lin(s, t', L_{\langle s,t' \rangle})$. Now we prove $\check{s} \succ_{acspo} u' \in EmbSmall(\check{t'})$. Remark that $BigHead(t') \subseteq BigHead(t)$ and $NoSmallHead(t') \subseteq NoSmallHead(t)$ implies $L_{\langle s,t' \rangle} \subseteq L_{\langle s,t \rangle}$.
  Let γ be the substitution which deslinearizes all the variables in $L_{\langle s,t \rangle} \setminus L_{\langle s,t' \rangle}$. For all $u' = f(t'_{i_1}, \ldots, tf_f(v), \ldots, t'_{i_k}) \in EmbSmall(\check{t'})$ there exists $u'' = f(t''_{i_1}, \ldots, tf_f(v), \ldots, t''_{i_k}) \in EmbSmall(\hat{t})$ such that, $\check{s} = \hat{s}\gamma$, $u' = u''\gamma$ and, by induction hypothesis, $\hat{s} \succ_{acspo} u$ implies $\hat{s} \succ_{acspo} u''$. Thus, by Lemma 5, we obtain $\check{s} \succ_{acspo} u'$.
  Finally, since $NoSmallHead(t') \subseteq NoSmallHead(t)$ and $BigHead(t') \subseteq BigHead(t)$ then we have $NoSmallHead(s) \succcurlyeq_{acspo} NoSmallHead(t')$ and $BigHead(s) \succcurlyeq_{acspo} BigHead(t')$.
  Therefore, if $BigHead(s) \succ\!\!\succ_{acspo} BigHead(t')$ we conclude $s \succ_{acspo} t'$ by case 5a, otherwise we conclude $s \succ_{acspo} t'$ by case 5b. □

Lemma 9. *$s \succeq_{acspo} t$ implies $s \succ_{acspo} t_i$ for all $i \in \{1, \ldots, m\}$.*

Proof. By induction on $|s| + |t|$. Let s be $f(s_1, ..., s_n)$ and t be $g(t_1, ..., t_m)$. If $s =_{AC} t$ then, by Property 1, for every t_i there is some s_j such that $s_j =_{AC} t_i$ and therefore, by case 1, we have $s \succ_{acspo} t_i$. Otherwise $s \succ_{acspo} t$. We distinguish several cases according to the definition of $\succ_{acspo}$.

1. If $s \succ_{acspo} t$ by case 1 then $s_j \succeq_{acspo} t$ for some $j \in \{1...n\}$. By induction hypothesis $s_j \succ_{acspo} t_i$ which, by case 1, implies $s \succ_{acspo} t_i$.
2. If $s \succ_{acspo} t$ by case 2 then $s' \succ_{acspo} t$ for some $s' \in EmbSmall(s)$ and hence, by induction hypothesis, $s' \succ_{acspo} t_i$. Therefore, $s \succ_{acspo} t_i$ by case 2.
3. If $s \succ_{acspo} t$ by case 3 then $s \succ_{acspo} t_i$ by definition.
4. If $s \succ_{acspo} t$ by case 4, then $\{s_1, ..., s_n\} \succ\!\!\succ_{acspo} \{t_1, ..., t_n\}$ and hence $\forall t_j \; \exists s_i$ s.t. $s_i \succeq_{acspo} t_j$. Thus, by case 1, we have $s \succ_{acspo} t_j$.
5. If $s \succ_{acspo} t$ by case 5 we distinguish two cases: (as we consider terms in flattened form $f =_{\mathcal{F}} g \neq_{\mathcal{F}} top(t_i)$ for all $i = 1, \ldots, m$):
   (a) If $t_i \in NoSmallHead(t)$ (i.e. $t_i \in \mathcal{X}$ or $top(t_i) \succ_{\mathcal{F}} g$) then, since $NoSmallHead(s) \succcurlyeq_{acspo} NoSmallHead(t)$ implies that there is some $s_j \in NoSmallHead(s)$ such that $s_j \succeq_{acspo} t_i$, we conclude $s \succ_{acspo} t_i$ by case 1.

(b) Otherwise $t_i = h(u_1, \ldots, u_r)$ with $g \succ_{\mathcal{F}} h$. Let $\langle \hat{s}, \hat{t} \rangle = lin(s, t, L_{\langle s,t \rangle})$ and $\hat{t} = g(t'_1, \ldots, t_i, \ldots, t'_m)$. By definition of $EmbSmall$ we have that for every u_k argument of t_i there is $t' = g(t'_1, \ldots, tf_g(u_k), \ldots, t'_m) \in EmbSmall(\hat{t})$. Moreover, since $s \succ_{acspo} t$ by case 5 implies $\hat{s} \succ_{acspo} t'$ then, by induction hypothesis, we have $\hat{s} \succ_{acspo} t'_j$ for all $j \in \{1, \ldots, i-1, i+1, \ldots, m\}$ and $\hat{s} \succ_{acspo} v$ for all v in $tf_g(u_k)$. Since $g \succ_{\mathcal{F}} h$, if we prove $\hat{s} \succ_{acspo} u_k$ for all u_k argument of t_i then we will conclude $\hat{s} \succ_{acspo} t_i$ by case 3 and hence, by Lemma 6, we will conclude $s \succ_{acspo} t_i$. Thus, now we prove $\hat{s} \succ_{acspo} u_k$. If $u_k = tf_g(u_k)$ then we are done. Otherwise $u_k = g(v_1, \ldots, v_r)$ and since $\hat{s} \succ_{acspo} t' = g(\ldots, v_1, \ldots, v_r, \ldots)$, by Lemma 8, implies $\hat{s} \succ_{acspo} g(v_1, \ldots, v_r)$ then we are done. □

Corollary 1. *$\succeq_{acspo}$ fulfils the subterm property.*

The next property is needed in the proof the lemma below. Both the property and the lemma will be used later in several proofs.

Property 3. $t \succeq_Q t'$ for all $t' \in EmbSmall(t)$.

Proof. It follows from condition (R1) and the AC-monotonicity of $\succeq_Q$. □

Lemma 10. *$s \succeq_{acspo} t$ and $top(t) \in \mathcal{F}_{AC}$ imply $s \succ_{acspo} t' \; \forall \, t' \in EmbSmall(t)$.*

Proof. By induction on $|s| + |t|$. If $s =_{AC} t$ then Property 1 implies that $\forall t' \in EmbSmall(t)$ $\exists s' \in EmbSmall(s)$ such that $s' =_{AC} t'$, and hence, we conclude $s \succ_{acspo} t'$ by case 2. Otherwise $s \succ_{acspo} t$ and we distinguish several cases according to the proof of $s \succ_{acspo} t$.

1. If $s \succ_{acspo} t$ by case 1 then $s_i \succeq_{acspo} t$ which by induction hypothesis implies $s_i \succ_{acspo} t'$ and hence $s \succ_{acspo} t'$ by case 1.
2. If $s \succ_{acspo} t$ by case 2 then $s' \succeq_{acspo} t$ for some $s' \in EmbSmall(s)$. By induction hypothesis we have $s' \succ_{acspo} t'$ and hence $s \succ_{acspo} t'$ by case 2.
3. If $s = f(s_1, \ldots, s_n) \succ_{acspo} g(t_1, \ldots, t_m) = t$ by case 3 then $s \succ_{acspo} t_i$ for all $i \in \{1, \ldots, m\}$ and either $f \succ_{\mathcal{F}} g$ or ($f =_{\mathcal{F}} g$ and $s \succ_q t$). Let $t' = g(t_1, \ldots, tf_g(w), \ldots, t_m) \in EmbSmall(t)$ and $t_j = h(...w...)$ with $g \succ_{\mathcal{F}} h$. We have $s \succ_{acspo} t_i$ and we need to prove $s \succ_{acspo} v$ for all v in $tf_g(w)$ that holds since $s \succ_{acspo} t_j = h(...w...)$ by Lemma 9 implies $s \succ_{acspo} w$, and if $tf_g(w) = v_1, \ldots, v_r$ then $s \succ_{acspo} w$ implies $s \succ_{acspo} v_j$ for all $j = 1, \ldots, r$ by applying Lemma 9 again.
   Since, by Property 3, $t \succeq_Q t'$ then we have that $s \succ_q t \succeq_Q t'$ implies $s \succ_q t'$ by compatibility of $\succ_q$ and $\succeq_Q$. Thus, we have that either $f \succ_{\mathcal{F}} g = top(t')$ or ($f =_{\mathcal{F}} g$ and $s \succ_q t'$) and hence we conclude $s \succ_{acspo} t'$ by case 3.
4. If $s \succ_{acspo} t$ by case 5 then we have $\hat{s} \succ_{acspo} t'' \in EmbSmall(\hat{t})$ and, by Lemma 6, we conclude $s \succ_{acspo} t' \in EmbSmall(t)$. □

The following two lemmas show that some relations between two terms also hold when using their linearized versions.

Lemma 11. *Let* $s = f(s_1, \ldots, s_n)$, $t = g(t_1, \ldots, t_m)$ *with* $f =_{\mathcal{F}} g \in \mathcal{F}_{AC}$ *and* $\langle \hat{s}, \hat{t} \rangle = lin(s, t, L_{\langle s,t \rangle})$. $NoSmallHead(s) \underline{\succcurlyeq}_{acspo} NoSmallHead(t)$ *implies* $NoSmallHead(\hat{s}) \underline{\succcurlyeq}_{acspo} NoSmallHead(\hat{t})$

Proof. By definition of $L_{\langle s,t \rangle}$, $NoSmallHead(s) \underline{\succcurlyeq}_{acspo} NoSmallHead(t)$ implies that for all $y \in L_{\langle s,t \rangle}$ with $1 \leq i_1 < \ldots < i_k \leq m$ such that $t_{i_l} = y$ for all $l = 1, \ldots, k$, there exists $1 \leq j_1 < \ldots < j_k \leq m$ such that $s_{j_l} = y$ for all $l = 1, \ldots, k$. Therefore, $NoSmallHead(\hat{s}) = (NoSmallHead(s) \setminus W) \cup W^L$ where $W = \{t_j | t_j \in L_{\langle s,t \rangle}\}$ and W^L is the set containing the renamed version in $\langle \hat{s}, \hat{t} \rangle$ of the elements in W, and $NoSmallHead(\hat{t}) = (NoSmallHead(t) \setminus W) \cup W^L$. Hence, $NoSmallHead(s) \underline{\succcurlyeq}_{acspo} NoSmallHead(t)$ implies $NoSmallHead(\hat{s}) \underline{\succcurlyeq}_{acspo} NoSmallHead(\hat{t})$ □

Lemma 12. *Let* $\langle \hat{s}, \hat{t} \rangle = lin(s, t, L_{\langle s,t \rangle})$. *If* $s \succ_{acspo} t$ *by case 5 then* $\hat{s} \succ_{acspo} \hat{t}$ *by case 5*

Proof. By condition (R2) of $\succeq_Q$ we have that $s \succeq_Q t$ implies $\hat{s} \succeq_Q \hat{t}$. Since $\hat{\hat{s}} = \hat{s}$ and $\hat{\hat{t}} = \hat{t}$ (up to renaming of variables), we have that $\hat{\hat{s}} \succ_{acspo} t'' \in EmbSmall(\hat{\hat{t}})$ and, by Lemma 11, $NoSmallHead(s) \underline{\succcurlyeq}_{acspo} NoSmallHead(t)$ implies $NoSmallHead(\hat{s}) \underline{\succcurlyeq}_{acspo} NoSmallHead(\hat{t})$.

Finally, since $BigHead(s) = BigHead(\hat{s})$ and $BigHead(t) = BigHead(\hat{t})$, it is easy to see that $\hat{s} \succ_{acspo} \hat{t}$ holds by the same subcase as $s \succ_{acspo} t$. □

Now we can prove the transitivity of the ordering.

Lemma 13. $\succ_{acspo}$ *is transitive, i.e.* $s \succ_{acspo} t$ *and* $t \succ_{acspo} u$ *implies* $s \succ_{acspo} u$.

Proof. By induction on $|s|+|t|+|u|$. We distinguish several cases according to the definition of $\succ_{acspo}$. Let $s = f(s_1, ..., s_n)$, $t = g(t_1, ..., t_m)$ and $u = h(u_1, ..., u_p)$.

1. If $s \succ_{acspo} t$ by case 1 then $s_i \succeq_{acspo} t$ for some $i \in \{1...n\}$. By induction hypothesis and AC-compatibility, $s_i \succeq_{acspo} t \succ_{acspo} u$ implies $s_i \succ_{acspo} u$ and hence $s \succ_{acspo} u$ by case 1.
2. If $s \succ_{acspo} t$ by 2 then $s' \succeq_{acspo} t$ for some $s' \in EmbSmall(s)$ that by induction hypothesis and AC-compatibility implies $s' \succ_{acspo} u$. Thus, by case 2, $s \succ_{acspo} u$.
3. If $t \succ_{acspo} u$ by case 1 then $t_i \succeq_{acspo} u$ for some $i \in \{1...m\}$. By Lemma 9, $s \succ_{acspo} t$ implies $s \succ_{acspo} t_i$. Thus, by induction hypothesis and AC-compatibility, $s \succ_{acspo} u$.
4. If $t \succ_{acspo} u$ by case 2 then $t' \succeq_{acspo} u$ for some $t' \in EmbSmall(t)$. By Lemma 10, $s \succ_{acspo} t$ implies $s \succ_{acspo} t'$. Thus, by induction hypothesis and AC-compatibility, $s \succ_{acspo} u$.
5. Otherwise we have (A) either $f \succ_{\mathcal{F}} g$ or ($f =_{\mathcal{F}} g$ and $s \succeq_Q t$); and (B) either $g \succ_{\mathcal{F}} h$ or ($g =_{\mathcal{F}} h$ and $t \succeq_Q u$). Additionally we have that $t \succ_{acspo} u$, by Lemma 9, implies $t \succ_{acspo} u_i$ and hence, by induction hypothesis, we obtain $s \succ_{acspo} u_i$.

- If either $f \succ_{\mathcal{F}} g$ or $g \succ_{\mathcal{F}} h$ then we have $f \succ_{\mathcal{F}} h$ and hence we conclude by case 3.
  Otherwise $f =_{\mathcal{F}} g =_{\mathcal{F}} h$ that, by transitivity of $=_{\mathcal{F}}$, implies $f =_{\mathcal{F}} h$.
- $f =_{\mathcal{F}} h$ and either $s \succ_q t$ or $t \succ_q u$. By transitivity of $\succ_q$ or by compatibility of $\succ_q$ with $\succeq_Q$ we obtain $s \succ_q u$. Thus, we conclude $s \succ_{acspo} t$ by 3.
  Otherwise $s \succeq_Q t \succeq_Q u$ that, by transitivity of $\succeq_Q$, implies $s \succeq_Q u$.
- $f =_{\mathcal{F}} h$, $s \succeq_Q u$ and $f, g, h \in \mathcal{F} \setminus \mathcal{F}_{AC}$. That is, both $s \succ_{acspo} t$ and $t \succ_{acspo} u$ hold by case 4, i.e. $\{s_1 ... s_n\} \gg_{acspo} \{t_1 ... t_m\} \gg_{acspo} \{u_1 ... u_p\}$. Therefore, by induction hypothesis and by transitivity of the multiset extension of a transitive relation, we conclude $\{s_1 ... s_n\} \gg_{acspo} \{u_1 ... u_p\}$, and hence $s \succ_{acspo} u$ holds by case 4.
- $f =_{\mathcal{F}} h$, $s \succeq_Q u$ and $f, g, h \in \mathcal{F}_{AC}$. That is, both $s \succ_{acspo} t$ and $t \succ_{acspo} u$ hold by case 5.

  Let $\langle \hat{s}, \hat{u} \rangle = lin(s, u, L_{\langle s,u \rangle})$, $\langle s^1, t^1 \rangle = lin(s, t, L_{\langle s,t \rangle})$ and $\langle t^2, u^2 \rangle = lin(t, u, L_{\langle t,u \rangle})$. First we will prove that $\hat{s} \succ_{acspo} u'' \in EmbSmall(\hat{u})$.
  For every argument $u_i = x$ with $x \notin L_{\langle t,u \rangle}$ we know by definition that $BigHead(t) \gg_{acspo} BigHead(u) \cup \{x\}$. Since $BigHead(s) \gtrsim_{acspo} BigHead(t)$ then we have that $BigHead(s) \gg_{acspo} BigHead(u) \cup \{x\}$ and hence $x \notin L_{\langle s,u \rangle}$ by definition. Therefore $L_{\langle s,u \rangle} \subseteq L_{\langle t,u \rangle}$.
  Let γ be a substitution which maps all the variables which are linearized in $\langle t^2, u^2 \rangle$ and not in $\langle \hat{s}, \hat{u} \rangle$ (i.e. in $L_{\langle t,u \rangle} \setminus L_{\langle s,u \rangle}$) to their original value. By Lemma 12, $t \succ_{acspo} u$ by case 5 implies $t^2 \succ_{acspo} u^2$ and applying Lemma 5, we obtain $t^2\gamma \succ_{acspo} u^2\gamma = \hat{u}$, and hence, by Lemma 10, $t^2\gamma \succ_{acspo} u''$ for all $u'' \in EmbSmall(\hat{u})$.
  Therefore, we have $u^2\gamma = \hat{u}$, $t^2\gamma \succ_{acspo} u''$ for all $u'' \in EmbSmall(\hat{u})$ and the variables linearized in $t^2\gamma$ are the ones in $L_{\langle s,u \rangle}$.

  Moreover we have that $NoSmallHead(s) \gtrsim_{acspo} NoSmallHead(t) \succeq \succeq_{acspo} NoSmallHead(u)$ implies $L_{\langle s,u \rangle} \subseteq L_{\langle s,t \rangle}$:

  > For all variable in $x \in L_{\langle s,u \rangle}$ we have that x is an argument of u and $BigHead(s) \not\gg_{acspo} BigHead(u) \cup \{x\}$. Additionally, we have that $NoSmallHead(t) \gtrsim_{acspo} NoSmallHead(u)$ and $BigHead(t) \not\gg_{acspo} BigHead(u) \cup \{x\}$ (since $BigHead(s) \succeq \succeq_{acspo} BigHead(t)$) implies that x is an argument of t and since $BigHead(s) \not\gg_{acspo} BigHead(t) \cup \{x\}$ we conclude $x \in L_{\langle s,t \rangle}$.

  Let γ' be a substitution which maps all the variables in $L_{\langle s,t \rangle} \setminus L_{\langle s,u \rangle}$ to their original value. By Lemma 12, $s \succ_{acspo} t$ by case 5 implies $s^1 \succ_{acspo} t^1$ and applying Lemma 5 we obtain $s^1\gamma' \succ_{acspo} t^1\gamma'$ having that the variables linearized in $t^1\gamma'$ are the ones in $L_{\langle s,u \rangle}$.
  Therefore, we have $\hat{s} = s^1\gamma'$ and $t^1\gamma' = t^2\gamma$, and hence, $\hat{s} = s^1\gamma' \succ_{acspo} t^1\gamma' = t^2\gamma \succ_{acspo} u''$. Thus, by induction hypothesis we conclude $\hat{s} \succ_{acspo} u''$.

  Finally, we have that $NoSmallHead(s) \gtrsim_{acspo} NoSmallHead(t)$ and $NoSmallHead(t) \gtrsim_{acspo} NoSmallHead(u)$ by both, induction

hypothesis and transitivity of the multiset extension of a transitive relation, implies $NoSmallHead(s) \underline{\gg}_{acspo} NoSmallHead(u)$, and

- If either $s \succ_{acspo} t$ or $t \succ_{acspo} u$ hold by case 5a then by induction hypothesis and by compatibility of $\gg_{acspo}$ and $\underline{\gg}_{acspo}$ or by transitivity of the multiset extension of a transitive relation (if both hold by case 5a) we have $BigHead(s) \gg_{acspo} BigHead(u)$ and hence $s \succ_{acspo} u$ by case 5a.
- If either $s \succ_{acspo} t$ or $t \succ_{acspo} u$ hold by case 5b then we have $\#(s) > \#(t) \geq \#(u)$ or $\#(s) \geq \#(t) > \#(u)$. Thus, $\#(s) > \#(u)$, and hence $s \succ_{acspo} u$ by case 5b.
- Otherwise both $s \succ_{acspo} t$ and $t \succ_{acspo} u$ hold by case 5c, that is, $\#(s) \geq \#(t) \geq \#(u)$ and $\{s_1, ..., s_n\} \gg_{acspo} \{t_1, ..., t_m\} \gg_{acspo} \{u_1, ..., u_p\}$. Hence, $\#(s) \geq \#(u)$ and by the same arguments used above for $BigHead$, we have $\{s_1, ..., s_n\} \gg_{acspo} \{u_1, ..., u_p\}$. Therefore, $s \succ_{acspo} u$ by case 5c. □

Now we can prove that the ordering is well-founded. We have simply adapted the ususal Nash-Williams' proof based on finding a contradiction from the existance of a minimal infinite sequence.

Lemma 14. *$\succ_{acspo}$ is well-founded.*

Proof. Suppose it is not well-founded and let $s_1 \succ_{acspo} s_2 \succ_{acspo} ...s_i \succ_{acspo} ...$ be a minimal infinite sequence. We define a minimal sequence as follows: from all the infinite sequences we take the ones with the minimal $|s_1|$. From all the sequences with the same size for $s_1, \ldots, s_k$ we take the ones with minimal size for s_{k+1}.

First we prove that neither case 1 nor case 2 can be applied in any step of the infinite sequence. Suppose $s_k \succ_{acspo} s_{k+1}$ holds by case 1, that is, $s_{k_j} \succeq_{acspo} s_{k+1}$ for some s_{k_j} argument of s_k. By minimality of the initial sequence, s_k cannot be s_1. Then, since by Lemma 9 $s_{k-1} \succ_{acspo} s_k$ implies $s_{k-1} \succ_{acspo} s_{k_j}$, we have either $s_1 \succ_{acspo} ...s_{k-1} \succ_{acspo} s_{k_j} \succ_{acspo} s_{k+1} \succ_{acspo} ...$ (if $s_{k_j} \succ_{acspo} s_{k+1}$) or $s_1 \succ_{acspo} ...s_{k-1} \succ_{acspo} s_{k+1} \succ_{acspo} ...$ (if $s_{k_j} =_{AC} s_{k+1}$) which, in both cases, contradict the minimality of the initial sequence. If $s_k \succ_{acspo} s_{k+1}$ by case 2, then $s'_k \succeq_{acspo} s_{k+1}$ for some $s'_k \in EmbSmall(s_k)$ and $top(s_k) \in \mathcal{F}_{AC}$. Since by Lemma 10 $s_{k-1} \succ_{acspo} s_k$ implies $s_{k-1} \succ_{acspo} s'_k$, following the same arguments used in case 1 we would contradict the minimality of the initial sequence.

Additionally, we have that from one term s_k onwards $\succ_{acspo}$ cannot hold by case 3 since $\succ_{\mathcal{F}}$ and $\succ_q$ are well-founded.

Thus, we have $s_k \succeq_Q s_{k+1} \succeq_Q ...., top(s_k) =_{\mathcal{F}} top(s_{k+1}) =_{\mathcal{F}} \ldots$ and $s_i \succ_{acspo} s_{i+1}$ $(\forall i \geq k)$ holds either by 4 (if $top(s_i) \notin \mathcal{F}_{AC}$) or by 5 (if $top(s_i) \in \mathcal{F}_{AC}$).

- If $top(s_k) = f \notin \mathcal{F}_{AC}$ then we have $\{\boldsymbol{s}_k\} \gg_{acspo} \{\boldsymbol{s}_{k+1}\} \gg_{acspo} ...$, (let $\boldsymbol{s}$ be the arguments of s) but from this infinite sequence we could extract an infinite sequence $s_{k_i} \succ_{acspo} s_{k+1_j} \succ_{acspo} ...$ The existence of this sequence would contradict the minimality of the initial one (note that if $k > 1$ by Lemma 9 we have $s_{k-1} \succ_{acspo} s_{k_j}$).

– Otherwise $top(s) = f \in \mathcal{F}_{AC}$ and $s_i \succ_{acspo} s_{i+1}\ \forall i \geq k$ holds by case 5.
From one term onwards case 5a cannot be applied since otherwise we would have $BigHead(s_k) \underline{\succcurlyeq}_{acspo} BigHead(s_{k+1}) \underline{\succcurlyeq}_{acspo} ...$ (which has an infinite number of strict steps) and hence from some s_{k_i} we could extract an infinite sequence and (since by the subterm property of $\succ_{acspo}$, $s_{k-1} \succ_{acspo} s_{k_j}$) the minimality of the initial sequence would be contradicted.
Moreover, from one term onwards case 5b cannot be applied either, since infinite steps of 5b would imply the existence of an infinite sequence $n_1 \geq n_2 \geq ...$ with an infinite number of strict steps.
Therefore, from s_k onwards case 5c is applied. This fact implies that $\{\boldsymbol{s}_k\} \succ \succ_{acspo} \{\boldsymbol{s}_{k+1}\} \succcurlyeq_{acspo} ...$ which cannot exist for the same arguments used in case $f \notin \mathcal{F}_{AC}$. □

Next lemma is necessary for proving both the stability under substitutions and the partial monotonicity fulfilled by $\succ_{acspo}$.

Lemma 15. *$s = f(s_1, \ldots, s_n) \succeq_{acspo} t$ with $f \in \mathcal{F}_{AC}$ implies $f(u_1, \ldots, s_1, \ldots, s_n, \ldots, u_m) \succ_{acspo} t$.*

Proof. By Lemma 7, $f(u_1, \ldots, s_1, \ldots, s_n, \ldots, u_m) \succ_{acspo} f(s_1, \ldots, s_n)$. Therefore by transitivity and AC-compatibility of $\succ_{acspo}$, $f(1, \ldots, s_1, \ldots, s_n, \ldots, u_m) \succ_{acspo} t$. □

Now we will prove the partial monotonicity fulfilled by $\succ_{acspo}$, which requires to show first a previous lemma.

Lemma 16.

A. $s' \in EmbSmall(s)$ and $s' \succ_{acspo} t$ by case 1 implies $s \succ_{acspo} t$ by case 1

Proof. Let $s = f(s_1, \ldots, s_i, \ldots, s_n)$ and $s' = f(s_1, \ldots, tf_f(u), \ldots, s_n)$ with $s_i = h(\ldots u \ldots)$ and $f \succ_{\mathcal{F}} h$. $s' \succ_{acspo} t$ by case 1 implies either $s_j \succeq_{acspo} t$ for some $j \neq i$ and hence $s \succ_{acspo} t$ by case 1, or $u_k \succeq_{acspo} t$ for some u_k in $tf_f(u)$. Since, by case 1, $s_i \succ_{acspo} u$ and if $u = f(u_1, \ldots, u_p)$, by Lemma 9, $s_i \succ_{acspo} u_k$ for all $k = 1, \ldots, p$ then, by transitivity and AC-compatibility of $\succ_{acspo}$, we have $s_i \succ_{acspo} t$ and hence $s \succ_{acspo} t$ by case 1. □

B. $s \succ_{acspo} t$ by case 2 and $top(t) \succ_{\mathcal{F}} top(s)$ implies $s \succ_{acspo} t$ by case 1

Proof. By induction on the size of s. $s \succ_{acspo} t$ by case 2 implies $s' \succeq_{acspo} t$ for some $s' \in EmbSmall(s)$, and $top(t) \succ_{\mathcal{F}} top(s) = top(s')$, by definition of $\succ_{acspo}$, implies that $s' \succ_{acspo} t$ either by case 1 or by case 2. If $s' \succ_{acspo} t$ by case 2 then, since $top(t) \succ_{\mathcal{F}} top(s')$, by induction hypothesis, we have $s' \succ_{acspo} t$ by case 1. Thus, $s' \succ_{acspo} t$ by case 1 and hence, by A, we conclude $s \succ_{acspo} t$ by case 1. □

Lemma 17. *$s \succ_{acspo} t$ and $cs = f(\ldots, tf_f(s), \ldots) \succeq_Q f(\ldots, tf_f(t), \ldots) = ct$ implies $cs = f(\ldots, tf_f(s), \ldots) \succ_{acspo} f(\ldots, tf_f(t), \ldots) = ct$ for all terms s and t and for every flattened context $f(...[]...)$.*

Proof. By induction on $|cs| + |ct|$.

Consider first $f \notin \mathcal{F}_{AC}$. Then $f(...tf_f(s)...) = f(...s...)$ and $f(...tf_f(t)...) = f(...t...)$. Since $f(...s...) \succeq_Q f(...t...)$ and $s \succ_{acspo} t$ implies $\{...s...\} \gg_{acspo} \{...t...\}$ then we conclude $f(...s...) \succ_{acspo} f(...t...)$ by case 4.

Otherwise $f \in \mathcal{F}_{AC}$. Now, since we know that $cs \succeq_Q ct$, we distinguish whether $cs \succ_q ct$ or $cs \succeq_Q ct$ (and $cs \not\succ_q ct$).

If $cs \succ_q ct$ then we conclude $cs = f(...tf_f(s)...) \succ_{acspo} f(...tf_f(t)...) = ct$ by case 3 since $cs \succ_{acspo} u$ by case 1 for all u argument in the context, and $cs \succ_{acspo} w$ by case 1 for all w argument of $tf_f(t)$ since $s \succ_{acspo} t$ and by Lemma 9 $s \succ_{acspo} t_j$ for all t_j argument of t.

Otherwise $f \in \mathcal{F}_{AC}$, $cs \succeq_Q ct$ and $cs \not\succ_q ct$ (and hence, by AC-monotonicity of $\succ_q$, $s \not\succ_q t$).

We distinguish several cases according to the top symbol of s and t and the proof of $s \succ_{acspo} t$. Let $s = g_1(s_1...s_n)$ and $t = g_2(t_1...t_m)$ or $t \in \mathcal{X}$.

Consider first $f =_\mathcal{F} g_1$. If $s \succ_{acspo} t$ holds by case 1 then we conclude by induction hypothesis and Lemma 15. If $s \succ_{acspo} t$ holds by case 2 we conclude $cs \succ_{acspo} ct$ by induction hypothesis and case 2. Additionally note that if $top(t) \succ_\mathcal{F} top(s)$ then, by definition of $\succ_{acspo}$ and Lemma 16, we have that $s \succ_{acspo} t$ by case 1.

Otherwise we have $f =_\mathcal{F} g_1 \succ_\mathcal{F} g_2$ or ($f =_\mathcal{F} g_1 =_\mathcal{F} g_2$ and $s \succ_{acspo} t$ by case 5) or $f \neq g_1$ (recall that we are in the case $s \not\succ_q t$).

Consider now $f \succ_\mathcal{F} g_1$ and $g_2 \succ_\mathcal{F} g_1$. By Lemma 16 $s \succ_{acspo} t$ by case 1 and hence $s_i \succeq_{acspo} t$. By induction hypothesis $f(...tf_f(s_i)...) \succeq_{acspo} f(...tf_f(t)...)$ and since $f \succ_\mathcal{F} top(s)$ implies $f(...tf_f(s_i)...) \in EmbSmall(cs)$ then we conclude $cs \succ_{acspo} ct$ by case 2.

Thus we only have to consider the cases (A) $g_1 \succ_\mathcal{F} f$, or (B) $f \succ_\mathcal{F} g_1 \succeq_\mathcal{F} g_2$, or (C) $f =_\mathcal{F} g_1 \succ_\mathcal{F} g_2$, or (D) ($f =_\mathcal{F} g_1 =_\mathcal{F} g_2$ and $s \succ_{acspo} t$ by case 5). We will prove that $cs \succ_{acspo} ct$ by case 5.

First we prove that $\widehat{cs} \succ_{acspo} u$ for all $u \in EmbSmall(\widehat{ct})$.

> For any u embedded through a term in the context $u = f(\dots tf_f(w) \dots \widehat{tf_f(t)} \dots)$ there is $u' = f(\dots tf_f(w) \dots \widehat{tf_f(s)} \dots) \in EmbSmall(\widehat{cs})$. Therefore, by induction hypothesis, $u' \succ_{acspo} u$ and hence $\widehat{cs} \succ_{acspo} u$ by case 2.
>
> Now we will prove $\widehat{cs} \succ_{acspo} u$ for all u embedded through a term in $\widehat{tf_f(t)}$.
>
> Consider first $u = f(\dots, tf_f(t_i), \dots)$ with $t = g_2(\dots, t_i, \dots)$ and $f \succ_\mathcal{F} g_2$. Since $s \succ_{acspo} t$ implies $s \succ_{acspo} t_i$ then, by induction hypothesis we obtain $\widehat{cs} \succ_{acspo} u$.
>
> Otherwise $g_2 =_\mathcal{F} f$ and hence neither case (B) nor case (C) hold. Thus, we have $g_1 \succ_\mathcal{F} g_2 =_\mathcal{F} f$ or ($g_1 =_\mathcal{F} g_2 =_\mathcal{F} f$ and $s \succ_{acspo} t$ by case 5). If $top(s) = g_1 \succ_\mathcal{F} f$ then none of $t_1, \dots, t_m$ need to be renamed and hence $u = f(\dots, tf_f(t'), \dots)$ and $t' \in EmbSmall(t)$. Therefore, we conclude $\widehat{cs} \succ_{acspo} u$ by induction hypothesis since $s \succ_{acspo} t$ implies $s \succ_{acspo} t' \in EmbSmall(t)$ by Lemma 10. Otherwise $top(s) =_\mathcal{F} top(t) =_\mathcal{F} f$

and $s \succ_{acspo} t$ by case 5 that implies $\hat{s} \succ_{acspo} t' \in EmbSmall(\hat{t})$, and hence, by induction hypothesis, $f(\widehat{...}tf_f(\hat{s})\widehat{...}) \succ_{acspo} f(\widehat{...}tf_f(t')\widehat{...})$. By definition of $L_{\langle cs,ct\rangle}$ and $L_{\langle s,t\rangle}$, from the facts $NoSmallHead(s) \underline{\succcurlyeq}_{acspo} NoSmallHead(t)$, $BigHead(s) \underline{\succcurlyeq}_{acspo} BigHead(t)$, $NoSmallHead(cs) = NoSmallHead(s) \cup W$, $NoSmallHead(ct) = NoSmallHead(t) \cup W$, and $BigHead(cs) = BigHead(s) \cup V$ and $BigHead(ct) = BigHead(t) \cup V$, follows that for all x argument of t, $x \in L_{\langle cs,ct\rangle}$ if and only if $x \in L_{\langle s,t\rangle}$. Therefore $\widehat{cs} = f(\widehat{...}tf_f(\hat{s})\widehat{...}) \succ_{acspo} f(\widehat{...}tf_f(t')\widehat{...}) = u$.

Finally, we distinguish cases (A),(B),(C) and (D).

1. If (A) $g_1 \succ_{\mathcal{F}} f$ (with $t \in \mathcal{X}$ or $t \notin \mathcal{X}$) then $tf_f(s) = s$ and since $s \in BigHead(cs)$ we have $NoSmallHead(cs) \underline{\succcurlyeq}_{acspo} NoSmallHead(ct)$ and $BigHead(cs) \succcurlyeq_{acspo} BigHead(ct)$. Therefore, $cs \succ_{acspo} ct$ by 5a.
2. If (B) $f \succ_{\mathcal{F}} g_1 \succeq_{\mathcal{F}} g_2$ then $cs \succ_{acspo} ct$ by 5c since $NoSmallHead(cs) = NoSmallHead(ct)$, $BigHead(cs) = BigHead(ct)$, $\#(cs) = \#(ct)$ and $s \succ_{acspo} t$ implies $\{\ldots, s, \ldots\} \succcurlyeq_{acspo} \{\ldots, t, \ldots\}$.
3. If (C) $g_1 =_{\mathcal{F}} f \succ_{\mathcal{F}} g_2$ then we have $tf_f(s) = s_1, \ldots, s_n$ and since $f \succ_{\mathcal{F}} g_2$ implies $t \notin NoSmallHead(ct)$ and $t \notin BigHead(ct)$ then we conclude $cs \succ_{acspo} ct$ by case 5a if some of $s_1 \ldots s_n$ belongs to $BigHead(cs)$ or by case 5b otherwise.
4. If (D) $g_1 =_{\mathcal{F}} f =_{\mathcal{F}} g_2$ and $s \succ_{acspo} t$ by case 5 then $tf_f(s) = s_1, \ldots, s_n$ and $tf_f(t) = t_1, \ldots, t_m$. From $NoSmallHead(cs) = NoSmallHead(s) \cup W$, $NoSmallHead(ct) = NoSmallHead(t) \cup W$ and $NoSmallHead(s) \underline{\succcurlyeq}_{acspo} NoSmallHead(t)$ we obtain $NoSmallHead(cs) \underline{\succcurlyeq}_{acspo} NoSmallHead(ct)$. We also have that $BigHead(cs) = BigHead(s) \cup V$ and $BigHead(ct) = BigHead(t) \cup V$. Therefore, if $s \succ_{acspo} t$ by case 5a then we conclude $cs \succ_{acspo} ct$ by case 5a, if $s \succ_{acspo} t$ by case 5b then we conclude $cs \succ_{acspo} ct$ by case 5b, and if $s \succ_{acspo} t$ by case 5c then we conclude $cs \succ_{acspo} ct$ by case 5c.

□

Now we will prove the stability under substitutions of $\succ_{acspo}$. The proof is built by distinguishing two different kinds of substitutions. In the following lemma we consider substitutions from variables to variables and in Lemma 20 we will consider substitutions from variables to terms like $q(y_1, \ldots, y_r)$ for some symbol q and new variables $y_1, \ldots, y_r$. Using these two lemmas we will finally prove the stability of $\succ_{acspo}$ in Lemma 21.

Lemma 18. *Let $\sigma = \{x \mapsto y\}$ for some $y \in \mathcal{X}$. $s \succ_{acspo} t$ by case (i) implies $s\sigma \succ_{acspo} t\sigma$ by case (i).*

Proof. By induction on the sum of sizes of s and t.

- If $s \succ_{acspo} t$ by case 1 then $s_i \succeq_{acspo} t$ for some s_i argument of s. By induction hypothesis and stability of $=_{AC}$, $s_i\sigma \succeq_{acspo} t\sigma$ and hence $s\sigma \succ_{acspo} t\sigma$ by case 1.

- If $s \succ_{acspo} t$ by case 2 then $s' \succeq_{acspo} t$ for some $s' \in EmbSmall(s)$. By induction hypothesis and stability of $=_{AC}$, $s'\sigma \succeq_{acspo} t\sigma$ and hence $s\sigma \succ_{acspo} t\sigma$ by case 2.
- If $s = f(s_i, \ldots, s_n) \succ_{acspo} g(t_1, \ldots, t_m) = t$ by case 3 then $s \succ_{acspo} t_i$ for all $i = 1, \ldots, m$ and either $f \succ_{\mathcal{F}} g$ or ($f =_{\mathcal{F}} g$ and $s \succ_q t$). By stability of $\succ_q$, $s\sigma \succ_q t\sigma$ and by induction hypothesis $s\sigma \succ_{acspo} t_i\sigma$. Therefore, $s\sigma \succ_{acspo} t\sigma$ by case 3.
- If $s \succ_{acspo} t$ by case 4 then $top(s) =_{\mathcal{F}} top(t) \notin \mathcal{F}_{AC}$, $s \succeq_Q t$ and $\{s_1, ..., s_n\} \succ \succ_{acspo} \{t_1, ..., t_n\}$. By stability of $\succeq_Q$, $s\sigma \succeq_Q t\sigma$ and, by induction hypothesis and stability of $=_{AC}$, $\{s_1\sigma, ..., s_n\sigma\} \succcurlyeq_{acspo} \{t_1\sigma, ..., t_n\sigma\}$. Thus, $s\sigma \succ_{acspo} t\sigma$ by case 4.
- If $s \succ_{acspo} t$ by case 5 then $s \succeq_Q t$, $top(s) =_{\mathcal{F}} top(t) \in \mathcal{F}_{AC}$, $\hat{s} \succ_{acspo} t'$ for all $t' \in EmbSmall(\hat{t})$, $NoSmallHead(s) \underline{\succcurlyeq}_{acspo} NoSmallHead(t)$ and either $BigHead(s) \succcurlyeq_{acspo} BigHead(t)$ or $\#(s) > \#(t)$ or $\#(s) \geq \#(t)$ and $\{s_1, ..., s_n\} \succcurlyeq_{acspo} \{t_1, ..., t_m\}$.
  Since $top(s\sigma) = top(s)$ and $top(t\sigma) = top(t)$, we have $top(s\sigma) =_{\mathcal{F}} top(t\sigma) \in \mathcal{F}_{AC}$. By stability of $\succeq_Q$, $s \succeq_Q t$ implies $s\sigma \succeq_Q t\sigma$.
  Now we will prove that $\widehat{s\sigma} \succ_{acspo} u$ for all $u \in EmbSmall(\widehat{t\sigma})$. We distinguish whether $x \in L_{\langle s,t \rangle}$ or $x \notin L_{\langle s,t \rangle}$.
  - Suppose $x \in L_{\langle s,t \rangle}$ and let $u_1, ..., u_k$ be the renamed versions of x in $\langle \hat{s}, \hat{t} \rangle$. Let $\gamma = \{u_1 \mapsto y, ..., u_k \mapsto y\}$ if $y \notin L_{\langle s\sigma, t\sigma \rangle}$ or the null substitution otherwise. By Lemma 5, $\hat{s} \succ_{acspo} t'$ implies $\hat{s}\gamma \succ_{acspo} t'\gamma$, and by induction hypothesis, $\hat{s}\gamma\sigma \succ_{acspo} t'\gamma\sigma$. Since $\hat{s}\gamma\sigma = \widehat{s\sigma}$, $\hat{t}\gamma\sigma = \widehat{t\sigma}$ and for all $u \in EmbSmall(\widehat{t\sigma})$ $u = t'\gamma\sigma$ for some $t' \in EmbSmall(\hat{t})$ then we conclude $\widehat{s\sigma} \succ_{acspo} u$ for all $u \in EmbSmall(\widehat{t\sigma})$.
  - Suppose $x \notin L_{\langle s,t \rangle}$.
    Consider first that there is not any $t_i = x$ argument of t. Let γ be the substitution which deslinearizes y if $y \in L_{\langle s,t \rangle}$ and $y \notin L_{\langle s\sigma, t\sigma \rangle}$, or the null substitution otherwise. Then we have that $\hat{s}\gamma\sigma = \widehat{s\sigma}$, $\hat{t}\gamma\sigma = \widehat{t\sigma}$, and for all $u \in EmbSmall(\widehat{t\sigma})$, $u = t'\gamma\sigma$ for some $t' \in EmbSmall(\hat{t})$. We know that $\hat{s} \succ_{acspo} t'$ for all $t' \in EmbSmall(\hat{t})$ and, by Lemma 5, $\hat{s} \succ_{acspo} t'$ implies $\hat{s}\gamma \succ_{acspo} t'\gamma$. Since, by induction hypothesis, $\hat{s}\gamma\sigma \succ_{acspo} t'\gamma\sigma$ then we are done.
    Otherwise there is some $t_i = x$. Now we prove that $y \notin L_{\langle s\sigma, t\sigma \rangle}$.
    $x \notin L_{\langle s,t \rangle}$ and x argument of t implies that $BigHead(s) \succcurlyeq_{acspo} BigHead(t) \cup \{x\}$. By induction hypothesis and by stability of the multiset extension of an stable (from variables to variables) relation $BigHead(s\sigma) \succcurlyeq_{acspo} BigHead(t\sigma) \cup \{y\}$. Thus, $y \notin L_{\langle s\sigma, t\sigma \rangle}$.
    Now we can prove $\widehat{s\sigma} \succ_{acspo} u$ for all $u \in EmbSmall(\widehat{t\sigma})$ exactly as done for the previous case (when there is not any x argument of t).
  Therefore, we have proved $\widehat{s\sigma} \succ_{acspo} u$ for all $u \in EmbSmall(\widehat{t\sigma})$.
  Besides, $NoSmallHead(s) \underline{\succcurlyeq}_{acspo} NoSmallHead(t)$, by induction hypothesis and by stability of the multiset extension of an stable (from variables to variables) relation, implies $NoSmallHead(s)\sigma \underline{\succcurlyeq}_{acspo} NoSmallHead(t)\sigma$. Therefore, since we have that $NoSmallHead(s\sigma) = NoSmallHead(s)\sigma$ and

$NoSmallHead(t\sigma) = NoSmallHead(t)\sigma$, we conclude $NoSmallHead(s\sigma) \succeq$ $\succeq_{acspo} NoSmallHead(t\sigma)$.
Finally, if we have that $BigHead(s) \gg_{acspo} BigHead(t)$, as $BigHead(s\sigma) = BigHead(s)\sigma$ and $BigHead(t\sigma) = BigHead(t)\sigma$ then, by induction hypothesis and by stability of the multiset extension of an stable (from variables to variables) relation, we have $BigHead(s\sigma) \gg_{acspo} BigHead(t\sigma)$ and hence $s\sigma \succ_{acspo} t\sigma$ by case 5a.
If $\#(s) > \#(t)$ then $\#(s\sigma) > \#(t\sigma)$ and hence $s\sigma \succ_{acspo} t\sigma$ by case 5b.
Otherwise, $\#(s) \geq \#(t)$ and hence $\#(s\sigma) \geq \#(t\sigma)$, and $\{s_1...s_n\} \gg\!\!=_{acspo} \{t_1...t_m\}$ that, by induction hypothesis and by stability of the multiset extension of an stable (from variables to variables) relation, implies $\{s_1\sigma...s_n\sigma\} \succeq$ $\succeq_{acspo} \{t_1\sigma...t_m\sigma\}$ and hence, $s\sigma \succ_{acspo} t\sigma$ by case 5c. □

Next lemma is necessary for the proof of Lemma 20, which, as said, is the main lemma for proving stability under substitutions.

Lemma 19. *Let* $s = f(s_1, \ldots, s_n)$, $t = g(t_1, \ldots, t_m)$ *and* $\langle \hat{s}, \hat{t} \rangle = lin(s, t, L_{\langle s,t \rangle})$ *with* $f =_{\mathcal{F}} g \in \mathcal{F}_{AC}$, $s \succeq_Q t$, $NoSmallHead(s) \gg_{acspo} NoSmallHead(t)$, $BigHead(s) \gg_{acspo} BigHead(t) \cup \{t_{i_1}, \ldots, t_{i_k}\}$ *with* $t_{i_1}, \ldots, t_{i_k}$ *arguments of* t *and* $f \succ_{\mathcal{F}} top(t_{i_j})$ *for all* $j = 1, \ldots k$, *and*
$\hat{s} \succ_{acspo} t' \in EmbSmall(\hat{t})$ *for all* t' *embedded through an argument* $t_q \notin \{t_{i_1}, \ldots, t_{i_k}\}$.

Then $\hat{s} \succ_{acspo} u$ *for all* $u \in M(\hat{t}, \{i_1, \ldots, i_k\})$ *where* $M(g(t_1, \ldots, t_m), S) = \{g(t'_1, \ldots, t'_m) \mid t'_j = t_j$ *if* $j \notin S$ *and*
$t'_j = t_j \cup \{t_{j_1}, \ldots, t_{j_r}\}$ *with* $t_j = h(t_{j_1}, \ldots, t_{j_r})$ *if* $j \in S\} \setminus \{g(t_1, \ldots, t_m)\}$.

Proof. Let u be any element in $M(\hat{t}, \{i_1, \ldots, i_k\})$. We will prove that $\hat{s} \succ_{acspo} u$ by case 5a. We will use induction on the size of u.

Since $top(\hat{s}) = top(s)$ and $top(\hat{t}) = top(t) = top(u)$, we have that $top(\hat{s}) =_{\mathcal{F}} top(u)$.

By condition (R2) $s \succeq_Q t$ implies $\hat{s} \succeq_Q \hat{t}$, and by Property 3 applied several times and by transitivity of $\succeq_Q$ we also have $\hat{t} \succeq_Q u$ and hence $\hat{s} \succeq_Q u$.

Now we prove that $NoSmallHead(\hat{s}) \gg\!\!=_{acspo} NoSmallHead(u)$. By definition of $BigHead$ and $NoSmallHead$, we have that $NoSmallHead(s) \gg_{acspo} NoSmallHead(t)$ and $BigHead(s) \gg_{acspo} BigHead(t) \cup \{t_{i_1}, \ldots, t_{i_k}\}$ imply $NoSmallHead(s) \gg_{acspo} NoSmallHead(t) \cup \{t_{i_1}, \ldots, t_{i_k}\}$. By definition of $\langle \hat{s}, \hat{t} \rangle$ and since $\{t_{i_1}, \ldots, t_{i_k}\}$ are not variables, we also have $NoSmallHead(\hat{s}) \succ$ $\succ_{acspo} NoSmallHead(\hat{t}) \cup \{t_{i_1}, \ldots, t_{i_k}\}$. Additionally, by the subterm property of $\succ_{acspo}$, $NoSmallHead(\hat{t}) \cup \{t_{i_1}, \ldots, t_{i_k}\} \gg_{acspo} NoSmallHead(u)$ and hence, we conclude $NoSmallHead(\hat{s}) \gg_{acspo} NoSmallHead(u)$.

We also have $BigHead(s) = BigHead(\hat{s})$ and $BigHead(t) = BigHead(\hat{t})$, and hence $BigHead(\hat{s}) \gg_{acspo} BigHead(\hat{t}) \cup \{t_{i_1}, \ldots, t_{i_k}\}$, and since, by definition of u, $BigHead(\hat{t}) \cup \{t_{i_1}, \ldots, t_{i_k}\} \gg_{acspo} BigHead(u)$ then we conclude $BigHead(\hat{s}) \gg_{acspo} BigHead(u)$.

Finally we have to prove $\hat{s} \succ_{acspo} u' \in EmbSmall(\hat{u})$. First we will show that if $\langle \hat{s}, \hat{t} \rangle = lin(s, t, L_{\langle s,t \rangle})$ then $\langle \hat{s}, \hat{u} \rangle = lin(s, u, L_{\langle s,u \rangle})$ and $\hat{u} = u$. By assumption

we have that $BigHead(s) \gg_{acspo} BigHead(t) \cup \{t_{i_1}, \ldots, t_{i_k}\}$. Therefore, by the subterm property of $\succ_{acspo}$ and by transitivity of $\gg_{acspo}$, it also holds that $BigHead(s) \gg_{acspo} BigHead(t) \cup Q$ where Q is a set of subterms (not necessarily proper) of $\{t_{i_1}, \ldots, t_{i_k}\}$, and hence, by the form of the elements u in $M(\hat{t}, \{i_1, \ldots, i_k\})$, $L_{\langle s,t \rangle} = L_{\langle s,u \rangle}$ and $u = \hat{u}$.

Therefore, for all $u' \in EmbSmall(\hat{u})$ either $u' \in M(\hat{t}, \{i_1, \ldots, i_k\})$ and hence, by induction hypothesis, $\hat{s} \succ_{acspo} u'$ or there exists $t^1 \in EmbSmall(\hat{t})$ (through an argument t_q with q not in $\{i_1, \ldots, i_k\}$) and $t^2, \ldots, t^j$ such that $t^2 \in EmbSmall(t^1), \ldots t^j \in EmbSmall(t^{j-1})$ and $u' \in EmbSmall(t^j)$. By assumption $\hat{s} \succ_{acspo} t^1$ and, by applying case 2 in every step, we have $t^1 \succ_{acspo} t^2 \succ_{acspo} \ldots \succ_{acspo} t^j \succ_{acspo} u'$ and hence, by transitivity of $\succ_{acspo}$, $\hat{s} \succ_{acspo} u'$. □

Lemma 20. *Let σ be $\{x \mapsto q(y_1, \ldots, y_r)\}$ for some symbol q and new variables $y_1, \ldots, y_r$. If $s \succ_{acspo} t$ then $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$.*

Proof. By induction on $|\overline{s\sigma} + \overline{t\sigma}|$. We distinguish several cases according to the proof of $s \succ_{acspo} t$.

- $s = f(s_1, \ldots, s_n) \succ_{acspo} t$ by case 1, i.e. $s_i \succeq_{acspo} t$ for some s_i argument of s. If $f \in \mathcal{F}_{AC}$, $s_i = x$ and $x\sigma = q(y_1, \ldots, y_r)$ with $q =_{\mathcal{F}} f$ then $t = x$ and since the arity of f is greater than 1, by Lemma 15, $\overline{s\sigma} = f(..., y_1, \ldots, y_r, ...) \succ_{acspo} f(y_1, \ldots, y_r) = \overline{t\sigma}$. Otherwise, by induction hypothesis and stability under substitutions of $=_{AC}$, $\overline{s_i\sigma} \succeq_{acspo} \overline{t\sigma}$ and hence, $\overline{s\sigma} = f(..., \overline{s_i\sigma}, ...) \succ_{acspo} \overline{t\sigma}$, by case 1.
- $s = f(s_1, \ldots, s_n) \succ_{acspo} t$ by case 2, i.e. $s' = f(s_1, ..., tf_f(v), ..., s_n) \succeq_{acspo} t$ with $s' \in EmbSmall(s)$, $s_i = h(...v...)$ and $f \succ_{\mathcal{F}} h$. By induction hypothesis $\overline{s'\sigma} \succeq_{acspo} \overline{t\sigma}$. If $h \in \mathcal{F}_{AC}$, $v = x$ and $x\sigma = q(y_1, \ldots, y_r)$ with $q =_{\mathcal{F}} h$ then we have $\overline{s_i\sigma} = h(..., y_1, ..., y_r, ...)$, and by Lemma 7, $\overline{s_i\sigma} = h(..., y_1, ..., y_r, ...) \succ_{acspo} h(y_1, ..., y_r) = x\sigma$.
  Since $h \in \mathcal{F}_{AC}$ implies $h(..., y_1, ..., y_r, ...) \succeq_Q h(y_1, ..., y_r)$ then, by AC-monotonicity of $\succeq_Q$, we also have that $\overline{s\sigma} = f(..., h(..., y_1, ..., y_r, ...), ...) \succeq_Q f(..., h(y_1, ..., y_r), ...) = \overline{s'\sigma}$, and since $h(..., y_1, ..., y_r, ...) \succ_{acspo} h(y_1, ..., y_r)$, by Lemma 17, we obtain $\overline{s\sigma} \succ_{acspo} \overline{s'\sigma}$. Thus, by transitivity and AC-compatibility of $\succ_{acspo}$, we conclude $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$.
  Otherwise $\overline{s'\sigma} \in EmbSmall(\overline{s\sigma})$, and hence we conclude $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$ by case 2.
- $s = f(s_1, \ldots, s_n) \succ_{acspo} t = g(t_1, \ldots, t_m)$ by case 3, i.e. $s \succ_{acspo} t_i$ for all $i = 1, \ldots, m$ and either $f \succ_{\mathcal{F}} g$ or ($f =_{\mathcal{F}} g$ and $s \succ_q t$). By induction hypothesis, $\overline{s\sigma} \succ_{acspo} \overline{t_i\sigma}$ for all $i = 1, \ldots, m$. If $g \in \mathcal{F}_{AC}$, $t_i = x$ and $g =_{\mathcal{F}} q = top(x\sigma)$, then, by Lemma 9, $\overline{s\sigma} \succ_{acspo} y_i$ for all y_i argument of $x\sigma = q(y_1, \ldots, y_r)$. Therefore we have $\overline{s\sigma} \succ_{acspo} \overline{t'_i\sigma}$ for all $t'_i\sigma$ argument of $\overline{t\sigma}$. Finally, since by stability under substitutions of $\succ_q$, $s \succ_q t$ implies $\overline{s\sigma} \succ_q \overline{t\sigma}$ then we have that either $f \succ_{\mathcal{F}} g$ or ($f =_{\mathcal{F}} g$ and $\overline{s\sigma} \succ_q \overline{t\sigma}$). Thus, $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$ by case 3.
- $s = f(s_1 ... s_n) \succ_{acspo} g(t_1 ... t_m) = t$ by case 4, i.e. $f =_{\mathcal{F}} g \notin \mathcal{F}_{AC}$, $s \succeq_Q t$ and $\{s_1, ..., s_n\} \gg_{acspo} \{t_1, ..., t_n\}$. By stability of $\succeq_Q$ we have $\overline{s\sigma} \succeq_Q \overline{t\sigma}$

and by induction hypothesis and by definition of the multiset extension, $\{\overline{s_1\sigma}, ..., \overline{s_n\sigma}\} \gtrdot_{acspo} \{\overline{t_1\sigma}, ..., \overline{t_n\sigma}\}$. Therefore, $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$ by case 4.

– $s = f(s_1...s_n) \succ_{acspo} g(t_1...t_m) = t$ by case 5.
Since $top(s) = top(\overline{s\sigma})$ and $top(t) = top(\overline{t\sigma})$ and $top(s) = f =_{\mathcal{F}} g = top(t) \in \mathcal{F}_{AC}$ then we also have $top(\overline{s\sigma}) =_{\mathcal{F}} top(\overline{t\sigma}) \in \mathcal{F}_{AC}$. By stability of $\succeq_Q$, $s \succeq_Q t$ implies $\overline{s\sigma} \succeq_Q \overline{t\sigma}$.
Now we will prove $NoSmallHead(\overline{s\sigma}) \gtrdot_{acspo} NoSmallHead(\overline{t\sigma})$. We have $NoSmallHead(s) = S \cup X$ with X the maximal subset of $NoSmallHead(s)$ containing x, and $NoSmallHead(t) = T \cup X'$ with X' the maximal subset of $NoSmallHead(t)$ containing x. Remark that $NoSmallHead(s) \gtrdot_{acspo} NoSmallHead(t)$ implies that $S \gtrdot_{acspo} T \cup (X' \setminus X)$ and hence, by induction hypothesis $\overline{S\sigma} \gtrdot_{acspo} \overline{T\sigma} \cup (X'\sigma \setminus X\sigma)$. If $q \succ_{\mathcal{F}} f$ then we have that $\overline{NoSmallHead(s)\sigma} = NoSmallHead(\overline{s\sigma})$ and $\overline{NoSmallHead(t)\sigma} = NoSmallHead(\overline{t\sigma})$, and as, by induction hypothesis, $\overline{NoSmallHead(s)\sigma} \succeq \succeq_{acspo} \overline{NoSmallHead(t)\sigma}$ then we are done. If $f \succ_{\mathcal{F}} q$ then we have that $NoSmallHead(\overline{s\sigma}) = \overline{S\sigma}$ and $NoSmallHead(\overline{t\sigma}) = \overline{T\sigma}$ and since $\overline{S\sigma} \succeq \succeq_{acspo} \overline{T\sigma}$ then we are done. Otherwise $f =_{\mathcal{F}} q$. Then $NoSmallHead(\overline{s\sigma}) = \overline{S\sigma} \cup Y$ where $Y = \bigcup_{x \in X}\{y_1, ..., y_r\}$, $NoSmallHead(\overline{t\sigma}) = \overline{T\sigma} \cup Y'$ where $Y' = \bigcup_{x \in X'}\{y_1, ..., y_r\}$, and since $\overline{S\sigma} \gtrdot_{acspo} \overline{T\sigma} \cup (X'\sigma \setminus X\sigma)$ and, by the subterm property, $\overline{T\sigma} \cup (X'\sigma \setminus X\sigma) \gtrdot_{acspo} \overline{T\sigma} \cup (Y' \setminus Y)$, then we conclude $NoSmallHead(\overline{s\sigma}) \gtrdot_{acspo} NoSmallHead(\overline{t\sigma})$.

We have to prove that $\widehat{\overline{s\sigma}} \succ_{acspo} u \in EmbSmall(\widehat{\overline{t\sigma}})$. We distinguish whether u is a term embedded through a subterm which is non-variable in $\hat{t}$ or u is embedded through a term $t_i\sigma = q(y_1, \ldots, y_r)$ with $t_i \in \mathcal{X}$. Let $\hat{t} = g(t'_1, \ldots, t'_m)$.
First we will show the relation between $L_{\langle s,t\rangle}$ and $L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle}$. If $x \notin L_{\langle s,t\rangle}$ then either x does not occur as argument of t, and hence $L_{\langle s,t\rangle} = L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle}$, or $BigHead(s) \gtrdot_{acspo} BigHead(t) \cup \{x\}$. If $top(t) = g \neq_{\mathcal{F}} q$ then we have $x, y_1...y_r \notin L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle}$. If $g =_{\mathcal{F}} q$ then since $x\sigma = q(y_1, ..., y_r)$ implies $BigHead(\overline{s\sigma}) = \overline{BigHead(s)\sigma}$, $BigHead(\overline{t\sigma}) = \overline{BigHead(t)\sigma}$ and, by induction hypothesis, $\overline{BigHead(s)\sigma} \gtrdot_{acspo} \overline{BigHead(t)\sigma} \cup \{q(y_1...y_r)\}$ and $q(y_1...y_r) \succ_{acspo} y_k$ for all $k = 1...r$ then we conclude $x, y_1...y_r \notin L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle}$. Therefore, we conclude that $x \notin L_{\langle s,t\rangle}$ implies $L_{\langle s,t\rangle} = L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle}$. If $x \in L_{\langle s,t\rangle}$ then if $top(t) = g \neq_{\mathcal{F}} q$ we have $L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle} = L_{\langle s,t\rangle} \setminus \{x\}$, and if $top(t) = g =_{\mathcal{F}} q$ then $L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle} = (L_{\langle s,t\rangle} \setminus \{x\}) \cup \{y_1, ..., y_r\}$ since $BigHead(s) \not\gtrdot_{acspo} BigHead(t) \cup \{x\}$ and $x\sigma = q(y_1, ..., y_r)$ with $y_1, ..., y_r$ new variables imply $BigHead(\overline{s\sigma}) \not\gtrdot_{acspo} BigHead(\overline{t\sigma}) \cup \{y_1, ..., y_r\}$.

- u is embedded through a term $t'_i\sigma = h(...v\sigma...)$ such that $t'_i = h(...v...)$ and $g \succ_{\mathcal{F}} h$. Since $s \succ_{acspo} t$ by case 5 then we have $\hat{s} \succ_{acspo} w = g(t'_1...tf_f(v)...t'_m) \in EmbSmall(\hat{t})$.
  As seen above, $x \notin L_{\langle s,t\rangle}$ implies $y_1, ..., y_r \notin L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle}$ and hence $\widehat{\overline{s\sigma}} = \overline{\hat{s}\sigma}$ and $\widehat{\overline{t\sigma}} = \overline{\hat{t}\sigma}$ and $u = \overline{w\sigma}$. Since by induction hypothesis, $\hat{s} \succ_{acspo} w$ implies $\overline{\hat{s}\sigma} \succ_{acspo} \overline{w\sigma}$ then we conclude $\widehat{\overline{s\sigma}} \succ_{acspo} u$.

Otherwise $x \in L_{\langle s,t\rangle}$. If $y_1, ..., y_r \notin L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle}$ then let δ_0 be the substitution which deslinearizes x in $\langle \hat{s}, \hat{t}\rangle$. Then we have $\widehat{\overline{s\sigma}} = \overline{\hat{s}\delta_0\sigma}$ and $\widehat{\overline{t\sigma}} = \overline{\hat{t}\delta_0\sigma}$. By Lemma 5 $\hat{s} \succ_{acspo} w$ implies $\hat{s}\delta_0 \succ_{acspo} w\delta_0$ and hence, by induction hypothesis $\overline{\hat{s}\delta_0\sigma} \succ_{acspo} \overline{w\delta_0\sigma}$. Since $\overline{w\delta_0\sigma} \in EmbSmall(\overline{\hat{t}\delta_0\sigma})$ and $u = \overline{w\delta_0\sigma}$ then we conclude $\overline{s\sigma} \succ_{acspo} u$.
Otherwise $x \in L_{\langle s,t\rangle}$ and $y_1, ..., y_r \in L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle}$ and hence $g =_{\mathcal{F}} q$. Let $x^1, ..., x^l$ be the renamed versions of x in $\langle \hat{s}, \hat{t}\rangle$ and let $\delta^1 = \{x^1 \mapsto q(y_1^1, ...y_r^1)\}$,...,$\delta^l = \{x^l \mapsto q(y_1^l, ...y_r^l)\}$ and $\delta = \delta^1 \cup \ldots \cup \delta^l$. By applying induction hypothesis l times, $\hat{s} \succ_{acspo} w$ implies $\overline{\hat{s}\delta} \succ_{acspo} \overline{w\delta}$ and $\overline{w\delta} \in EmbSmall(\overline{\hat{t}\delta})$. By induction hypothesis $\overline{\hat{s}\delta} \succ_{acspo} \overline{w\delta}$ implies $\overline{\overline{\hat{s}\delta}\sigma} \succ_{acspo} \overline{\overline{w\delta}\sigma} \in EmbSmall(\overline{\overline{\hat{t}\delta}\sigma})$ and, since $\widehat{\overline{s\sigma}} = \overline{\overline{\hat{s}\delta}\sigma}$, $\widehat{\overline{t\sigma}} = \overline{\overline{\hat{t}\delta}\sigma}$ (up to renaming of variables) and $u = \overline{\overline{w\delta}\sigma}$, we conclude $\widehat{\overline{s\sigma}} \succ_{acspo} u$.

- u is embedded through $t_i\sigma = q(y_1, \ldots, y_r)$ such that $t_i = x$. Thus, $g \succ_{\mathcal{F}} q$ and hence $y_1, \ldots, y_r \notin L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle}$. Therefore we have that $\overline{t\sigma} = g(\overline{t_1\sigma} \ldots, q(y_1, \ldots, y_r), \ldots \overline{t_m\sigma})$ and $\widehat{\overline{t\sigma}} = g(w_1 \ldots, q(y_1, \ldots, y_r), \ldots w_m)$ where $w_i = \overline{t_i\sigma}$ if $w_i \notin L_{\langle \overline{s\sigma}, \overline{t\sigma}\rangle}$, or w_i is a new variable obtained when linearizing $\langle \widehat{\overline{s\sigma}}, \widehat{\overline{t\sigma}}\rangle$. Thus, $u = g(w_1, \ldots, y_k, \ldots, w_m)$ for some $k = 1, \ldots, r$. We distinguish whether $x \in L_{\langle s,t\rangle}$ or $x \notin L_{\langle s,t\rangle}$.
  - Consider first $x \in L_{\langle s,t\rangle}$. Let $\{x^1, \ldots, x^l\}$ be the set of linearized versions of x in $\langle \hat{s}, \hat{t}\rangle$. Take $\varepsilon = \{x^j \mapsto y_k\}$ for the $x^j \in \{x^1, \ldots, x^l\}$ which corresponds to position i (recall that $t_i = x$). By Lemma 12 $s \succ_{acspo} t$ by case 5 implies $\hat{s} \succ_{acspo} \hat{t}$ by case 5 that, by Lemma 5, implies $\hat{s}\varepsilon \succ_{acspo} \hat{t}\varepsilon$ by case 5.
    Now take $\gamma_0 = \{x^p \mapsto x \mid p \in \{1, ..., j-1, j+1, ..., l\}$. By Lemma 5 $\hat{s}\varepsilon \succ_{acspo} \hat{t}\varepsilon$ by case 5 implies $\hat{s}\varepsilon\gamma_0 \succ_{acspo} \hat{t}\varepsilon\gamma_0$ by case 5. Therefore, by induction hypothesis we conclude $\overline{\hat{s}\varepsilon\gamma_0\sigma} \succ_{acspo} \overline{\hat{t}\varepsilon\gamma_0\sigma}$. Since $\overline{\hat{s}\varepsilon\gamma_0\sigma} \in EmbSmall(\widehat{\overline{s\sigma}})$ and $\overline{\hat{t}\varepsilon\gamma_0\sigma} = u$ then we conclude $\widehat{\overline{s\sigma}} \succ_{acspo} u$ by case 2.
  - Otherwise $x \notin L_{\langle s,t\rangle}$ and hence, $BigHead(s) \succcurlyeq_{acspo} BigHead(t) \cup \{x, ..., x\}$ (since $t_i = x$). Thus, by induction hypothesis, we have $\overline{BigHead(s)\sigma} \succcurlyeq_{acspo} \overline{BigHead(t)\sigma} \cup \{q(y_1, ..., y_r)...q(y_1, ..., y_r)\}$.
    Since $f =_{\mathcal{F}} g \succ_{\mathcal{F}} q$ implies $\overline{BigHead(s)\sigma} = BigHead(\overline{s\sigma})$ and $\overline{BigHead(t)\sigma} = BigHead(\overline{t\sigma})$ then we have that $BigHead(\overline{s\sigma}) \succ \succ_{acspo} BigHead(\overline{t\sigma}) \cup \{q(y_1, \ldots, y_r) \ldots q(y_1, ..., y_r)\}$ (take as many terms $q(y_1, ..., y_r)$ as x arguments in t).
    We also know that $NoSmallHead(\overline{s\sigma}) \succcurlyeq_{acspo} NoSmallHead(\overline{t\sigma})$ and that $\overline{s\sigma} \succeq_Q \overline{t\sigma}$. Moreover we have proved that $\widehat{\overline{s\sigma}} \succ_{acspo} u \in EmbSmall(\widehat{\overline{t\sigma}})$ for all u embedded through a term in $\widehat{\overline{t\sigma}}$ different to $q(y_1, ..., y_r)$. Therefore, by Lemma 19 we conclude $\widehat{\overline{s\sigma}} \succ_{acspo} u \in EmbSmall(\widehat{\overline{t\sigma}})$ for all u embedded through any argument $q(y_1, ..., y_r)$ in $\widehat{\overline{t\sigma}}$.

Therefore we have proved $\widehat{\overline{s\sigma}} \succ_{acspo} u$ for all $u \in EmbSmall(\widehat{\overline{t\sigma}})$.

Finally, there are two cases:

- If the number of x as arguments in s is greater or equal than the number of x as arguments in t then $BigHead(\overline{s\sigma}) = \overline{BigHead(s)\sigma} \cup Q$ and $BigHead(\overline{t\sigma}) = \overline{BigHead(t)\sigma} \cup Q'$ with $Q' \subseteq Q$. Note that if $g \succeq_{\mathcal{F}} q$ then $Q = Q' = \emptyset$, and if $q \succ_{\mathcal{F}} g$ then either $Q' \subset Q$ or s and t have the same number of x as arguments. Thus, if $Q' \subset Q$, since by induction hypothesis $BigHead(s) \succcurlyeq_{acspo} BigHead(t)$ implies $\overline{BigHead(s)\sigma} \succcurlyeq_{acspo} \overline{BigHead(t)\sigma}$ then we have $BigHead(\overline{s\sigma}) = \overline{BigHead(s)\sigma} \cup Q \succcurlyeq_{acspo} \overline{BigHead(t)\sigma} \cup Q' = BigHead(\overline{t\sigma})$ and hence we conclude $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$ by case 5a. Otherwise $Q = Q'$. If $BigHead(s) \succcurlyeq_{acspo} BigHead(t)$ then by induction hypothesis we have $\overline{BigHead(s)\sigma} \succcurlyeq_{acspo} \overline{BigHead(t)\sigma}$ and hence we conclude $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$ by case 5a. Otherwise, if $\#(s) > \#(t)$ then since $\#(\overline{s\sigma}) > \#(\overline{t\sigma})$ we conclude $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$ by case 5b (note that if the number of x as arguments in s is greater than the number of x as arguments in t then we are in this case).
  Otherwise $\#(s) \geq \#(t)$ and hence $\#(\overline{s\sigma}) \geq \#(\overline{t\sigma})$ and $\{s_1, ..., s_n\} \succ \succ_{acspo} \{t_1, ..., t_m\}$ and s and t have the same number of x as arguments). Then by induction hypothesis we have $\{\overline{s_1\sigma}, ..., \overline{s_n\sigma}\} \succcurlyeq_{acspo} \{\overline{t_1\sigma}, ..., \overline{t_m\sigma}\}$, and since by definition of the multiset extension, $M \cup \{q(y_1...y_r)\} \succcurlyeq_{acspo} N \cup \{q(y_1...y_r)\}$ implies $M \cup \{y_1...y_r\} \succcurlyeq_{acspo} N \cup \{y_1...y_r\}$ then we conclude $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$ by case 5c.
- If the number of x as arguments in s is smaller than the number of x as arguments in t then $NoSmallHead(s) \succcurlyeq_{acspo} NoSmallHead(t)$ implies $BigHead(s) \succcurlyeq_{acspo} BigHead(t) \cup \{x...x\}$ (take as many as x arguments of t). By induction hypothesis we have $\overline{BigHead(s)\sigma} \succ \succ_{acspo} \overline{BigHead(t)\sigma} \cup \{x\sigma...x\sigma\}$. Since $BigHead(\overline{s\sigma}) \supseteq \overline{BigHead(s)\sigma} \succ \succ_{acspo} \overline{BigHead(t)\sigma} \cup \{x\sigma, ..., x\sigma\} \supseteq BigHead(\overline{t\sigma})$ then we conclude $BigHead(\overline{s\sigma}) \succcurlyeq_{acspo} BigHead(\overline{t\sigma})$ and hence $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$ by case 5a.

□

Finally, by an easy induction using Lemma 18, we can prove the stability under substitutions of the ordering.

Lemma 21. *$\succ_{acspo}$ is stable under substitutions, i.e. $s \succ_{acspo} t$ implies $\overline{s\sigma} \succ_{acspo} \overline{t\sigma}$ for every substitution σ.*

Proof. By induction on $|\sigma|$ defined as the multiset $\{|w| \mid (x \mapsto w) \in \sigma\}$. Let $\sigma = \sigma_1 \cup \sigma_2$ such that for all $(x \mapsto w) \in \sigma$, if $w \in \mathcal{X}$ then $(x \mapsto w) \in \sigma_1$ otherwise $(x \mapsto w) \in \sigma_2$. By Lemma 18 $s \succ_{acspo} t$ implies $s' = s\sigma_1 \succ_{acspo} t\sigma_1 = t'$. Now we will prove that $s' \succ_{acspo} t'$ implies $\overline{s\sigma} = \overline{s'\sigma_2} \succ_{acspo} \overline{t'\sigma_2} = \overline{t\sigma}$.

If σ_2 is empty it holds trivially. Otherwise $\sigma_2 = \{x \mapsto q(w_1, \ldots, w_k)\} \cup \sigma_2'$. Then, by Lemma 20, we have $\overline{s'\{x \mapsto q(y_1, \ldots, y_k)\}} \succ_{acspo} \overline{t'\{x \mapsto q(y_1, \ldots, y_k)\}}$ with $y_1, \ldots, y_k$ new variables, and taking $\gamma = \{y_1 \mapsto w_1, \ldots, y_k \mapsto w_k\} \cup \sigma_2'$, we have $|\sigma| \gg |\gamma|$, and hence, by induction hypothesis, $\overline{s\sigma} = \overline{s'\sigma_2} = \overline{(s'\{x \mapsto q(y_1, \ldots, y_k)\})\gamma} \succ_{acspo} \overline{(t'\{x \mapsto q(y_1, \ldots, y_k)\})\gamma} = \overline{t'\sigma_2} = \overline{t\sigma}$. □

Now, we can prove Theorem 2 given in Section 3.

Theorem 2. *$\succ_{acspo}$ is a well-founded AC-compatible ordering which is stable under substitutions.*

Proof. It follows from Lemmas 4, 13, 14 and 21. □

In the following lemma we prove the monotonicity of $\succ_{acmspo}$.

Lemma 22. *$\succ_{acmspo}$ is monotonic. That is, if $s \succ_{acmspo} t$ then $cs = f(...tf_f(s)...) \succ_{acmspo} f(...tf_f(t)...) = ct$ for all terms s and t and for every flattened context $f(...[]...)$.*

Proof. $s \succ_{acmspo} t$ by definition implies $s \succeq_I t$ and $s \succ_{acspo} t$. By monotonicity of $\succeq_I$, $s \succeq_I t$ implies $cs \succeq_I ct$, and by monotonicity of $\succeq_I$ on $\succeq_Q$, $s \succeq_I t$ implies $cs \succeq_Q ct$. Thus, we have $cs \succeq_Q ct$ and hence, by Lemma 17, $f(...tf_f(s)...) \succ_{acspo} f(...tf_f(t)...)$. Therefore, $f(...tf_f(s)...) \succ_{acmspo} f(...tf_f(t)...)$.

Finally, Theorem 3 given in Section 4, follows easily.

Theorem 3. *$\succ_{acmspo}$ is an AC-compatible reduction ordering.*

Proof. It is proved using that $\succeq_I$ is an AC-compatible monotonic and stable under substitution quasi-ordering, Theorem 2 and Lemma 22.

References

[AG00] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000.

[BCL87] A. Ben-Cherifa and P. Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *Science of Computer Programming*, 9:137–160, 1987.

[BFR00] C. Borralleras, M. Ferreira, and A. Rubio. Complete monotonic semantic path orderings. Proc. of *17th Int. Conf. on Automated Deduction*, LNAI 1831:346–364, 2000.

[BP85] L. Bachmair and D.A. Plaisted. Termination orderings for associative-commutative rewriting systems. *Journal of Symbolic Computation*, 1:329–349, 1985.

[BR01] C. Borralleras and A. Rubio. A Monotonic Higher-Order Semantic Path Ordering. Proc. of *8th Int. Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR'01)* LNAI 2250:531–547, 2001.

[BR02] C. Borralleras and A. Rubio. TERMPTATION: TERMination Proof Techniques automATION. Available at `//www.lsi.upc.es/~albert/`.

[CMMU00] E. Cotejean, C. Marché, B. Monate and X. Urban. CiME system. 2000. Available at `//cime.lri.fr/`.

[CMR97] E. Cotejean, C. Marché and L. Rabehasaina. Rewrite systems for natural, integral, and rational arithmetic. Proc. of *8th Int. Conf. on Rewriting Techniques and Applications, RTA '97*, LNCS 1232:98–112, 1997.

[Der82] N. Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.

[DP93] C. Delor and L. Puel. Extension of the associative path ordering to a chain of associative commutative symbols. Proc. of *5th Int. Conf. on Rewriting Techniques and Applications RTA'93*, LNCS 690:389–404, 1993.

[GK01] J. Giesl and D. Kapur Dependency pairs for Equational Rewriting. Proc. of *12th Int. Conf. on Rewriting Techniques and Applications, RTA'01*, LNCS 2051:93–108, 2001.

[KL80] S. Kamin and J. J. Levy. Two generalizations of the recursive path ordering. Unpublished note, Dept. of Computer Science, Univ. of Illinois, Urbana, IL, 1980.

[Knu97] D. E. Knuth. The Art of Computer Programming. Vol. 2: Seminumerical Algorithms. Third Edition. Addison-Wesley, 1997.

[KS00] D. Kapur and G. Sivakumar. Proving Associative-Commutative Termination Using RPO-compatible Orderings. In *Proc. Automated Deduction in Classical and Non-Classical Logics*, LNAI 1761:40–62, 2000.

[KSZ95] D. Kapur, G. Sivakumar and H. Zhang. A Path Ordering for Proving Termination of AC Rewrite Systems. In *Journal of Automated Reasoning*, 14(2):293-316, 1995.

[KT01] K. Kusakari and Y. Toyama. On proving AC-termination by AC-dependency pairs. IEICE Transactions on Information and Systems, E84-D(5):604–612, 2001.

[MU98] C. Marché and X. Urbain. Termination of associative-commutative rewriting by dependency pairs. Proc. of *9th Int. Conf. on Rewriting Techniques and Applications, RTA'98*, LNCS 1379:241–255, 1998.

[RN95] A. Rubio and R. Nieuwenhuis. A total AC-compatible ordering based on RPO *Theoretical Computer Science*, 142(2):209–227, 1995.

[Rub02] A. Rubio. A fully syntactic *AC*-RPO. *Information and Computation*, 178(2):515-533, 2002.

[Urb01] X. Urbain. Approche incrémentale des preuves automatiques de terminaison. PhD thesis, Université Paris-Sud, UFR Scientifique d'Orsay, 2001.