Anti-unification on Absorption and Commutative Theories

David M. Cerna
Czech Academy of Sciences Institute of Computer Science
Prague, Czech Republic.
dcerna@cs.cas.cz

Andrés Felipe González Barragán
Universidade de Brasília
Brasília D.F., Brazil
andres.felipe@aluno.unb.br

Mauricio Ayala-Rincón
Universidade de Brasília
Brasília D.F., Brazil
ayala@unb.br

Temur Kutsia
Research Institute for Symbolic Computation
Johannes Kepler University Linz, Linz, Austria
kutsia@risc.jku.at

Anti-unification is a problem coined almost half a century ago by Plotkin and Reynolds to address generalities among terms. Techniques to solve this problem are of practical interest for computing commonalities and regularities in expressions, providing mechanisms to check data cloning and integrity, and code regularities offering opportunities for parallelization and prototyping of algorithms. This paper presents an algorithm for the anti-unification problem modulo commutative and absorption theories. Absorption constants and function symbols behave as zero and product in arithmetic theories or as "true" and disjunction in Boolean algebra. The technique is built over an algorithm developed by the authors, given as a set of inference rules, that allows only absorption symbols. This paper adapts absorption anti-unification rules and adds rules that treat commutative and absorption-commutative symbols. The algorithm is proven sound, and its completeness is discussed.

1 Introduction

Anti-unification is the problem of expressing commonalities between terms in the most particular manner. For instance, $f(g(g(a)),f(a,b))$ and $f(g(g(a)),f(a,g(a)))$ can be generalized with terms as x, $f(x,y)$, $f(g(x),f(z,y))$, but the *least general generalization* is $f(g(g(a)),f(a,y))$ since it maximally captures the commonalities of both terms, including subterm coincidences. Namely, a generalization of terms s and t, is a term r such that there exist substitutions σ and ρ with $r\sigma = s$, and $r\rho = t$.

Interest in anti-unification is increasing because of new theoretical developments and industrial applications in different areas. Among others, exciting recent applications include the detection of code regularities for efficient parallel compilation [7], efficient searching of common patterns to detect equivalent code in corpora of programs [8], and production of recommendations of code changes and adaptations to prevent bugs and misconfigurations in large computational services [12].

The generalizations presented above are related to the syntactic case of the anti-unification problem. But operators may have algebraic properties. For instance, if the symbol f is commutative, the terms $f(f(a,c),f(b,d)))$ and $f(f(a,d),f(b,c))$ have two incomparable least general generalizations $r_1 = f(f(a,x),f(b,y)))$ and $r_2 = f(f(d,x),f(c,y)))$. These generalizations are incomparable since each of them is not an instance of the other: there exists neither σ with $r_1\sigma = r_2$ nor ρ with $r_2\rho = r_1$. Anti-unification modulo equational theories (including operators with algebraic equations properties as commutativity) are highly applicable and give rise to more complex problems than syntactic anti-unification. Alpuente *et al.* investigated anti-unification modulo associativity (A), commutativity (C), and unital (U), including the cases of theories with operators that hold combinations of these properties [1, 2, 3]. Interesting aspects arise, as the fact that theories combining two unital symbols are *nullary* [10], which means

Submitted to:
LSFA 2024

a set of minimal least general generalizations does not need to exist for some problems in this theory. (The type of anti-unification problems, analogous to the type classification of unification problems as defined in, e.g., [13], can be nullary, unary, finitary, or infinitary.) Also, and more practical, it was proved that anti-unification in semirings is also nullary [9]. See [11] for a recent survey about anti-unification and its applications.

In recent work [6], the authors presented a sound and complete algorithm that solves anti-unification modulo absorption ($\mathfrak{a}$-)theories, theories with operators that satisfy the axioms $\{f(\varepsilon_f,x) \approx \varepsilon_f, f(x,\varepsilon_f) \approx \varepsilon_f\}$ (as happens with zero and multiplication, true and disjunction, etc.). In addition, the type of anti-unification modulo absorption was proved infinitary.

Several algebras which own operators with absorption properties like semi-groups and monoids may include commutative properties. Interesting examples of these algebras are the integers with multiplication with zero as absorption constant; the integers with the greatest common divisor *gcd* with one as the absorption constant; the powerset of a given set with the intersection $\cap$ with $\varnothing$ as absorption constant; Boolean algebras with two binary operations, where each operation is commutative and has a zero element. This justifies the interest in anti-unification over theories with C- and $\mathfrak{a}$-symbols.

Contribution. This paper discusses current progress in extending the anti-unification algorithm modulo absorption theories given in [6], allowing the inclusion of commutative symbols. The proposed algorithm deals with all possible combinations, that is, it allows theories with C-symbols, $\mathfrak{a}$-symbols, and $\mathfrak{a}$C-symbols (i.e., symbols that satisfy both absorption and commutativity properties).

Organization. Section 2 presents the required background on anti-unification. Section 3 introduces the set of inference rules giving the procedure to solve anti-unification problems modulo $\mathfrak{a}$C. The procedure works over configurations that include unsolved problems, solved problems, a substitution, and a delayed set. This section proves the termination and soundness of the algorithm. Section 4 discusses how the delayed set computed in final configurations is used to build abstractions that are used with the final substitution and the solved problem to construct the least general generalizations. This section also drafts the analysis of the completeness of the algorithm. Finally, Section 5 shortly concludes and presents possible future paths of research.

2 Preliminaries

Let $\mathcal{V}$ be a countable set of variables and $\mathcal{F}$ a set of function symbols, each associated with a fixed arity. Additionally, we assume $\mathcal{F}$ contains a special constant $\star$, referred to as the *wild card*. The set of terms derived from the sets mentioned above is denoted by $\mathcal{T}(\mathcal{F},\mathcal{V})$, whose members are constructed using the grammar $t ::= x \mid f(t_1,\ldots,t_n)$, where $x \in \mathcal{V}$ and $f \in \mathcal{F}$ with arity $n \geq 0$. When $n = 0$, f is called a *constant*. Constant and function symbols, terms, and variables are denoted by lower-case letters of the first, second, third, and fourth quarter of the alphabet, respectively (e.g., $a,b,\ldots$; $f,g,\ldots$; $r,s,\ldots$; $w,x,\ldots$). The set of variables occurring in a term t is denoted by $var(t)$. The *size* of a term is defined inductively as: $size(x) = 1$, and $size(f(t_1,\ldots,t_n)) = 1 + \sum_{i=1}^{n} size(t_i)$. The *head* of a term t is defined as $head(x) = x$ and $head(f(t_1,\ldots,t_n)) = f$, for $n \geq 0$.

The set of *positions* of a term t, denoted as $pos(t)$, is the set of sequences of positive integers, defined as $pos(x) = \{\lambda\}$, $pos(f(t_1,\ldots,t_n)) = \{\lambda\} \cup \bigcup_{i=1}^{n} \{i.p \mid p \in pos(t_i)\}$, where λ denotes the empty string. The prefix order $\sqsubseteq$ over positions is defined as usual. The subterm of s at position $p \in pos(s)$, $s|_p$, is defined recursively as $s|_\lambda = s$, and $f(s_1,\ldots,s_n)|_{i\cdot q} = s_i|_q$, for $1 \leq i \leq n$.

A *substitution* is a function $\sigma : \mathcal{V} \to \mathcal{T}(\mathcal{F},\mathcal{V})$ such that $\sigma(x) \neq x$ for only finitely many variables.

Substitutions are denoted by lower-case Greek letters. The *domain* of σ, denoted as $dom(\sigma)$, is the set of variables such that that $\sigma(x) \neq x$. Substitutions are extended to terms as usual. The *range* of σ, denoted $ran(\sigma)$, is the set of terms $\{\sigma(x) \mid x \in dom(\sigma)\}$. The set of variables in $ran(\sigma)$ is denoted as $rvar(\sigma)$. We refer to a *ground* term t if $var(t) = \emptyset$ and a ground substitution σ if for all $t \in ran(\sigma)$, t is ground. Postfix notation denotes the application of a substitution σ to the term t: $t\sigma$. The identity substitution, denoted by id, is such that $dom(id) = \emptyset$.

A substitution σ can be described as a finite sets of *bindings* as $\{x \mapsto x\sigma \mid x \in dom(\sigma)\}$. The *composition* of substitutions ρ and σ, $(\rho \circ \sigma)$, is written $\sigma\rho$. Substitution application satisfies $x(\sigma\rho) = (x\sigma)\rho$ for each $x \in \mathcal{V}$. The *restriction of a substitution* σ to a set of variables V, denoted by $\sigma|_V$, is a substitution defined as $\sigma|_V(x) = \sigma(x)$ for all $x \in V$ and $\sigma|_V(x) = x$ otherwise.

Definition 2.1 (Equational theory [13]). An equational theory T_E is a class of algebraic structures with a set of equational axioms E over $\mathcal{T}(\mathcal{F},\mathcal{V})$.

The relation $\{(s,t) \in \mathcal{T}(\mathcal{F},\mathcal{V}) \times \mathcal{T}(\mathcal{F},\mathcal{V}) \mid E \vDash (s,t)\}$ induced by a set of equalities E gives the set of equalities satisfied by all structures in the theory of E. We will use the notation $s \approx_E t$ for (s,t) belonging to this set. Also, we will identify T_E with the set of axioms E. Groups, monoids, and semirings are examples of equational theories.

The focus of this work is anti-unification modulo equational theories that may include commutative symbols, for short C-symbols, with axioms for commutativity, $\{f(x,y) \approx f(y,x)\}$, absorption symbols, for short $\mathfrak{a}$-Symbols, with absorption axioms, $\{f(x,\varepsilon_f) \approx \varepsilon_f,\, f(\varepsilon_f,x) \approx \varepsilon_f\}$, as well as symbols that are both commutative and absorption, for short $\mathfrak{a}$C-symbols, with axioms $\{f(\varepsilon_f,x) \approx \varepsilon_f, f(x,y) \approx f(y,x)\}$. Symbols f and ε_f are called *related* $\mathfrak{a}$*-symbols*. Theories with only $\mathfrak{a}$-symbols or only C-symbols or only $\mathfrak{a}$C-symbols are called $\mathfrak{a}$-theories or C-theories or $\mathfrak{a}$C-theories, respectively. Theories that combine these classes of symbols are distinguished by referring to the specific properties; for instance, an ($\mathfrak{a}$)(C)($\mathfrak{a}$C)-theory contains different symbols holding the three sets of axioms. When no confusion arises, we will refer to such theories as $\mathfrak{a}$C-theories. Also, when no confusion arises, we will say that an $\mathfrak{a}$C-symbol is a C-symbol or an $\mathfrak{a}$-symbol.

Including $\mathfrak{a}$-, C-, and $\mathfrak{a}$C-symbols requires some adaptations in the notions of positions and subterms. For terms s and t, the set of *C-relative equal positions of t and s* or, for short, *C-positions*, denoted as $pos_C(s,t)$ is the set that contains exactly all pairs of positions (p,q), such that $p \in pos(s)$ and $q \in pos(t)$ and $|p| = |q|$, and for all $p' \sqsubset p$ and $q' \sqsubset q$ with $|p'| = |q'|$, $head(s|_{p'}) = head(t|_{q'})$, and if $head(s|_{p'})$ is neither a C-symbol nor an $\mathfrak{a}$C-symbol (for short, C-symbol), then $p'.i \sqsubseteq p$ and $q'.i \sqsubseteq q$, for (the same) $i = 1,2$, but if $head(s|_{p'})$ is a C-symbol, then $p'.i \sqsubseteq p$ and $q'.j \sqsubseteq q$, for (maybe distinct) $1 \leq i,j \leq 2$.

Example 2.1. [C-relative equal positions] Let f and h be an $\mathfrak{a}$C-symbol and a syntactic unary symbol, respectively. Consider the terms $s = h(f(f(a,x),b))$ and $t = h(f(f(y,z),f(a,a)))$. Then

$$pos_C(s,t) = \left\{ \begin{array}{l} (\lambda,\lambda),(1,1),(1.1,1.1),(1.1,1.2),(1.2,1.1),(1.2,1.2),(1.1.1,1.1.1),(1.1.1,1.1.2),\\ (1.1.1,1.2.1),(1.1.1,1.2.2),(1.1.2,1.1.1),(1.1.2,1.1.2),(1.1.2,1.2.1),(1.1.2,1.2.2) \end{array} \right\}$$

A *collapsable* position and subterm of a term s, is any position $p \in pos(s)$ and the associated subterm $s|_p$, such that $head(s|_p) = f$, an $\mathfrak{a}$-symbol, and there exists p' such that $p \sqsubset p'$, and $s|_{p'}$ is a variable, say x, and for all q with $p \sqsubset q \sqsubset p'$, $head(s|_q) = f$. Notice that instantiating x with ε_f, the subterm at position p collapses: $s|_p\{x \mapsto \varepsilon_f\} \approx_{\mathfrak{a}} \varepsilon_f$.

Example 2.2. [Collapsable Positions and Subterm] Following Example 2.1, the collapsable positions of s and t are the positions $1, 1.1 \in pos(s)$, and $1, 1.1 \in pos(t)$, respectively. Additionally, the associated subterms are those in the collapsable positions; for example, $t|_{1.1} = f(y,z)$ is one of them, since $t|_{1.1}\{y \mapsto \varepsilon_f\} \approx_{\mathfrak{a}} \varepsilon_f$. Observe that 1.2 is not a collapsable position of t.

For any $(p,q) \in pos_C(s,t)$, such that $\{\varepsilon_f, f\} = \{head(s|_p), head(t|_q)\}$, (p,q) is said to be a pair of *absorption positions* regarding the symbol f.

The set of absorption positions regarding symbol f is denoted as $\mathfrak{a}p_f(s,t)$.

Example 2.3 (Absorption Positions)**.** Let $s = g(g(\varepsilon_f, b), h(f(b,c)))$ and $t = g(h(\varepsilon_f), g(a, f(b,a)))$ be terms, where g and f are a C-symbol and an $\mathfrak{a}$C-symbol, respectively, being f and ε_f related $\mathfrak{a}$-symbols. Then, $\mathfrak{a}p_f(s,t) = \{(1.1, 2.2), (2.1, 1.1)\}$.

Definition 2.2 (E-generalization, $\preceq_E$)**.** The generalization relation of the theory induced by E holds for terms $r, s \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, written $r \preceq_E s$, if there exists a substitution σ such that $r\sigma \approx_E s$. In this case, we say that r is *more general* than s modulo E. If $r \preceq_E s$ and $r \preceq_E t$, we say that r is an E-generalization of s and t. The set of all E-generalizations of s and t is denoted as $\mathcal{G}_E(s,t)$.

Example 2.4 ($\mathfrak{a}$C-generalization, $\preceq_{\mathfrak{a}C}$)**.** Consider f an $\mathfrak{a}$C-symbol, then the term $r = f(b, f(x,a))$ is an $\mathfrak{a}$C-generalization of the terms $f(f(a,a),b)$ and ε_f since $r \preceq_{\mathfrak{a}C} f(f(a,a),b)$, and $r \preceq_{\mathfrak{a}C} \varepsilon_f$.

Indeed, $r\{x \mapsto a\} \approx_{\mathfrak{a}C} f(f(a,a),b)$, and $r\{x \mapsto \varepsilon_f\} = f(b, f(\varepsilon_f, a)) \approx_{\mathfrak{a}C} \varepsilon_f$.

Definition 2.3 (Minimal complete set of E-generalizations ($mcsg_E$))**.** The *minimal complete set of E-generalizations* of the terms s and t, denoted as $mcsg_E(s,t)$, is a subset of $\mathcal{G}_E(s,t)$ satisfying:

1. For each $r \in \mathcal{G}_E(s,t)$ there exists $r' \in mcsg_E(s,t)$ such that $r \preceq_E r'$. (Completeness)
2. If $r, r' \in mcsg_E(s,t)$ and $r \preceq_E r'$, then $r = r'$. (Minimality)

Example 2.5. Continuing Example 2.4. The minimal complete set of $\mathfrak{a}$C-generalizations of the terms $f(f(a,a),b)$ and ε_f is given as

$$mcsg_{\mathfrak{a}C}(f(f(a,a),b), \varepsilon_f) = \{f(f(x,a),b), f(f(x,x),b), f(f(a,a),x)\}$$

Definition 2.4 (Anti-unification type)**.** The anti-unification type of an equational theory induced by E may have one of the following *types*:

- *Unitary*: $mcsg_E(s,t)$ exists for all $s,t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and is always singleton.
- *Finitary*: $mcsg_E(s,t)$ exists and is finite for all $s,t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, and there exist $s', t' \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ for which $1 < |mcsg_E(s',t')| < \infty$.
- *Infinitary*: $mcsg_E(s,t)$ exists for all $s,t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, and $mcsg_E(s',t')$ is infinite for some terms $s', t' \in \mathcal{T}(\mathcal{F}, \mathcal{V})$.
- *Nullary*: for some $s,t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, $mcsg_E(s,t)$ does not exist.

3 A Sound Algorithm for ($\mathfrak{a}$)(C)($\mathfrak{a}$C)-Anti-Unification

In this section, the algorithm AUNIF is described through the inferences rules in Table 1, which computes generalizations for terms in $\mathfrak{a}$-, C-, and $\mathfrak{a}$C-theories and their combinations. AUNIF transforms quadruples called *configurations* consisting of an unsolved (*active*), and a solved (*store*) set of labelled pair of terms, and a substitution that expresses the *anti-unifier* under construction. The other component is an abstraction set, here called *delayed set*, that as in the case of $\mathfrak{a}$-theories needs to be considered (cf. [6]).

An *anti-unification triple* (AUT) is a triple of the form $s \triangleq_x t$, where $x \in \mathcal{V}$, called the *label of the AUT*, and $s,t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. Given a set A of AUTs, $labels(A) = \{x \mid s \triangleq_x t \in A\}$. A set of AUTs is *valid* if its labels are pairwise disjoint. We extend the notion of *size* to AUTs and sets of AUTs as the sum of the sizes of the terms in the AUTs. The *wild card* plays an integral role in our algorithm for computing generalizations when some absorption constant is expanded. In particular, an AUT is referred to as *wild* if the wild card is either the left or right side. The algorithm aims to compute a set of terms generalizing the input AUT and a set of *solved* AUTs from which we can compute how such terms generalize the input AUT.

Definition 3.1 (Solved AUT)**.** An AUT $s \triangleq_x t$ is *solved* over E if $head(s) \neq head(t)$, and for $\mathfrak{a}$- and $\mathfrak{a}$C-theories, $head(s)$ and $head(t)$ are not related $\mathfrak{a}$-symbols, and $s \triangleq_x t$ is not wild.

In the next sections, the inference rules are applied over *configurations* defined below.

Definition 3.2 (Configuration)**.** A configuration is a quadruple of the form $\langle A; S; D; \theta \rangle$, where:

- A is a valid set of AUTs; (*Active set*)
- S is a valid set of solved AUTs; (*Store*)
- D is a valid set of *wild* AUTs; (*Delayed set*)
- θ is a *substitution* such that $rvar(\theta) = labels(A) \cup labels(S) \cup labels(D)$. (*Anti-unifier*)

In addition, the following conditions hold:

- $labels(A), labels(S), labels(D)$, and $dom(\theta)$ are pairwise disjoint, and
- all terms occurring in a configuration are in their $\mathfrak{a}$-normal forms: an absorption constant does not occur as the argument to its $\mathfrak{a}$-symbol.

The inferences rules in Table 1 can be used to compute generalizations for E-theories where E may contain $\mathfrak{a}$-, C-, and $\mathfrak{a}$C-symbols. These rules will be referred to as follows: Decompose ($\stackrel{Dec}{\Longrightarrow}$), Solve ($\stackrel{Sol}{\Longrightarrow}$), Commutative ($\stackrel{Com}{\Longrightarrow}$), Expansions for Left Absorption, ($\stackrel{ExpLA1}{\Longrightarrow}$and $\stackrel{ExpLA2}{\Longrightarrow}$), Expansions for Right Absorption ($\stackrel{ExpRA1}{\Longrightarrow}$and $\stackrel{ExpRA2}{\Longrightarrow}$), Expansion Absorption in Both sides ($\stackrel{ExpBA1}{\Longrightarrow}$) and ($\stackrel{ExpBA2}{\Longrightarrow}$), and Merge ($\stackrel{Mer}{\Longrightarrow}$).

The algorithm AUNIF exhaustively applies all possible inference rules to each configuration.

By $\mathcal{C} \Longrightarrow \mathcal{C}'$ we denote the application of some inference rule of Table 1 to $\mathcal{C}$ resulting in $\mathcal{C}'$. By $\mathcal{C} \Longrightarrow^* \mathcal{C}'$ we denote a finite sequence of inference rule applications starting at $\mathcal{C}$ and ending with $\mathcal{C}'$. In both cases we say $\mathcal{C}'$ is *derived* from $\mathcal{C}$. An initial configuration is a configuration of the form $\langle A; \emptyset; \emptyset; \iota \rangle$, where $\iota = \{\mathtt{f}_A(x) \mapsto x \mid x \in labels(A)\}$ with $\mathtt{f}_A : \mathcal{V} \to (\mathcal{V} \setminus labels(A))$ being a bijection over variables. A configuration $\mathcal{C}$ is referred to as *final* if no inference rule applies to $\mathcal{C}$.

The set of *final configurations* derived from an initial configuration $\mathcal{C}$ is denoted by AUNIF($\mathcal{C}$).

The following lemma and theorem state that the algorithm AUNIF preserves configurations and is terminating.

Lemma 3.1 (Configuration Preservation)**.** If $\mathcal{C}$ is a configuration and $\mathcal{C} \Longrightarrow \mathcal{C}'$, then $\mathcal{C}'$ is a configuration.

Proof. According to the rules in Table 1, we can have the following two cases:

- A rule removes an AUT $s \triangleq_x t$ from the active set of $\mathcal{C}$. Then either $s \triangleq_x t$ occurs in the store of $\mathcal{C}'$, or the anti-unifier component of $\mathcal{C}'$ is the composition of the anti-unifier component of $\mathcal{C}$ with $\{x \mapsto r\}$, where $var(r)$ are fresh variables labelling newly added AUTs in the active and delayed sets of $\mathcal{C}'$.
- A rule removes an AUT $s \triangleq_x t$ from the store of $\mathcal{C}$. Then the store of $\mathcal{C}'$ is a subset of the store of $\mathcal{C}$ and the anti-unifier component of $\mathcal{C}'$ is the composition of the anti-unifier component of $\mathcal{C}$ with $\{x \mapsto y\}$, where y is a label of an AUT in the store of $\mathcal{C}$ such that $x \neq y$.

$$(\stackrel{Dec}{\Longrightarrow})\quad \frac{\langle\{f(s_1,\dots,s_n)\triangleq_x f(t_1,\dots,t_n)\}\uplus A;S;D;\theta\rangle}{\langle\{s_1\triangleq_{y_1} t_1,\dots,s_n\triangleq_{y_n} t_n\}\cup A;S;D;\theta\{x\mapsto f(y_1,\dots,y_n)\}\rangle}$$

where f is an n-ary symbol, $n\geq 0$, and $y_1,\dots,y_n$ are fresh variables.

$$(\stackrel{Com}{\Longrightarrow})\quad \frac{\langle\{f(s_1,s_2)\triangleq_x f(t_1,t_2)\}\uplus A;S;D;\theta\rangle}{\langle\{s_1\triangleq_{y_1} t_2,s_2\triangleq_{y_2} t_1\}\cup A;S;D;\theta\{x\mapsto f(y_1,y_2)\}\rangle}$$

for f a C- or $\mathfrak{a}$C-symbol and y_1,y_2 fresh variables.

$$(\stackrel{Sol}{\Longrightarrow})\quad \frac{\langle\{s\triangleq_x t\}\uplus A;S;D;\theta\rangle}{\langle A;\{s\triangleq_x t\}\cup S;D;\theta\rangle}$$

where $head(s)\neq head(t)$ and they are not related $\mathfrak{a}$-symbols.

$$(\stackrel{Mer}{\Longrightarrow})\quad \frac{\langle\varnothing;\{s_1\triangleq_x t_1,s_2\triangleq_y t_2\}\cup S;D;\theta\rangle}{\langle\varnothing;\{s_2\triangleq_y t_2\}\cup S;D;\theta\{x\mapsto y\}\rangle}$$

where $s_1\approx_E s_2$ and $t_1\approx_E t_2$.

In the following rules, f is an $\mathfrak{a}$-symbol and y_1,y_2 are fresh variables:

$$(\stackrel{ExpLA1}{\Longrightarrow})\quad \frac{\langle\{\varepsilon_f\triangleq_x f(t_1,t_2)\}\uplus A;S;D;\theta\rangle}{\langle\{\varepsilon_f\triangleq_{y_1} t_1\}\cup A;S;\{\star\triangleq_{y_2} t_2\}\cup D;\theta\{x\mapsto f(y_1,y_2)\}\rangle}$$

$$(\stackrel{ExpLA2}{\Longrightarrow})\quad \frac{\langle\{\varepsilon_f\triangleq_x f(t_1,t_2)\}\uplus A;S;D;\theta\rangle}{\langle\{\varepsilon_f\triangleq_{y_2} t_2\}\cup A;S;\{\star\triangleq_{y_1} t_1\}\cup D;\theta\{x\mapsto f(y_1,y_2)\}\rangle}$$

$$(\stackrel{ExpRA1}{\Longrightarrow})\quad \frac{\langle\{f(s_1,s_2)\triangleq_x \varepsilon_f\}\uplus A;S;D;\theta\rangle}{\langle\{s_1\triangleq_{y_1} \varepsilon_f\}\cup A;S;\{s_2\triangleq_{y_2} \star\}\cup D;\theta\{x\mapsto f(y_1,y_2)\}\rangle}$$

$$(\stackrel{ExpRA2}{\Longrightarrow})\quad \frac{\langle\{f(s_1,s_2)\triangleq_x \varepsilon_f\}\uplus A;S;D;\theta\rangle}{\langle\{s_2\triangleq_{y_2} \varepsilon_f\}\cup A;S;\{s_1\triangleq_{y_1} \star\}\cup D;\theta\{x\mapsto f(y_1,y_2)\}\rangle}$$

$$(\stackrel{ExpBA1}{\Longrightarrow})\quad \frac{\langle\{\varepsilon_f\triangleq_x \varepsilon_f\}\uplus A;S;D;\theta\rangle}{\langle A;S;\{\varepsilon_f\triangleq_{y_1} \star,\ \star\triangleq_{y_2}\varepsilon_f\}\cup D;\theta\{x\mapsto f(y_1,y_2)\}\rangle}$$

$$(\stackrel{ExpBA2}{\Longrightarrow})\quad \frac{\langle\{\varepsilon_f\triangleq_x \varepsilon_f\}\uplus A;S;D;\theta\rangle}{\langle A;S;\{\star\triangleq_{y_1} \varepsilon_f,\ \varepsilon_f\triangleq_{y_2}\star\}\cup D;\theta\{x\mapsto f(y_1,y_2))\}\rangle}$$

Table 1: Inference rules for the AUNIF algorithm for ($\mathfrak{a}$)(C)($\mathfrak{a}$C)-theories.

In both cases, the properties of a configuration are preserved. □

Theorem 3.1 (Termination)**.** AUNIF is terminating for any configuration $\mathcal{C}$.

Proof. Let $\mathcal{C}=\langle A;S;D;\theta\rangle$. We define $size(\mathcal{C}):=(size(A),size(S))$ and compare these pairs lexicographically. This ordering is well-founded since the size of a set of AUTs is a natural number. Observe that if $\mathcal{C}\Longrightarrow\mathcal{C}'$ then $size(\mathcal{C})>size(\mathcal{C}')$. Thus, every sequence of rule applications terminates. Furthermore, any configuration can be transformed by rules from Table 1 into finitely many ways. Thus, by König's Lemma, AUNIF($\mathcal{C}$) is finite and finitely computable. □

The next proof of soundness is valid for ($\mathfrak{a}$)(C)($\mathfrak{a}$C)-theories.

Theorem 3.2 (Soundness)**.** Let $\langle A_0;S_0;D_0;\theta_0\rangle\Longrightarrow^*\langle\varnothing;S_n;D_n;\theta_n\rangle$ be a derivation to a final configuration. Then for all $s\triangleq_x t\in A_0\cup S_0$, $x\theta_n\in\mathcal{G}_E(s,t)$, for E any combination of C, $\mathfrak{a}$, and $\mathfrak{a}$C.

Proof. We proceed by induction over the derivation length.

Basecase. If the derivation has length 0, then it starts with a final configuration implying that $A_0 = \varnothing$ and for all $s \triangleq_x t \in S_0$, since $x \notin dom(\theta_0)$, $x\theta_0 = x \in \mathcal{G}_E(s,t)$.

Stepcase. Now consider our derivation having the following form:

$$\langle A_0; S_0; D_0; \theta_0 \rangle \Longrightarrow \langle A_1; S_1; D_1; \theta_1 \rangle \Longrightarrow^n \langle \varnothing; S_{n+1}; D_{n+1}; \theta_{n+1} \rangle \tag{1}$$

We assume for the induction hypothesis (IH) that for derivations of the form

$$\langle A_1; S_1; D_1; \theta_1 \rangle \Longrightarrow^n \langle \varnothing; S_{n+1}; D_{n+1}; \theta_{n+1} \rangle,$$

the theorem holds and show that the theorem hold for derivations of the form presented in Equation 1. We continue the proof considering any options for the transition from $\langle A_0; S_0; D_0; \theta_0 \rangle$ to $\langle A_1; S_1; D_1; \theta_1 \rangle$.

1. **(Dec)**. Assume that the derivation is of the form:

$$\langle \{ f(s_1, \ldots, s_m) \triangleq_y f(t_1, \ldots, t_m) \} \cup A'; S_0; D_0; \theta_0 \rangle \overset{Dec}{\Longrightarrow}$$
$$\langle \{ s_1 \triangleq_{x_1} t_1, \ldots, s_m \triangleq_{x_m} t_m \} \cup A'; S_1; D_1; \theta_1 \rangle \Longrightarrow^n \langle \varnothing; S_{n+1}; D_{n+1}; \theta_{n+1} \rangle$$

   where $\theta_1 = \theta_0 \{ y \mapsto f(x_1, \ldots, x_m) \}$. By the Induction hypothesis, we know that for all $1 \le i \le m$, $x_i \theta_{n+1} \in \mathcal{G}_E(s_i, t_i)$ implying that

$$f(x_1, \ldots, x_m)\theta_{n+1} \in \mathcal{G}_E(f(s_1, \ldots, s_m), f(t_1, \ldots, t_m)).$$

2. **(Sol)**. Assume that the derivation is of the form:

$$\langle \{ s \triangleq_y t \} \cup A'; S_0; D_0; \theta_0 \rangle \overset{Sol}{\Longrightarrow} \langle \{ A'; S_1; D_1; \theta_1 \} \rangle \Longrightarrow^n \langle \varnothing; S_{n+1}; D_{n+1}; \theta_{n+1} \rangle,$$

   where $S_1 = \{ s \triangleq_y t \} \cup S_0$. By IH, θ_{n+1} generalizes all the AUTs with labels in S_1. Thus, $y\theta_{n+1} \in \mathcal{G}_E(s,t)$.

3. **(Com)**. Assume that the derivation is of the form:

$$\langle \{ f(s_1, s_2) \triangleq_y f(t_1, t_2) \} \cup A'; S_0; D_0; \theta_0 \rangle \overset{Com}{\Longrightarrow}$$
$$\langle \{ s_1 \triangleq_{x_1} t_2, s_2 \triangleq_{x_2} t_1 \} \cup A'; S_1; D_1; \theta_1 \rangle \Longrightarrow^n \langle \varnothing; S_{n+1}; D_{n+1}; \theta_{n+1} \rangle$$

   where $\theta_1 = \theta_0 \{ y \mapsto f(x_1, x_2) \}$. By the Induction hypothesis, we know that $x_1 \theta_{n+1} \in \mathcal{G}_E(s_1, t_2)$ and $x_2 \theta_{n+1} \in \mathcal{G}_E(s_2, t_1)$, implying that

$$f(x_1, x_2)\theta_{n+1} \in \mathcal{G}_E(f(s_1, s_2), f(t_1, t_2)).$$

4. **(Mer)** Assume that the derivation is of the form:

$$\langle \varnothing; \{ s_1 \triangleq_y t_1, s_2 \triangleq_z t_2 \} \cup S'; D_0; \theta_0 \rangle \overset{Mer}{\Longrightarrow}$$
$$\langle \varnothing; \{ s_2 \triangleq_z t_2 \} \cup S'; D_1; \theta_1 \rangle \Longrightarrow^n \langle \varnothing; S_{n+1}; D_{n+1}; \theta_{n+1} \rangle.$$

   Notice that $\theta_1 = \theta_0 \{ y \mapsto z \}$, where z is the label of the AUT $\{ s_2 \triangleq_z t_2 \} \in S_0$. By IH, $z\theta_{n+1} \in \mathcal{G}_E(s_2, t_2)$ implying that $y\theta_{n+1} = y\{ y \mapsto z \}\theta_{n+1} \in \mathcal{G}_E(s_1, t_1)$, since $s_1 \approx_E s_2$, $t_1 \approx_E t_2$.

5. **(ExpLA1)**. Assume that the derivation is of the form:

$$\langle\{\varepsilon_f \triangleq_y f(s,t)\} \cup A'; S_0; D_0; \theta_0\rangle \overset{ExpLA1}{\Longrightarrow}$$
$$\langle\{\varepsilon_f \triangleq_{x_1} s\} \cup A'; S_1; D_1; \theta_1\rangle \Longrightarrow^n \langle\varnothing; S_{n+1}; D_{n+1}; \theta_{n+1}\rangle$$

where $D_1 = \{\star \triangleq_{x_2} t\} \cup D_0$ and $\theta_1 = \theta_0\{y \mapsto f(x_1,x_2)\}$. By the IH, all the AUTs in $\{\varepsilon_f \triangleq_{x_1} s\} \cup A'$ are generalized by the substitution θ_{n+1}, thus, $x_1\theta_{n+1} \in \mathcal{G}_E(\varepsilon_f, s)$. Furthermore, since $x_2 \in labels(D)$ then $x_2\theta_{n+1} = x_2$ and $x_2 \preceq_E t$. We can build the generalization $y\theta_{n+1} = f(x_1\theta_{n+1}, x_2\theta_{n+1})$. Observe that $f(x_1\theta_{n+1}, x_2\theta_{n+1}) = f(x_1\theta_{n+1}, x_2) \in \mathcal{G}_E(f(\varepsilon_f,t), f(s,t))$ and since $f(\varepsilon_f,t) \approx_E \varepsilon_f$, we get that $y\theta_{n+1}$ belongs to $\mathcal{G}_E(\varepsilon_f, f(s,t))$.

6. The analysis of the rules **(ExpLA2)**, **(ExpRA1)** and **(ExpRA2)** is analogous to the previous one.

7. **(ExpBA1)**. Assume that the derivation is of the form:

$$\langle\{\varepsilon_f \triangleq_y \varepsilon_f\} \cup A'; S_0; D_0; \theta_0\rangle \overset{ExpBA1}{\Longrightarrow}$$
$$\langle A'; S_1; D_1; \theta_1\rangle \Longrightarrow^n \langle\varnothing; S_{n+1}; D_{n+1}; \theta_{n+1}\rangle$$

where $D_1 = \{\varepsilon_f \triangleq_{x_1} \star, \star \triangleq_{x_2} \varepsilon_f\} \cup D_0$ and $\theta_1 = \theta_0\{y \mapsto f(x_1,x_2)\}$. Notice, $x_i\theta_{n+1} = x_i$ and $x_i \preceq_E \varepsilon_f$, for $i \in \{1,2\}$. This implies that $y\theta_{n+1} = f(x_1\theta_{n+1}, x_2\theta_{n+1}) = f(x_1,x_2) \in \mathcal{G}_E(\varepsilon_f, \varepsilon_f)$. The case **(ExpBA2)** is analogous.

□

Including C-symbols in $\mathfrak{a}$-theories gives rise to new generalizations not considered before in [6] as shown in the Example below.

Example 3.1. Let $\langle\{g(g(a,b),a) \triangleq_x g(b,g(b,b))\}; \varnothing; \varnothing; \iota\rangle$ be an initial configuration, where g is a C-symbol. The rules (Dec) and (Com) can be applied, giving rise to three different derivations. The first derivation, which starts with the rule (Dec), computes the generalization $g(w_1,w_2)$. Rule (Dec) generates the active set $\{g(a,b) \triangleq_{w_1} b, a \triangleq_{w_2} g(b,b)\}$ and the anti-unifier $\{x \mapsto g(w_1,w_2)\}$. By exhaustive application of the rule (Com), two other derivations are computed, coincidentally leading to the same final configuration.

Derivation 2:

$$\langle\{g(g(a,b),a) \triangleq_x g(b,g(b,b))\}; \varnothing; \varnothing; \iota\rangle \overset{Com}{\Longrightarrow}$$
$$\langle\{g(a,b) \triangleq_w g(b,b), a \triangleq_z b\}; \varnothing; \varnothing; \{x \mapsto g(w,z)\}\rangle \overset{Dec}{\Longrightarrow}$$
$$\langle\{a \triangleq_{y_1} b, b \triangleq_{y_2} b, a \triangleq_z b\}; \varnothing; \varnothing; \{x \mapsto g(g(y_1,y_2),z), \ldots\}\rangle \overset{Dec}{\Longrightarrow}$$
$$\langle\{a \triangleq_{y_1} b, a \triangleq_z b\}; \varnothing; \varnothing; \{x \mapsto g(g(y_1,b),z), \ldots\}\rangle \overset{Sol\ \times 2}{\Longrightarrow}$$
$$\langle\varnothing; \{a \triangleq_{y_1} b, a \triangleq_z b\}; \varnothing; \{x \mapsto g(g(y_1,b),z), \ldots\}\rangle \overset{Mer}{\Longrightarrow}$$
$$\langle\varnothing; \{a \triangleq_z b\}; \varnothing; \{x \mapsto g(g(z,b),z), \ldots\}\rangle$$

Derivation 3:

$$\langle\{g(g(a,b),a) \triangleq_x g(b,g(b,b))\}; \varnothing; \varnothing; \iota\rangle \overset{Com}{\Longrightarrow}$$
$$\langle\{g(a,b) \triangleq_w g(b,b), a \triangleq_z b\}; \varnothing; \varnothing; \{x \mapsto g(w,z)\}\rangle \overset{Com}{\Longrightarrow}$$

$$\langle \{a \triangleq_{y_3} b, b \triangleq_{y_4} b, a \triangleq_z b\}; \varnothing; \varnothing; \{x \mapsto g(g(y_3,y_4),z),\dots\}\rangle \overset{Dec}{\Longrightarrow}$$
$$\langle \{a \triangleq_{y_3} b, a \triangleq_z b\}; \varnothing; \varnothing; \{x \mapsto g(g(y_3,b),z),\dots\}\rangle \overset{Sol\ \times 2}{\Longrightarrow}$$
$$\langle \varnothing; \{a \triangleq_{y_3} b, a \triangleq_z b\}; \varnothing; \{x \mapsto g(g(y_3,b),z),\dots\}\rangle \overset{Mer}{\Longrightarrow}$$
$$\langle \varnothing; \{a \triangleq_z b\}; \varnothing; \{x \mapsto g(g(z,b),z),\dots\}\rangle$$

Hence, the generalization computed through these derivations is given by $g(g(z,b),z)$. This generalization is less general than the generalization $g(w_1,w_2)$ computed in the first derivation.

4 Abstraction Computation and Completeness

In this section, we construct the *abstraction set* and substitutions from the store S and the delayed set D as in [6]. This set builds less general generalizations after applying AUNIF when $\mathfrak{a}$-symbols are involved in the AUTs and $D \neq \varnothing$. If C-symbols are included, the set is defined using the relation induced by the axioms of $\mathfrak{a}$C-symbols.

Definition 4.1 (Abstraction set)**.** Let t be a term in $\mathfrak{a}$-normal form, and σ be a substitution whose range is in $\mathfrak{a}$-normal form. For E equal to $\mathfrak{a}$ or $\mathfrak{a}$C, the set defined below is the abstraction set of t with respect to σ over E.

$$\uparrow(t,\sigma) := \{r \mid r\sigma \approx_E t,\ r \text{ is in an } \mathfrak{a}\text{-normal form and } var(r) \subseteq dom(\sigma)\}.$$

In words, $\uparrow(t,\sigma)$ is the set of all those E-generalizations of t, whose σ-instances equal t, and that may contain only variables from $dom(\sigma)$, for E equal to $\mathfrak{a}$ or $\mathfrak{a}$C.

Example 4.1. Consider the term $f(a,h(a))$ with f an $\mathfrak{a}$C-symbol, h a syntactic symbol, and the substitution $\sigma = \{x \mapsto a\}$. Then the abstraction set of $f(a,h(a))$ with respect to σ over $\mathfrak{a}$C is:

$$\uparrow(f(a,h(a)),\sigma) = \{f(a,h(a)), f(a,x), f(x,a), f(x,h(x)), f(h(x),x), f(x,h(a)), f(h(a),x)\}.$$

Given a configuration $\langle A;S;D;\theta\rangle$, the AUTs contained in D are of the form $\star \triangleq_x t$ or $t \triangleq_x \star$ for some t. The labels occurring in D also happen in the images of θ. Here, we should interpret $\star$ as any term. Essentially, the abstraction substitution defined below extends θ by replacing the labels of D with a generalization of the non-wildcard term of the associated AUT and some arbitrary term. While this is sufficient for constructing more specific generalizations, we consider restricting the variables occurring in the introduced terms.

Definition 4.2 (Abstraction substitutions)**.** Let $\mathcal{C} = \langle A;S;D;\theta\rangle$ be a configuration such that $D \neq \varnothing$. A substitution τ is called an *abstraction substitution* of $\mathcal{C}$ if $dom(\tau) = labels(D)$, and for each $y \in dom(\tau)$ we have $y\tau \in \uparrow_y(D,S)$, where

$$\uparrow_y(D,S) := \begin{cases} \uparrow(t,\{x \mapsto r \mid l \triangleq_x r \in S,\ \text{for some } l\}) & \text{if } \star \triangleq_y t \in D,\\ \uparrow(s,\{x \mapsto l \mid l \triangleq_x r \in S,\ \text{for some } r\}) & \text{if } s \triangleq_y \star \in D.\end{cases}$$

The set of abstraction substitutions of $\mathcal{C}$ is denoted by $\Psi(D,S)$.

Corollary 4.1. Let $\langle A;S;D;\theta\rangle$ be a configuration such that $D \neq \varnothing$. Then for any $y \in labels(D)$ and $\tau \in \Psi(T,S)$, $var(y\tau) \subseteq labels(S)$.

In contrast with [6], allowing C- and $\mathfrak{a}$-symbols gives rise to new generalizations. The abstraction set enables the computation of all the possibilities for any interpretation of $\star$ in the expansion. The Example below shows how to obtain a generalization from a final configuration and the abstraction set.

Example 4.2. Let $g(\varepsilon_f, f(a,a))$ and $g(\varepsilon_f, f(g(a,a),a))$ be terms modulo $\mathfrak{a}$C with g a C-symbol and f $\mathfrak{a}$C-symbol. One of the possible derivations:

$$\textbf{Derivation 1:} \quad \langle \{g(\varepsilon_f, f(a,a)) \triangleq_x g(\varepsilon_f, f(g(a,a),a))\}; \varnothing; \varnothing; \iota\rangle \overset{Com}{\Longrightarrow}$$
$$\langle \{\varepsilon_f \triangleq_{y_1} f(g(a,a),a), f(a,a) \triangleq_{y_2} \varepsilon_f\}; \varnothing; \varnothing; \{x \mapsto g(y_1,y_2)\}\rangle \overset{ExpLA2}{\Longrightarrow}$$
$$\langle \{\varepsilon_f \triangleq_{w_2} a, f(a,a) \triangleq_{y_2} \varepsilon_f\}; \varnothing; \{\star \triangleq_{w_1} g(a,a)\}; \{x \mapsto g(f(w_1,w_2),y_2),\ldots\}\rangle \overset{ExpRA2}{\Longrightarrow}$$
$$\langle \{\varepsilon_f \triangleq_{w_2} a, a \triangleq_{z_2} \varepsilon_f\}; \varnothing; \{\star \triangleq_{w_1} g(a,a), a \triangleq_{z_1} \star\}; \{x \mapsto g(f(w_1,w_2), f(z_1,z_2)),\ldots\}\rangle \overset{Sol\ \times 2}{\Longrightarrow}$$
$$\langle \varnothing; \{\varepsilon_f \triangleq_{w_2} a, a \triangleq_{z_2} \varepsilon_f\}; \{\star \triangleq_{w_1} g(a,a), a \triangleq_{z_1} \star\}; \{x \mapsto g(f(w_1,w_2), f(z_1,z_2)),\ldots\}\rangle$$

For this final configuration, it is possible to find the abstraction set for related variables w_1 and z_1 in the final delayed set, using the substitutions $\sigma = \{w_2 \mapsto \varepsilon_f, z_2 \mapsto a\}$ and $\rho = \{w_2 \mapsto a, z_2 \mapsto \varepsilon_f\}$.

$$\uparrow_{w_1}(D,S) = \uparrow(g(a,a),\rho) = \{g(a,a), g(a,w_2), g(w_2,a), g(w_2,w_2)\} \text{ and } \uparrow_{z_1}(D,S) = \uparrow(a,\sigma) = \{a, z_2\}$$

Then, the generalization $g(f(g(w_2,a),w_2), f(z_2,z_2))$ of the initial terms is obtained by the substitution $\{w_1 \mapsto g(w_2,a), z_1 \mapsto z_2\} \in \Psi(D,S)$, where D and S are the final delayed and store sets respectively.

Lemma 4.1. Let $\langle A_0; S_0; D_0; \theta_0\rangle \Longrightarrow^* \langle \varnothing; S_n; D_n; \theta_n\rangle$ be a derivation. Then for all $\star \triangleq_u t \in D_n$ (resp. for all $s \triangleq_u \star \in D_n$) and $\tau \in \Psi(D_n,S_n)$, there exists a term r such that $u\tau \in \mathcal{G}_{\mathfrak{a}\mathrm{C}}(r,t)$ (resp. $u\tau \in \mathcal{G}_{\mathfrak{a}\mathrm{C}}(r,s)$).

Proof. Let η be a ground substitution with $dom(\eta) = var(u\tau)$. Then $r = u\tau\eta$. □

Theorem 4.1. Let $\langle A_0; S_0; D_0; \theta_0\rangle \Longrightarrow^* \langle \varnothing; S_n; D_n; \theta_n\rangle$ be a derivation to a final configuration and $s \triangleq_x t \in A_0 \cup S_0$. Then for all $\tau \in \Psi(D_n,S_n)$, $x\theta_n\tau \in \mathcal{G}_{\mathfrak{a}\mathrm{C}}(s,t)$.

Proof. From Theorem 3.2, $x\theta_n \in \mathcal{G}_{\mathfrak{a}\mathrm{C}}(s,t)$. Furthermore, every $u \in labels(D_n)$ is unique, only occurs once in $x\theta_n$, and $u\theta_n\tau = u\tau$. Notice that $var(x\theta_n) = labels(S_n) \cup labels(D_n)$, since any variable in $labels(S_n)$ is generalization of the solved problems and from Lemma 4.1 any application of τ in a variable in $labels(D_n)$ is generalization of the relative subterms of s and t, this implies that $x\theta_n\tau \in \mathcal{G}_{\mathfrak{a}\mathrm{C}}(s,t)$. □

Theorem 4.2 (Completeness)**.** Let $r \in \mathcal{G}_{\mathfrak{a}\mathrm{C}}(s,t)$. Then for all configurations $\langle A; S; D; \theta\rangle$ such that $s \triangleq_x t \in A \cup S$ there exist a final configuration $\langle \varnothing; S'; D'; \theta'\rangle \in \text{AUNIF}(\langle A; S; D; \theta\rangle)$ and $\tau \in \Psi(D',S')$ such that $r \preceq_{\mathfrak{a}\mathrm{C}} x\theta'\tau$.

Proof. (Draft) The proof is by structural induction over r.

Basecase

1. Let r be a variable. For any case for s and t, the algorithm produces a more particular generalization $x\theta''\tau$ than r.
2. Let r be a constant (not absorption constant). Then $s = t = r$ and from a configuration $\langle A; S; D; \theta\rangle$ where $s \triangleq_x t \in A \cup S$, it is possible to reach a configuration $\langle A'; S; D; \theta'\rangle$ using the $\overset{Dec}{\Longrightarrow}$ rule such that $x\theta' = s = t = r$. Thus, for any final configuration $\langle \varnothing; S''; D''; \theta''\rangle \in \text{AUNIF}(\langle A'; S'; D'; \theta'\rangle)$, $r \preceq_{\mathfrak{a}\mathrm{C}} x\theta''\tau$ trivially follows.

Stepcase

1. $r = g(r_1,\dots,r_n)$, $s = g(s_1,\dots,s_n)$, and $t = g(t_1,\dots,t_n)$; and r_i is a generalization of $s_i \triangleq_{y_i} t_i$ for $1 \le i \le n$. From $\langle A;S;D;\theta\rangle$ we can reach $\langle A';S';D';\theta'\rangle$, using the (Dec) rule, such that $s_i \triangleq_{y_i} t_i \in A'$. Observe, if for all $1 \le i < j \le n$, $var(r_i) \cap var(r_j) = \varnothing$, then the generalizations $r_1^*,\dots,r_n^*$, resulting from the IH provide a more particular generalization $g(r_1^*,\dots,r_n^*)$. Otherwise, Let $V \subseteq var(r)$ such that for $z \in V$ there exist $1 \le i < j \le n$ such that $z \in var(r_i) \cap var(r_j)$. For any $z \in V$ with $z = r|_p$, there are three cases to consider when assembling solutions (see Figure 1):
   (i) There exist $(p,q_1) \in pos_C(r,s)$ and $(p,q_2) \in pos_C(r,t)$, such that both the subterms $s|_{q_1}$, and $t|_{q_2}$ cannot be equal to the same $\mathfrak{a}$-constant symbol, and are generalized by z. We will consider final configurations $\langle\varnothing;S^*;D^*;\theta^*\rangle \in \text{AUNIF}(\langle\{s|_{q_1} \triangleq_{x'} t|_{q_2}\};\varnothing;\varnothing;\iota\rangle)$ and substitutions $\tau^* \in \Psi(D^*,S^*)$. We will use substitutions $\theta^*\tau^*$ to assemble generalizations obtained for multiple occurrences of z in the different arguments of r.
   (ii) If there does not exist $(p,\_) \in pos_C(r,s)$, but there exists $(p,q) \in pos_C(r,t)$, then, for some $\mathfrak{a}$-symbol f, there exists $(q_1,q_2) \in \mathfrak{a}p_f(s,t)$, such that $q_2 \sqsubset q$, where $s|_{q_1} = \varepsilon_f$, $head(t|_{q_2}) = f$. For $p' \sqsubset p$ such that $(p',q_2) \in pos_C(r,t)$, $r|_{p'}$ should be a collapsing subterm of r. Then as in the case above, we consider the final configuration $\langle\varnothing;S^*;D^*;\theta^*\rangle \in \text{AUNIF}(\langle\{\varepsilon_f \triangleq_{x'} t|_q\};\varnothing;\varnothing;\iota\rangle)$ and $\tau^* \in \Psi(D^*,S^*)$ to align the instantiation of z in the multiple occurrences into r to the assemble of the generalization. Since $r|_{p'}$ is a collapsing subterm, z may be a variable collapsing this subterm (maybe in another position different from p). But the collapse of $r|_{p'}$ will depend on the occurrences of other variables and how z should be instantiated in other subproblems generated by the application of (Dec).

      Symmetric treatment is applied to the case, where s and t interchange roles.
   (iii) The third case happens when neither exists $(p,\_) \in pos_C(r,s)$ nor $(p,\_) \in pos_C(r,t)$. Let p' and p'' the longest prefixes of p such that $(p',q_1) \in pos_C(r,s)$ and $(p'',q_2) \in pos_C(r,t)$. Then $r|_{p'}$ and $r|_{p''}$ should be collapsing subterms heading by $\mathfrak{a}$-function symbols, say f and g. Thus, $(p',q_1) \in \mathfrak{a}p_f(r,s)$, and $(p'',q_2) \in \mathfrak{a}p_f(r,t)$. W.l.o.g., assuming that $p' \sqsubseteq p''$, we consider the final configuration $\langle\varnothing;S^*;D^*;\theta^*\rangle \in \text{AUNIF}(\langle\{s|_{q_1} \triangleq_{x'} t|_{q_2'}\};\varnothing;\varnothing;\iota\rangle)$, where $(q_1,q_2') \in pos_C(s,t)$. As in the previous cases, a substitution $\tau^* \in \Psi(D^*,S^*)$ will be used to align the instantiation of z in the multiple occurrences into r to assemble the generalization. As in the previous case, z may be the variable collapsing any of the subterms $r|p'$ and $r|_{p''}$ (it may have several occurrences in $r|p'$), but, in general, the collapsing of both these subterms will depend on other variables and how z should be instantiated in other subproblems.

   By the IH, there exists a final configuration $\langle\varnothing;S'';D'';\ \theta''\rangle \in \text{AUNIF}(\langle A';S';D';\theta'\rangle)$ and $\tau_i \in \Psi(D'',S'')$ such that $r_i \preceq_{\mathfrak{a}C} y_i\theta''\tau_i$ where $1 \le i \le n$. Note, we can choose the same configuration $\langle\varnothing;S'';D'';\theta''\rangle$ for all AUTs $s_i \triangleq_{y_i} t_i$ as the algorithm produces all combinations of solutions to the subproblems. Furthermore, we can choose $\langle\varnothing;S'';D'';\theta''\rangle$ such that $S^* \subseteq S''$ and $D^* \subseteq D''$ modulo label renaming, where S^* and D^* are the set fo AUTs mentioned in the cases mentioned above. Now, we define γ_i as the substitution such that $r_i\gamma_i \approx_{\mathfrak{a}C} y_i\theta''\tau_i$. By the above construction, we can safely assume for all $z \in var(r_1) \cap var(r_2)$ such that z has not been replaced by an absorption constant, that $z\gamma_i \approx_{\mathfrak{a}C} z\theta^*\tau^*$ as there exist AUTs corresponding to S^* and D^* in S'' and D'', respectively.

   Now let μ be a substitution and r_i' $(1 \le i \le n)$ be terms such that for all $1 \le i \le n$, $r_i = r_i'\mu$ and $g(r_1',\dots,r_n') \preceq_{\mathfrak{a}C} g(y_1\theta'',\dots,y_n\theta'')$. If μ is the identity substitution, then we are done. Otherwise, we can use μ to construct a $\tau \in \Psi(D'',S'')$. Additionally, we need to consider the $\tau_i \in \Psi(D'',S'')$

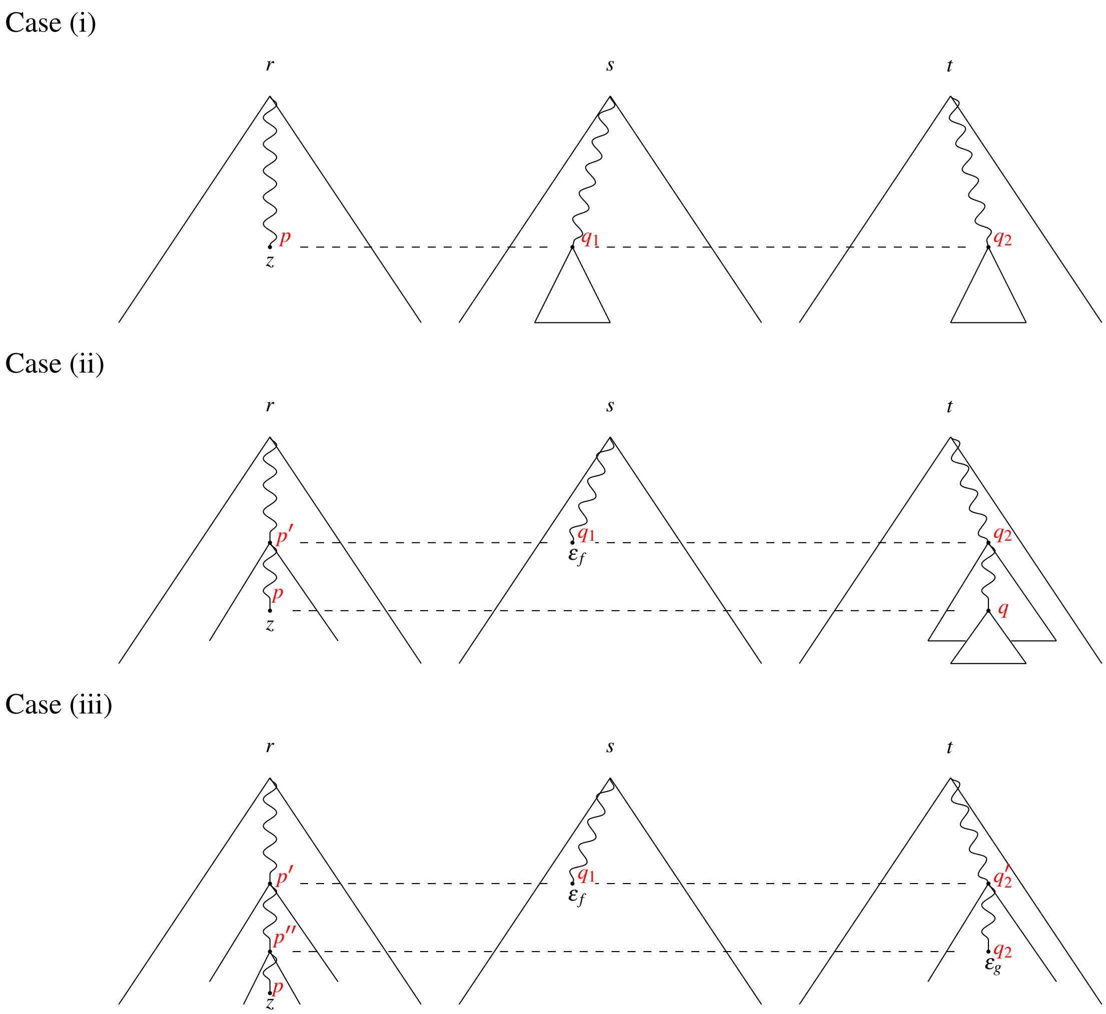

Figure 1: Illustration of the three cases of (Dec) in Theorem 4.2

derived above for each r_i, where $1 \leq i \leq n$, and the corresponding substitutions γ_i. Thus, $r'_i\mu \preceq_{\alpha C} y_i\theta''\tau_i$ and $r'_i\mu\gamma_i \approx_{\alpha C} y_i\theta''\tau_i$.

Now let μ_i^1 and μ_i^2 be substitutions such that $\mu\gamma_i = (\mu_i^1\mu_i^2)|_{dom(\mu\gamma_i)}$ and $r'_i\mu_i^1 \approx_{\alpha C} y_i\theta''$. This is possible given the assumption that $g(r'_1,\ldots,r'_n) \preceq_{\alpha C} g(y_1\theta'',\cdots,y_n\theta'')$. Note that $r'_i\mu_i^1 \approx_{\alpha C} y_i\theta''$ implies that for every $x \in dom(\mu_i^2)$ there exists a $w \in dom(\tau_i)$ such that $w\tau_i \approx_{\alpha C} x\mu_i^2$.

We now construct $\tau \in \Psi(D'',S'')$ using the μ_i^2, that is for all $1 \leq j \leq n$ and $x \in dom(\mu_j^2)$ there exists a $w \in dom(\tau)$ such that $w\tau \approx_{\alpha C} x\mu_j^2$. It now follows that $r_i \preceq_{\alpha C} y_i\theta''\tau$ holds for all $1 \leq i \leq n$ and thus we have shown that $g(r_1,\ldots,r_n) \preceq_{\alpha C} g(y_1,\cdots,y_n)\theta''\tau$.

2. $r = f(r_1,r_2)$, where f is a C- or an αC-symbol and $s = f(s_1,s_2)$ and $t = f(t_1,t_2)$. The next two cases need to be considered: (i) if r_1 is a generalization of $s_1 \triangleq_{y_1} t_1$ and r_2 is a generalization of $s_2 \triangleq_{y_1} t_2$, then we are in a special case of item 1; (ii) if r_1 is a generalization of $s_1 \triangleq_{y_1} t_2$ and r_2 is a generalization of $s_2 \triangleq_{y_1} t_1$, then we apply (Com) to $\langle A;S;D;\theta\rangle$ resulting in the configuration $\langle A';S';D';\theta'\rangle$ where $\{s_i \triangleq_{y_i} t_{3-i}\} \in A'$, for $i \in \{1,2\}$. By IH, at least one of the solutions obtained in

this branch will be structurally smaller than r (see item 1).

3. $r = f(r_1,r_2)$, where f is an $\mathfrak{a}$- or an $\mathfrak{a}$C-symbol and, w.l.o.g, $s = \varepsilon_f$ and $t = f(s_1,s_2)$. Then from $\langle A;S;D;\theta\rangle$ we can derive a configuration $\langle A';S';D';\theta'\rangle$ using the (ExpLA1) rule such that $\star \triangleq_{y_2} s_2 \in D'$ and $\varepsilon_f \triangleq_{y_1} s_1 \in A'$. Now let $\langle\varnothing;S'';D'';\theta''\rangle \in \text{AUNIF}(\langle A';S';D';\theta'\rangle)$ be a final configuration.

   By the induction hypothesis we know that $r_1 \preceq_{\mathfrak{a}C} y_1\theta''\tau_1$ for some $\tau_1 \in \Psi(S'',D'')$. Let μ' be a substitution such that $r_1\mu' \approx_{\mathfrak{a}C} y_1\theta''\tau_1$ and $V_2 \subseteq var(r)$ such that $V_2 \cap var(r_1) = \varnothing$. Using V_2 we define a bijective renaming ν such that for all $z \in V_2$, $z\nu \notin var(r_1\mu') \cup var(r_1)$.

   We will now consider the term $r\nu\mu' = f(r_1\mu', r_2\nu\mu')$. Note that for all variables $z \in var(r_1) \cap var(r_2\nu)$, it must be the case that $z\mu' \preceq_{\mathfrak{a}C} z\mu^*$ where $r_1\mu^* \approx_{\mathfrak{a}C} s_1$ and $r_2\mu^* \approx_{\mathfrak{a}C} s_2$. Thus, observe that $r_2\nu\mu' \preceq_{\mathfrak{a}C} s_2$.

   Now let γ' be a substitution such that $dom(\gamma') = var(r_2\nu\mu')$, $r_2\nu\mu'\gamma' \approx_{\mathfrak{a}C} s_2$, and $r_1\mu'\gamma' \approx_{\mathfrak{a}C} s_1$. Now consider $V_2' = \{z \mid z \in dom(\gamma') \wedge z \notin var(r_1\mu')\}$ and $\nu' = \{z \mapsto l \mid z \in V_2' \wedge z\gamma' = l\}$. Note that $r_2\nu\mu'\nu' \preceq_{\mathfrak{a}C} s_2$ and there exists $t^* \in \uparrow_{y_2}(D'',S'')$ such that $r_2\nu\mu'\nu' \approx_{\mathfrak{a}C} t^*$ by the definition of the *abstraction set*. For terms in $\uparrow_{y_2}(D'',S'')$ we know how to build a $\tau_2 \in \Psi(D'',S'')$.

   Now let μ_1' and μ_2' be substitutions such that $r_1\mu' \approx_{\mathfrak{a}C} r_1'\mu_1'\mu_2'$ and for all $z \in dom(\mu_2')$ there exists $y \in dom(\tau_1)$ such that $z\mu_2' \approx_{\mathfrak{a}C} y\tau_1$. Notice we can apply the same rewriting to $r_2\nu\mu'\nu'$ that is $r_2'\mu_1''\mu_2'' \approx_{\mathfrak{a}C} r_2\nu\mu'\nu'$. We are free to choose the $dom(\nu')$ such that it does not compose with the range of μ'. Thus for variables $z \in var(r_1'\mu_1') \cap var(r_2'\mu_1'')$ such that $z \in dom(\mu_2'')$, there exists $y \in dom(\tau_2)$ such that $z\mu_2'' \approx_{\mathfrak{a}C} y\tau_2$ and $z\mu_2' \approx_{\mathfrak{a}C} y\tau_1$. We can safely assume that the $dom(\tau_2) \cap var(ran(\tau_1)) = \varnothing$, thus we can choose $\tau \in \Psi(D'',S'')$ such that $\tau = \tau_1\tau_2$ as the required substitution; So, $r \preceq_{\mathfrak{a}C} f(y_1,y_2)\theta''\tau$.

4. $r = f(r_1,r_2)$, where f is an $\mathfrak{a}$- or an $\mathfrak{a}$C-symbol and, $s = \varepsilon_f$ and $t = \varepsilon_f$. Then from $\langle A;S;D;\theta\rangle$ we can derive a configuration $\langle A';S';D';\theta'\rangle$ using, w.l.o.g, the (ExpBA1) rule such that $\varepsilon_f \triangleq_{y_1} \star, \star \triangleq_{y_2} \varepsilon_f \in D'$. Now let $\langle\varnothing;S'';D'';\theta''\rangle \in \text{AUNIF}(\langle A';S';D';\theta'\rangle)$ be a final configuration. Because $y_1, y_2 \in labels(D')$, $y_1\theta' = y_1$ and $y_2\theta' = y_2$. Thus, there exist $s \in \uparrow_{y_1}(D'',S'')$, $t \in \uparrow_{y_2}(D'',S'')$, a renaming ν, and $\tau \in \Psi(D'',S'')$ such that $r_1\nu \approx_{\mathfrak{a}C} y_1\tau$ and $r_2\nu \approx_{\mathfrak{a}C} y_2\tau$; this follows from the abstraction set containing all terms $\mathfrak{a}$-equivalent to ε_f under the substitution derived from S''. The substitution ν is required to rename variables in r by the appropriate variables in $labels(S'')$.

   Notice that if $r = f(r_1,r_2)$ is an $\mathfrak{a}$C-symbol, all the possible cases were covered in the abovementioned items.

□

Example 4.3. This Example illustrates cases 1(i) and 1(ii) in the stepcase of the completeness proof sketch of Theorem 4.2.

Consider the AUT $s \triangleq t$, where $s = g(\varepsilon_f, f(h(\varepsilon_f),a))$, and $t = g(f(h(\varepsilon_f),a),\varepsilon_f)$ for g a syntactic symbol, f an $\mathfrak{a}$C-symbol.

1. Let r be the generalization of s and t:

$$r = g(\underbrace{z}_{r_1}, \underbrace{f(h(z),y)}_{r_2})$$

   Let $\mu = \{w_1 \mapsto z, w_2 \mapsto h(z), w_3 \mapsto y\}$. Then:

$$r = g(\underbrace{w_1}_{r'_1}, \underbrace{f(w_2, w_3)}_{r'_2})\mu$$

Consider the variable z: $z \in var(r_1) \cap var(r_2)$. Note that $(1,1) \in pos_C(r,t)$, and z is a generalization of $\varepsilon_f = s|_1$ and $f(h(\varepsilon_f), a) = t|_1$. Then, the case 1(i) applies. Take the final configuration $\langle \varnothing; \{\varepsilon_f \triangleq_{y'} a\}; \{\star \triangleq_{z'} h(\varepsilon_f)\}; \{x' \mapsto f(z', y')\}\}\rangle \in \text{AUNIF}(\langle \{\varepsilon_f \triangleq_{x'} f(h(\varepsilon_f), a)\}; \varnothing; \varnothing; \iota\rangle)$ and $\tau^* = \{z' \mapsto h(\varepsilon_f)\}$. The store of this configuration will get the final configuration of the original problem:

$$\langle \varnothing; \{\varepsilon_f \triangleq_{v_1} a, a \triangleq_{v_2} \varepsilon_f\}; \{\star \triangleq_{u_1} h(\varepsilon_f), h(\varepsilon_f) \triangleq_{u_2} \star\}; \theta\rangle$$

Above, $\theta = \{x \mapsto g(f(u_1, v_1), f(u_2, v_2)), y_1 \mapsto f(u_1, v_1), y_2 \mapsto f(u_2, v_2)\}$.
In particular, $g(w_1, f(w_2, w_3)) \preceq_{\alpha C} g(y_1, y_2)\theta$, and we have:

$$r'_1\mu \preceq_{\alpha C} y_1\theta \underbrace{\{u_1 \mapsto h(\varepsilon_f)\}}_{\tau_1} = f(h(\varepsilon_f), v_1)$$

$$r'_2\mu \preceq_{\alpha C} y_2\theta \underbrace{\{u_2 \mapsto h(f(h(\varepsilon_f), v_1))\}}_{\tau_2} = f(h(f(h(\varepsilon_f), v_1)), v_2)$$

For $\gamma_1 = \{z \mapsto f(h(\varepsilon_f), v_1)\}$ and $\gamma_2 = \{y \mapsto v_2, z \mapsto f(h(\varepsilon_f), v_1)\}$, $r'_1\mu\gamma_1 \approx_{\alpha C} y_1\theta\tau_1$ and $r'_2\mu\gamma_2 \approx_{\alpha C} y_2\theta\tau_2$. Hence, the substitution $\mu_1^1\mu_1^2$ restricted to $dom(\mu\gamma_1)$ and $\mu_2^1\mu_2^2$ restricted to $dom(\mu\gamma_2)$ are such that:

$$\mu\gamma_1 = (\underbrace{\{w_1 \mapsto f(u, v_1)\}}_{\mu_1^1} \underbrace{\{u \mapsto h(\varepsilon_f)\}}_{\mu_1^2})|_{dom(\mu\gamma_1)},$$

$$\mu\gamma_2 = (\underbrace{\{w_2 \mapsto v, w_3 \mapsto v_2\}}_{\mu_2^1} \underbrace{\{v \mapsto h(f(h(\varepsilon_f, v_1)))\}}_{\mu_2^2})|_{dom(\mu\gamma_2)}.$$

Finally, we define $\tau = \{u_1 \mapsto h(\varepsilon_f), u_2 \mapsto h(f(h(\varepsilon_f), v_1))\}$. Thus,

$$g(z, f(h(z), y)) \preceq_{\alpha C} g(f(h(\varepsilon_f), v_1), f(h(f(h(\varepsilon_f), v_1)), v_2)).$$

2. Let r be the generalization of s and t:

$$r = g(\underbrace{f(z, h(f(z, y)))}_{r_1}, \underbrace{f(h(f(z, y)), y)}_{r_2})$$

Let $\mu = \{w_1 \mapsto z, w_2 \mapsto h(f(z, y)), w_3 \mapsto h(f(z, y)), w_4 \mapsto y\}$. Then:

$$r = g(\underbrace{f(w_1, w_2)}_{r'_1}, \underbrace{f(w_3, w_4)}_{r'_2})\mu$$

Consider the variable z: $z \in var(r_1) \cap var(r_2)$. Note that $(1,1) \in \mathfrak{a}p_f(s,t)$ and $(1.1,1.2) \in pos_C(r,t)$, and z is a generalization of $\varepsilon_f = s|_1$ and $a = t|_{1.2}$. Then, the case 1(ii) applies.

Take the final configuration $\langle \varnothing; \{\varepsilon_f \triangleq_{x'} a\}; \varnothing; \iota \rangle \in$ AUNIF$(\langle \{\varepsilon_f \triangleq_{x'} a\}; \varnothing; \varnothing; \iota \rangle)$ and $\tau^* = id$. The store of this configuration will get the final configuration of the original problem:

$$\langle \varnothing; \{\varepsilon_f \triangleq_{v_1} a, a \triangleq_{u_2} \varepsilon_f\}; \{\star \triangleq_{u_1} h(\varepsilon_f), h(\varepsilon_f) \triangleq_{v_2} \star\}; \theta \rangle$$

Above, $\theta = \{x \mapsto g(f(u_1,v_1), f(u_2,v_2)), y_1 \mapsto f(u_1,v_1), y_2 \mapsto f(u_2,v_2)\}$.

In particular, $g(f(w_1,w_2), f(w_3,w_4)) \preceq_{\mathfrak{a}C} g(y_1,y_2)\theta$, and we have:

$$r_1'\mu \preceq_{\mathfrak{a}C} y_1\theta \underbrace{\{v_1 \mapsto h(f(u_1,v_2))\}}_{\tau_1} = f(u_1, h(f(u_1,v_2)))$$

$$r_2'\mu \preceq_{\mathfrak{a}C} y_2\theta \underbrace{\{u_2 \mapsto h(f(u_1,v_2))\}}_{\tau_2} = f(h(f(u_1,v_2)), v_2)$$

For $\gamma_1 = \gamma_2 = \{z \mapsto u_1, y \mapsto v_2\}$, $r_1'\mu\gamma_1 \approx_{\mathfrak{a}C} y_1\theta\tau_1$ and $r_2'\mu\gamma_2 \approx_{\mathfrak{a}C} y_2\theta\tau_2$. Hence, the substitution $\mu_1^1\mu_1^2$ restricted to $dom(\mu\gamma_1)$ and $\mu_2^1\mu_2^2$ restricted to $dom(\mu\gamma_2)$ are such that:

$$\mu\gamma_1 = (\underbrace{\{w_1 \mapsto u_1, w_2 \mapsto v_1\}}_{\mu_1^1} \underbrace{\{v_1 \mapsto h(f(u_1,v_2))\}}_{\mu_1^2})|_{dom(\mu\gamma_1)},$$

$$\mu\gamma_2 = (\underbrace{\{w_3 \mapsto u_2, w_4 \mapsto v_2\}}_{\mu_2^1} \underbrace{\{u_2 \mapsto h(f(u_1,v_2))\}}_{\mu_2^2})|_{dom(\mu\gamma_2)}.$$

Finally, we define $\tau = \{v_1 \mapsto h(f(u_1,v_2)), u_2 \mapsto h(f(u_1,v_2))\}$. Thus,

$$g(f(z,h(f(z,y))), f(h(f(z,y)),y)) \preceq_{\mathfrak{a}C} g(f(u_1,h(f(u_1,v_2))), f(h(f(u_1,v_2)),v_2)).$$

5 Conclusion

This work discusses current work to extend the previous algorithm by the authors for anti-unification modulo $\mathfrak{a}$-theories (in [6]) to anti-unification modulo $\mathfrak{a}$C-theories. The proposed algorithm deals with theories that may contain all combinations of $\mathfrak{a}$, C-, and $\mathfrak{a}$C-symbols.

The introduced algorithm is proved to be terminating and sound, and the paper drafts the crucial considerations in the inductive analysis of its completeness.

Immediate future possible steps in this investigation are empowering the algorithm to deal with associativity and unity, properties of great interest by their applicability and theoretical complexity (as mentioned in the introduction, theories with more than one unital operator are of anti-unification type nullary [10]). Another research target of great interest is the development of formal verifications of anti-unification algorithms as those developed for C- and AC-unification in the prototype verification system PVS [4, 5].

Acknowledgments.

This work was supported by the Czech Science Foundation Grant No. 22-06414L; the Austrian Science Fund (FWF) project P 35530; Cost Action CA20111 EuroProofNet; the Brazilian agency CNPq, Grant Universal 409003/21-2, and RG 313290/21-0; and the Brazilian Federal District Research Foundation FAPDF, Grant DE 00193-00001175/2021-11. The Brazilian Higher Education Council (CAPES) supported our Brazilian-Austrian cooperation under CAPES PrInt financial code 001.

References

[1] María Alpuente, Santiago Escobar, Javier Espert & José Meseguer (2014): *ACUOS: A System for Modular ACU Generalization with Subtyping and Inheritance*. In: European Conference on Logics in Artificial Intelligence JELIA, LNCS 8761, Springer, p. 573–581, doi:10.1007/978-3-319-11558-0_40.

[2] María Alpuente, Santiago Escobar, Javier Espert & José Meseguer (2019): *ACUOS^2: A High-Performance System for Modular ACU Generalization with Subtyping and Inheritance*. In: European Conference on Logics in Artificial Intelligence, JELIA, LNCS 11468 LNAI, Springer, pp. 171–181, doi:10.1007/978-3-030-19570-0_11.

[3] María Alpuente, Santiago Escobar, Javier Espert & José Meseguer (2022): *Order-sorted equational generalization algorithm revisited*. Ann. Math. Artif. Intell. 90(5), pp. 499–522, doi:10.1007/s10472-021-09771-1.

[4] Mauricio Ayala-Rincón, Washington de Carvalho Segundo, Maribel Fernández, Gabriel Ferreira Silva & Daniele Nantes-Sobrinho (2021): *Formalising nominal C-unification generalised with protected variables*. Math. Struct. Comput. Sci. 31(3), pp. 286–311, doi:10.1017/S0960129521000050.

[5] Mauricio Ayala-Rincón, Maribel Fernández, Gabriel Ferreira Silva & Daniele Nantes Sobrinho (2022): *A Certified Algorithm for AC-Unification*. In: 7th International Conference on Formal Structures for Computation and Deduction, FSCD, LIPIcs 228, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 8:1–8:21, doi:10.4230/LIPICS.FSCD.2022.8.

[6] Mauricio Ayala-Rincón, David M. Cerna, Andrés Felipe González Barragán & Temur Kutsia (2024): *Equational Anti-Unification over Absorption Theories*. In: Automated Reasoning - 12th International Joint Conference, IJCAR 2024, Nancy, France, Proceedings, Lecture Notes in Computer Science 14740, Springer, pp. 317–337, doi:10.1007/978-3-031-63501-4_17.

[7] Adam D. Barwell, Christopher Brown & Kevin Hammond (2018): *Finding parallel functional pearls: Automatic parallel recursion scheme detection in Haskell functions via anti-unification*. Future Gener. Comput. Syst. 79, pp. 669–686, doi:10.1016/j.future.2017.07.024.

[8] David Cao, Rose Kunkel, Chandrakana Nandi, Max Willsey, Zachary Tatlock & Nadia Polikarpova (2023): *babble: Learning Better Abstractions with E-Graphs and Anti-unification*. Proceedings of the ACM on Programming Languages 7(POPL), pp. 396–424, doi:10.1145/3571207.

[9] David M. Cerna (2020): *Anti-unification and the theory of semirings*. Theor. Comput. Sci. 848, pp. 133–139, doi:10.1016/j.tcs.2020.10.020.

[10] David M. Cerna & Temur Kutsia (2020): *Unital Anti-Unification: Type and Algorithms*. In: 5th Int. Conference on Formal Structures for Computation and Deduction, FSCD, LIPIcs 167, pp. 26:1–26:20, doi:10.4230/LIPICS.FSCD.2020.26.

[11] David M. Cerna & Temur Kutsia (2023): *Anti-unification and Generalization: A Survey*. In: Proceedings of the 32nd Int. Joint Conference on Artificial Intelligence, IJCAI, pp. 6563–6573, doi:10.24963/ijcai.2023/736.

[12] Sonu Mehta, Ranjita Bhagwan, Rahul Kumar, Chetan Bansal, Chandra Shekhar Maddila, Balasubramanyan Ashok, Sumit Asthana, Christian Bird & Aditya Kumar (2020): *Rex: Preventing Bugs and Misconfiguration in Large Services Using Correlated Change Analysis*. In: 17th USENIX Symposium on Networked Systems

Design and Implementation, NSDI 2020, USENIX Association, pp. 435–448. Available at `https://www.usenix.org/conference/nsdi20/presentation/mehta`.

[13] Jörg H. Siekmann (1989): *Unification Theory*. *J. Symb. Comput.* 7(3/4), pp. 207–274, doi:10.1016/S0747-7171(89)80012-4.