On Anti-unification in Absorption Theories

Andrés Felipe González Barragán[1], David M. Cerna[2], Mauricio Ayala-Rincón[1], and Temur Kutsia[3]

[1] University of Brasilia (UNB), Brasília D.F., Brazil
agonzale@risc.uni-linz.ac.at,ayala@unb.br
[2] Czech Academy of Sciences Institute of Computer Science (CAS ICS), Prague, Czech Republic
dcerna@cs.cas.cz
[3] Research Institute for Symbolic Computation (RISC), Johannes Kepler University (JKU), Linz, Austria
kutsia@risc.jku.at

Abstract

The anti-unification problem for theories with absorption constants and operators such as zero for product or true and false for disjunction and conjuction is addressed. A proved sound algorithm that computes generalizations is introduced. Although the problem is at least of type infinitary, the algorithm computes a finite set of final configurations from which the least general generalizations are built.

1 Introduction

Several identification problems, searching for similarities and analogies, are related to equational questions. In particular, problems such as matching and unification look for identifications through computational procedures over algebraic expressions. In contrast, generalization procedures look for similarities or regularities in the structure of expressions. Generalization, known as anti-unification, is a dual problem to unification. Over half a century ago, anti-unification was formulated by Plotkin [6] and Reynolds [7] as a question on generalizing first-order algebraic terms. Nevertheless, it has been studied on numerous equational theories, for example, theories with associative, commutative [1], idempotent [4] operators, and unital theories [5, 2], and combinations of these properties on the semiring theory [3]. This document discusses anti-unification for one of the theories that many common operators have, known as absorption theories (`Abs`). These theories contain binary operator(s) f with related absorption constant ε_f. The operators satisfy the axioms $f(x,\varepsilon_f) \approx \varepsilon_f$ and/or $f(\varepsilon_f,x) \approx \varepsilon_f$. It is possible to find the absorption theories in Semirings, Rings for one operation, and in Boolean algebras for both operations.

1.1 Preliminaries

We consider an alphabet consisting of a set $\mathcal{F}$ of function symbols with their arities, and a countable set of variables $\mathcal{V}$ with the symbol '_' (anonymous variable). Terms over this alphabet, $\mathcal{T}(\mathcal{F},\mathcal{V})$, have 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*. `Abs` denotes an equational theory with one or more absorption symbol(s). Equality of terms modulo `Abs` is denoted as $s \approx_{\mathtt{Abs}} t$. For a substitution θ, $\mathit{Dom}(\theta)$ and $\mathit{Rvar}(\theta)$ denote the sets of variables in the domain of θ and the variables occurring in the range of θ. $\mathit{Var}(e)$ will denote the set of variables occurring in an expression. The *composition* of substitutions σ and ρ is written as $\rho\sigma$ and θ_0^n abbreviates n compositions: $\theta_0\theta_1\ldots\theta_n$.

Definition 1 (Abs-generalization, $\preceq_{\texttt{Abs}}$, *lgg*, solution)**.** *The precedence generalization relation of* Abs*, holds for terms r and s, $r \preceq_{\texttt{Abs}} s$, if a substitution σ exists such that $r\sigma \approx_{\texttt{Abs}} s$, read as "$r$ is* more general *than s." Also, $\prec_{\texttt{Abs}}$ and $\simeq_{\texttt{Abs}}$ denote the strict precedence and equivalence relations induced by $\preceq_{\texttt{Abs}}$, respectively. If $r \preceq_{\texttt{Abs}} s$ and $r \preceq_{\texttt{Abs}} t$, r is said to be an* Abs*-generalization of s and t.*

A least general generalization *(lgg) of the terms s and t modulo* Abs *, is a term r that is an* Abs*-generalization of s and t, and such that for all other* Abs*-generalization r' of s and t, $r \simeq_{\texttt{Abs}} r'$.*

Each Abs*-generalization r of s and t is associated with two substitutions σ and ρ such that $r\sigma \approx_{\texttt{Abs}} s$ and $r\rho \approx_{\texttt{Abs}} t$. The triple $\langle r, \sigma, \rho\rangle$ is the* Abs-solution *associated to the* Abs*-generalization r.*

Example 1.1. Consider the terms ε_f and $f(f(b,c),a)$ over the Abs theory. Notice that $f(f(b,x),a)$ is an *lgg* of ε_f and $f(f(b,c),a)$ modulo Abs. Indeed, $\sigma = \{x \mapsto \varepsilon_f\}$ and $\rho = \{x \mapsto c\}$ satisfy $f(f(b,x),a)\sigma = f(f(b,\varepsilon_f),a) \approx_{\texttt{Abs}} \varepsilon_f$ and $f(f(b,x),a)\rho = f(f(b,c),a)$, then $\langle f(f(b,x),a), \sigma, \rho\rangle$ is an Abs-solution.

Definition 2 (Anti-unification equation (AUE) and valid set of AUEs)**.** *An* anti-unification equation *(AUE) is a triple of the form $s \overset{x}{\triangleq} t$, where x is the* label *of the AUE, and s and t are terms. The set of labels of a set of AUEs, A, is denoted as $labels(A)$. A* valid set of AUEs *is a set of AUEs where all the labels are different.*

Definition 3 (Minimal complete set of generalizations)**.** *A* minimal complete set of generalizations *(mcsg) over the equational theory E, of the terms s and t, denoted as $mcsg_E(s,t)$, is a set of terms such that:*

1. *Each $r \in mcsg_E(s,t)$ is an E-generalization of s and t,*
2. *for each E-generalization r of s and t, there exist $r' \in mcsg_E(s,t)$ such that $r \preceq_E r'$ and,*
3. *if $r, r' \in mcsg_E(s,t)$ and $r \preceq_E r'$ then $r = r'$.*

Definition 4 (Anti-unification type)**.** *The anti-unification type of the equational theory E is said to be* unitary *or* finitary *if $mcsg_E(s,t)$ is unitary or finite, for all terms s and t, respectively. It is* nullary *if $mcsg_E(s,t)$ does not exist for some terms s and t. Otherwise, it is said to be* infinitary.

2 A sound algorithm for anti-unification modulo Abs

The algorithm is built from a set of inference rules that transform quadruples, called configurations, of the form $\langle A; S; T; \theta\rangle$, where A is the valid set of *unsolved* AUEs; S is the *store*, the valid set of solved AUEs; T is the *abstraction*, the valid set of AUEs, which store the comparisons through anonymous variables in an expansion of some absorption constant. The anonymous variables are assumed to mimic the structure of the other term in the AUE. Finally, θ is a *substitution* mapping the labels of the AUEs to their respective generalizations. We only consider configurations with the following properties:

(i) The sets $labels(A), labels(S), labels(T)$ and $Dom(\theta)$ are pairwise disjoint.

(ii) $Rvar(\theta) = labels(A) \cup labels(S) \cup labels(T)$.

Table 1: generalization rules for `Abs` theory

$(\stackrel{Dec}{\Longrightarrow})$	$$\frac{\langle\{f(s_1,\ldots,s_n)\overset{x}{\triangleq}f(t_1,\ldots,t_n)\}\cup A;S;T;\theta\rangle}{\langle\{s_1\overset{y_1}{\triangleq}t_1,\ldots,s_n\overset{y_n}{\triangleq}t_n\}\cup A;S;T;\theta\{x\mapsto f(y_1,\ldots,y_n)\}\rangle}$$ where f is an n-ary function symbol with $n\geq 0$, and $y_1,\ldots,y_n$ are fresh variables.
$(\stackrel{Sol}{\Longrightarrow})$	$$\frac{\langle\{s\overset{x}{\triangleq}t\}\cup A;S;T;\theta\rangle}{\langle A;\{s\overset{x}{\triangleq}t\}\cup S;T;\theta\rangle}$$ where $head(s)\neq head(t)$ and they are non-related absorption symbols.
$(\stackrel{ExpLA1}{\Longrightarrow})$	$$\frac{\langle\{\varepsilon_f\overset{x}{\triangleq}f(t_1,t_2)\}\cup A;S;T;\theta\rangle}{\langle\{\varepsilon_f\overset{y_1}{\triangleq}t_1\}\cup A;S;\{\_\overset{y_2}{\triangleq}t_2\}\cup T;\theta\{x\mapsto f(y_1,y_2)\}\rangle}$$ where f is an absorption function symbol, and y_1,y_2 are fresh variables.
$(\stackrel{ExpLA2}{\Longrightarrow})$	$$\frac{\langle\{\varepsilon_f\overset{x}{\triangleq}f(t_1,t_2)\}\cup A;S;T;\theta\rangle}{\langle\{\varepsilon_f\overset{y_2}{\triangleq}t_2\}\cup A;S;\{\_\overset{y_1}{\triangleq}t_1\}\cup T;\theta\{x\mapsto f(y_1,y_2)\}\rangle}$$ where f is an absorption function symbol, and y_1,y_2 are fresh variables.
$(\stackrel{ExpRA1}{\Longrightarrow})$	$$\frac{\langle\{f(s_1,s_2)\overset{x}{\triangleq}\varepsilon_f\}\cup A;S;T;\theta\rangle}{\langle\{s_1\overset{y_1}{\triangleq}\varepsilon_f\}\cup A;S;\{s_2\overset{y_2}{\triangleq}\_\}\cup T;\theta\{x\mapsto f(y_1,y_2)\}\rangle}$$ where f is an absorption function symbol, and y_1,y_2 are fresh variables.
$(\stackrel{ExpRA2}{\Longrightarrow})$	$$\frac{\langle\{f(s_1,s_2)\overset{x}{\triangleq}\varepsilon_f\}\cup A;S;T;\theta\rangle}{\langle\{s_2\overset{y_2}{\triangleq}\varepsilon_f\}\cup A;S;\{s_1\overset{y_1}{\triangleq}\_\}\cup T;\theta\{x\mapsto f(y_1,y_2)\}\rangle}$$ where f is an absorption function symbol, and y_1,y_2 are fresh variables.
$(\stackrel{Mer}{\Longrightarrow})$	$$\frac{\langle\emptyset;\{s\overset{x}{\triangleq}t,s\overset{y}{\triangleq}t\}\cup S;T;\theta\rangle}{\langle\emptyset;\{s\overset{y}{\triangleq}t\}\cup S;T;\theta\{x\mapsto y\}\rangle}$$

The algorithm Ant_Unif is an exhaustive application of the inference rules in Table 1 to transform an *initial configuration* $\langle A;\emptyset;\emptyset;\iota\rangle$ into a set of final configurations with an empty set of unsolved AUEs of the form $\langle\emptyset,S,T,\theta\rangle$ and there are no different AUEs with the same terms s,t and with a different label. The *starting substitution* ι is given by a set of bindings of the form $\{x_{st}\mapsto x\mid x\in \mathit{labels}(A)\}$, where each x_{st} is a distinguished starting label. The final substitution θ instantiates the initial labels into the common structure of pairs of terms in each initial AUE $s\overset{x}{\triangleq}t$, so that $x\theta$ is an `Abs`-generalization of s and t. The final store S determines the manner in which `Abs`-solutions $\langle x\theta,\sigma,\rho\rangle$ are built from the generalization, so that $x\theta\sigma\approx_{\texttt{Abs}} s$ and $x\theta\rho\approx_{\texttt{Abs}} s$.

The rules in Table 1 read as: Decompose $(\stackrel{Dec}{\Longrightarrow})$, Solve $(\stackrel{Sol}{\Longrightarrow})$, Expansions for Left Absorption, $(\stackrel{ExpLA1}{\Longrightarrow})$, $(\stackrel{ExpLA2}{\Longrightarrow})$ and for Right Absorption $(\stackrel{ExpRA1}{\Longrightarrow})$, $(\stackrel{ExpRA2}{\Longrightarrow})$, and Merge $(\stackrel{Mer}{\Longrightarrow})$.

Lemma 2.1 (Preservation of configurations under Ant_Unif)**.** If Ant_Unif is applied to any configuration $\langle A;S:T;\theta\rangle$ the result is a configuration too.

Proof. Initial configurations hold the conditions of a configuration. And, by case analysis, it is proved that the conditions are preserved after each rule application. □

Theorem 2.1 (Termination)**.** The procedure Ant_Unif is terminating. Particularly, for any configuration $\langle A;S;T;\theta\rangle$, it outputs a finite set of configurations of the form $\langle\emptyset;S';T';\theta'\rangle$.

Proof. Observe that the sum of the length of the terms in the AUEs in A of the configurations strictly decreases after each application of any of the rules except for the rule (Mer). However, the rule (Mer) is only applied when the set of AUEs in the configuration is empty, decreasing the number of AUEs in the store. Therefore the procedure terminates.

Furthermore, since for any configuration, Ant_Unif generates at most two recursive calls by König Lemma, the output is a finite set of configurations. Notice that only the branching instructions in lines 7 and 13 of the procedure Ant_Unif generate two branches in the execution flow, obtained after applications of rules (ExpLA1), (ExpLA2), (ExpRA1), and (ExpRA2). □

From termination, for each $\langle A;S;T;\theta\rangle$, Ant_Unif returns a finite number of final configurations, we denoted this set by Ant_Unif($\langle A;S;T;\theta\rangle$), the set of *computed configurations.*

Definition 5 (Left and right substitutions related to a set of AUEs)**.** *Let W be a finite valid set of AUEs. The left and right substitutions related to W are defined as follows:*

$$\sigma_W = \{y \mapsto s \mid s \overset{y}{\triangleq} t \in W\}, \text{ and } \rho_W = \{y \mapsto t \mid s \overset{y}{\triangleq} t \in W\}.$$

Definition 6 (Computed solutions)**.** *Let $\mathcal{D}$ be a derivation from a configuration to a final configuration: $\langle A;S;T;\theta\rangle \Longrightarrow^* \langle \varnothing;S';T';\theta'\rangle$. The computed generalization is defined as*

$$\langle \{x\theta'\}_{x\in L}, \sigma_{\mathcal{D}}, \rho_{\mathcal{D}}\rangle$$

Above, $L = labels(A) \cup labels(S) \cup labels(T)$ and $\sigma_{\mathcal{D}} = \sigma_{S'\cup T'}$ and $\rho_{\mathcal{D}} = \rho_{S'\cup T'}$.

Theorem 2.2 (Soundness)**.** If $\langle A_0;S_0;T_0;\theta_0\rangle \Longrightarrow^* \langle \varnothing;S_n;T_n;\theta_0^n\rangle$ is a derivation from the configuration $\langle A_0;S_0;T_0;\theta_0\rangle$ to a final configuration, then for each $s \overset{x}{\triangleq} t \in A_0 \cup S_0 \cup T_0$, $x\theta_0^n$ is a generalization of s and t, and $x\theta_0^n\sigma_{\mathcal{D}} \approx_{\texttt{Abs}} s$ and $x\theta_0^n\rho_{\mathcal{D}} \approx_{\texttt{Abs}} t$, where $\langle \{x\theta_0^n\}_{x\in L}, \sigma_{\mathcal{D}}, \rho_{\mathcal{D}}\rangle$ is the computed solution.

Proof. The induction is over the length derivation that the configuration reaches the final configuration.

Base Case. We consider a final configuration. Hence, for all $s \overset{x}{\triangleq} t \in S_0 \cup T_0$, $x\theta_0 = x$, by the first condition of configuration. Also, $x\theta_0\sigma_{\mathcal{D}} = s$ and $x\theta_0\rho_{\mathcal{D}} = t$ hold.

Induction Step. Consider a derivation of the form:

$$\langle A_0;S_0;T_0;\theta_0\rangle \Longrightarrow \langle A_1;S_1;T_1;\theta_0^1\rangle \Longrightarrow^n \langle \varnothing;S_{n+1};T_{n+1};\theta_0^{n+1}\rangle$$

Above $\langle \varnothing;S_{n+1};T_{n+1};\theta_0^{n+1}\rangle$ is a final configuration. We assume the hypothesis for the derivation $\langle A_1;S_1;T_1;\theta_0^1\rangle \Longrightarrow^n \langle \varnothing;S_{n+1};T_{n+1};\theta_0^{n+1}\rangle$.

Notice that, by the first condition of configuration, and for any $s \overset{x}{\triangleq} t \in A_1 \cup S_1 \cup T_1$, it holds $x\theta_0^{n+1} = x\theta_2^{n+1}$. Similarly, for any $s \overset{x}{\triangleq} t \in A_0 \cup S_0 \cup T_0$, it holds $x\theta_0^{n+1} = x\theta_1^{n+1}$. The first step depends on the rule application.

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

$$\langle \{f(s_1,\dots,s_m) \overset{y}{\triangleq} f(t_1,\dots,t_m)\} \cup A';S_0;T_0;\theta_0\rangle \overset{Dec}{\Longrightarrow}$$

$$\langle \{s_1 \overset{x_1}{\triangleq} t_1,\dots,s_m \overset{x_m}{\triangleq} t_m\} \cup A';S_1;T_1;\theta_0\{y \mapsto f(x_1,\dots,x_m)\}\rangle \Longrightarrow^n \langle \varnothing;S_{n+1};T_{n+1};\theta_0^{n+1}\rangle$$

By induction hypothesis, all the AUEs in A_1 are generalized by the substitution θ_0^{n+1}. This implies that $x_i\theta_0^{n+1}$ is a generalization of s_i and t_i, for $1 \le i \le m$; i.e., $x_i\theta_2^{n+1}\sigma_{\mathcal{D}} \approx_{\texttt{Abs}} s_i$ and $x_i\theta_2^{n+1}\rho_{\mathcal{D}} \approx_{\texttt{Abs}} t_i$ for $1 \le i \le m$. Hence, $y\theta_0^{n+1}\rho_{\mathcal{D}} = f(x_1\theta_2^{n+1}\rho_{\mathcal{D}},\dots,x_m\theta_2^{n+1}\rho_{\mathcal{D}}) \approx_{\texttt{Abs}} f(s_1,\dots,s_m)$ and $y\theta_0^{n+1}\sigma_{\mathcal{D}} = f(x_1\theta_2^{n+1}\sigma_{\mathcal{D}},\dots,x_m\theta_2^{n+1}\sigma_{\mathcal{D}}) \approx_{\texttt{Abs}} f(t_1,\dots,t_m)$.

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

$$\langle \{s \overset{y}{\triangleq} t\} \cup A'; S_0; T_0; \theta_0 \rangle \overset{Sol}{\Longrightarrow} \langle \{A'; \{s \overset{y}{\triangleq} t\} \cup S_0; T_0; \theta_0\}\rangle \Longrightarrow^* \langle \varnothing; S_{n+1}; T_{n+1}; \theta_0^{n+1}\rangle$$

   By induction hypotheses, θ_0^{n+1} generalize all the AUEs with labels in S_1 then $y\theta_0^{n+1}$ is a generalization of s and t.

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

$$\langle \{\varepsilon_f \overset{y}{\triangleq} f(s,t)\} \cup A'; S_0; T_0; \theta_0 \rangle \overset{ExpLA1}{\Longrightarrow}$$

$$\langle \{\varepsilon_f \overset{x_1}{\triangleq} s\} \cup A'; S_1; \{\_ \overset{x_2}{\triangleq} t\} \cup T_0; \theta_0\{y \mapsto f(x_1,x_2)\}\rangle \Longrightarrow^n \langle \varnothing; S_{n+1}; T_{n+1}; \theta_0^{n+1}\rangle$$

   By induction hypothesis all the AUEs in S_1 and T_1 are generalized by the substitution θ_0^{n+1}. This implies that $x_1\theta_2^{n+1}$ is a generalization of ε_f and s, $x_2\theta_2^{n+1}$ is a generalization of $\_$ and t with the substitutions $\sigma_{\mathcal{D}}$ and $\rho_{\mathcal{D}}$, respectively. Additionally, $y\theta_0^{n+1}\sigma_{\mathcal{D}} = f(x_1\theta_2^{n+1}\sigma_{\mathcal{D}}, x_2\theta_2^{n+1}\sigma_{\mathcal{D}}) \approx_{\mathtt{Abs}} f(\varepsilon_f, \_) \approx_{\mathtt{Abs}} \varepsilon_f$ and $y\theta_0^{n+1}\rho_{\mathcal{D}} = f(x_1\theta_2^{n+1}\rho_{\mathcal{D}}, x_2\theta_2^{n+1}\rho_{\mathcal{D}}) \approx_{\mathtt{Abs}} f(s,t)$. Hence, $y\theta_0^{n+1}$ is a generalization of the terms ε_f and $f(s,t)$.

4. The analysis of **(ExpLA2), (ExpRA1)** and **(ExpRA2)** rules is analogous to the analysis of the rule **(ExpLA1)**.

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

$$\langle \varnothing; \{s \overset{y}{\triangleq} t, s \overset{z}{\triangleq} t\} \cup S'; T_0; \theta_0 \rangle \overset{Mer}{\Longrightarrow} \langle \{\varnothing; \{s \overset{z}{\triangleq} t\} \cup S'; T_0; \theta_0\{y \mapsto z\}\}\rangle \Longrightarrow^* \langle \varnothing; S_{n+1}; T_{n+1}; \theta_0^{n+1}\rangle$$

   Notice that $\theta_0^1 = \theta_0\{y \mapsto z\}$, where z is the label of the AUE $\{s \overset{z}{\triangleq} t\} \in S_0$. By induction hypothesis $z\theta_2^{n+1}$ is a generalization of s and t. Then, $y\theta_0^{n+1} = y\{y \mapsto z\}\theta_2^{n+1} = z\theta_2^{n+1}$ is a generalization of s and t with substitutions $\sigma_{\mathcal{D}}$ and $\rho_{\mathcal{D}}$, respectively.

□

Definition 7 (Abstraction)**.** *Let t be a term in* Abs*-normal form and σ be a substitution with images in* Abs*-normal form. The abstraction of t with respect to σ is the set:*

$$\mathtt{Abstract}(t,\sigma) \coloneqq \{r \mid r\sigma \approx_{\mathtt{Abs}} t,\ r \text{ is an } \mathtt{Abs}\text{-normal form, and } Var(r) \subseteq \{\_\} \cup Dom(\sigma)\}$$

Example 2.1. Let $t = h(\varepsilon_f)$ and $\sigma = \{y \mapsto a, v \mapsto \varepsilon_f\}$. Then:

$$\mathtt{Abstract}(t,\sigma) = \{h(\varepsilon_f), h(v), h(f(v,\_)), h(f(\_,v)), h(f(v,v)), h(f(f(v,\_),\_)), \dots\}$$

Definition 8 (Abstraction substitution)**.** *Given a configuration $\langle A; S; T; \theta\rangle$ with $T \neq \varnothing$, an abstraction substitution of this configuration is any substitution τ such that (i) $Dom(\tau) = labels(T)$, and (ii) for each $y \in Dom(\tau)$, $y\tau \in \mathtt{Abstract}_y(T,S)$,*

Above the set $\mathtt{Abstract}_y(T,S)$ is defined as $\mathtt{Abstract}_y(T,S) = \mathtt{Abstract}(t,\rho_S)$ if $\{\_ \overset{y}{\triangleq} t\} \subseteq T$; or as $\mathtt{Abstract}_y(T,S) = \mathtt{Abstract}(s,\sigma_S)$ if $\{s \overset{y}{\triangleq} \_\} \subseteq T$. The set of all possible abstraction substitutions of this configuration is denoted as $\Psi(T,S)$.

If the anonymous variables occur within a term $t \in \mathtt{Abstract}_y(T,S)$, we interpret each occurrence as an arbitrary (possibly different) ground term. For instance, in the term $h(f(v,\_))$ of the Example 2.1, we can substitute any ground term s in place of the anonymous variable, resulting in $h(f(v,s))\sigma \approx_{\mathtt{Abs}} t$.

After applying the algorithm Ant_Unif, it is possible to obtain less general generalizations by considering each possible substitution of τ. This can potentially lead to an infinite set of generalizations, depending on the cardinality of the abstraction sets.

Example 2.2. We illustrate how the algorithm Ant_Unif and the application of the possible τ solves the anti-unification problem $\varepsilon_f \triangleq f(a, f(h(a), b))$. The initial configuration is given by $\langle \varepsilon_f \overset{x}{\triangleq} f(a, f(h(a), b)); \varnothing; \varnothing; \iota\rangle$.

$$
\begin{aligned}
\text{Branch 1.}\quad & \langle \{\varepsilon_f \overset{x}{\triangleq} f(a, f(h(a), b))\}; \varnothing; \varnothing; \iota\rangle \overset{\mathit{ExpLA1}}{\Longrightarrow} \\
& \langle \{\varepsilon_f \overset{y}{\triangleq} a\}; \varnothing; \{\_ \overset{z}{\triangleq} f(h(a), b)\}; \iota\{x \mapsto f(y,z)\}\rangle \overset{\mathit{Sol}}{\Longrightarrow} \\
& \langle \varnothing; \{\varepsilon_f \overset{y}{\triangleq} a\}; \{\_ \overset{z}{\triangleq} f(h(a), b)\}; \iota\{x \mapsto f(y,z)\}\rangle
\end{aligned}
$$

Since $\mathtt{Abstract}_z(f(h(a),b), \{y \mapsto a\}) = \{f(h(a),b), f(h(y),b)\}$, then the set of substitutions generated by the abstraction is $\Psi(T,S) = \{\{z \mapsto f(h(a),b)\}, \{z \mapsto f(h(y),b)\}\}$. Hence, the *lgg*s of this branch are given by $f(y, f(h(a), b))$ and $f(y, f(h(y), b)$ with the substitutions given by the store: $\sigma_1 = \{y \mapsto \varepsilon_f\}$, $\rho_1 = \{y \mapsto a\}$.

$$
\begin{aligned}
\text{Branch 2.}\quad & \langle \{\varepsilon_f \overset{x}{\triangleq} f(a, f(h(a), b))\}; \varnothing; \varnothing; \iota\rangle \overset{\mathit{ExpLA2}}{\Longrightarrow} \\
& \langle \{\varepsilon_f \overset{v}{\triangleq} f(h(a), b)\}; \varnothing; \{\_ \overset{u}{\triangleq} a\}; \iota\{x \mapsto f(u,v)\}\rangle \overset{\mathit{ExpLA1}}{\Longrightarrow} \\
& \langle \{\varepsilon_f \overset{y}{\triangleq} h(a)\}; \varnothing; \{\_ \overset{u}{\triangleq} a, \_ \overset{z}{\triangleq} b\}; \iota\{x \mapsto f(u, f(y,z)), v \mapsto f(y,z)\}\rangle \overset{\mathit{Sol}}{\Longrightarrow} \\
& \langle \varnothing; \{\varepsilon_f \overset{y}{\triangleq} h(a)\}; \{\_ \overset{u}{\triangleq} a, \_ \overset{z}{\triangleq} b\}; \iota\{x \mapsto f(u, f(y,z)), v \mapsto f(y,z)\}\rangle
\end{aligned}
$$

Since $\mathtt{Abstract}_u(T,S) = \mathtt{Abstract}(a, \{y \mapsto h(a)\}) = \{h(a)\}$ for the label u and the set $\mathtt{Abstract}_z(T,S) = \mathtt{Abstract}(b, \{y \mapsto h(a)\}) = \{b\}$ for the label z, from the final abstraction T and store S. Hence, the unique *lgg* of this branch is $f(a, f(y, b))$, with the substitutions given by the store: $\sigma_2 = \{y \mapsto \varepsilon_f\}$, $\rho_2 = \{y \mapsto h(a)\}$.

$$
\begin{aligned}
\text{Branch 3.}\quad & \langle \{\varepsilon_f \overset{x}{\triangleq} f(a, f(h(a), b))\}; \varnothing; \varnothing; \iota\rangle \overset{\mathit{ExpLA2}}{\Longrightarrow} \\
& \langle \{\varepsilon_f \overset{v}{\triangleq} f(h(a), b)\}; \varnothing; \{\_ \overset{u}{\triangleq} a\}; \iota\{x \mapsto f(u,v)\}\rangle \overset{\mathit{ExpLA2}}{\Longrightarrow} \\
& \langle \{\varepsilon_f \overset{z}{\triangleq} b\}; \varnothing; \{\_ \overset{u}{\triangleq} a, \_ \overset{y}{\triangleq} h(a)\}; \iota\{x \mapsto f(u, f(y,z)), v \mapsto f(y,z)\}\rangle \overset{\mathit{Sol}}{\Longrightarrow} \\
& \langle \varnothing; \{\varepsilon_f \overset{z}{\triangleq} b\}; \{\_ \overset{u}{\triangleq} a, \_ \overset{y}{\triangleq} h(a)\}; \iota\{x \mapsto f(u, f(y,z)), v \mapsto f(y,z)\}\rangle
\end{aligned}
$$

From the final abstraction T and store S. $\mathtt{Abstract}_u(T,S) = \mathtt{Abstract}(a, \{z \mapsto b\}) = \{a\}$ and $\mathtt{Abstract}_y(T,S) = \mathtt{Abstract}(h(a), \{z \mapsto b\}) = \{h(a)\}$, , the unique *lgg* of this branch is: $f(a, f(h(a), z))$ with the substitutions given by the store: $\sigma_3 = \{z \mapsto \varepsilon_f\}$, $\rho_3 = \{z \mapsto b\}$.

Hence, all the Abs-solutions computed:

$$
\langle f(y, f(h(a), b)), \sigma_1, \rho_1\rangle, \langle f(y, f(h(y), b), \sigma_1, \rho_1\rangle, \langle f(a, f(y, b)), \sigma_2, \rho_2\rangle, \langle f(a, f(h(a), z)), \sigma_3, \rho_3\rangle.
$$

3 Conclusion and work in progress

Configuration preservation and termination provide the requirements to prove the soundness of the ANT_UNIF algorithm. However, to conclude that the problem type is infinitary, proving it is non-nullary is required. Furthermore, analyzing the completeness of ANT_UNIF will require additional considerations and specialized notions. In particular, the notion of E-solutions (triples of the form $\langle r, \sigma, \rho \rangle$ as given in Definition 1) is enough for analyzing soundness. In contrast, for completeness, given a configuration $\langle A; S; T; \theta \rangle$ and a solution of all equational generalization questions in the set of unsolved AUEs $A = \{s_i \overset{x_i}{\triangleq} t_i\}_{i \in I}$, a generalization should be given as a "synchronized" set of generalizations and substitutions σ and ρ of the form $\langle \langle r_{x_i} \rangle_{i \in I}, \sigma, \rho \rangle$. The same should be done for the computed solutions and, of course, such a solution should consider the store, the abstraction and the substitution parts of the configuration.

Future work will focus on proving that for any generalization r of the AUE $s \overset{x}{\triangleq} t$, it would be possible to find a generalization $x\theta$ computed by ANT_UNIF and a τ generated by the final configuration of the computation such that $x\theta\tau$ is less general than r. At this point, establishing whether ANT_UNIF is complete depends on additional formal analysis. Still, we expect the algorithm to be complete as it is clear that any solution must be a substitution instance of a syntactic generalization.

Acknowledgments. The preparation of this paper has been supported by the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES), under the PrInt program. It is partially supported by the Austrian Science Fund (FWF), under project P 35530, the Georgian Rustaveli National Science Foundation Project FR-21-16725, and by the Czech Science Foundation Grant No. 22-06414L, and the Cost Action CA20111 EuroProofNet.

References

[1] María Alpuente, Santiago Escobar, Javier Espert, and José Meseguer. A modular order-sorted equational generalization algorithm. *Information and Computation*, 235:98–136, 2014.

[2] María Alpuente, Santiago Escobar, José Meseguer, and Julia Sapiña. Order-sorted equational generalization algorithm revisited. *Ann. Math. Artif. Intell.*, 90(5):499–522, 2022.

[3] David M. Cerna. Anti-unification and the theory of semirings. *Theo. Com. Sci.*, 848:133–139, 2020.

[4] David M. Cerna and Temur Kutsia. Idempotent anti-unification. *ACM Trans. Comput. Log.*, 21(2):10:1–10:32, 2020.

[5] David M. Cerna and Temur Kutsia. Unital Anti-Unification: Type and Algorithms. In *5th International Conference on Formal Structures for Computation and Deduction, FSCD*, volume 167 of *LIPIcs*, pages 26:1–26:20, 2020.

[6] Gordon D. Plotkin. A note on inductive generalization. *Machine Intelligence 5*, 5:153–163, 1970.

[7] Jhon C. Reynolds. Transformational system and the algebric structure of atomic formulas. *Machine Intelligence 5*, 5:135–151, 1970.