On Anti-unification in Absorption Theories

Andrés Felipe González Barragan[†] (UnB)
David M. Cerna (CAS ICS), Mauricio Ayala-Rincón (UnB)
Temur Kutsia (RISC - U. Linz)

[†]Author supported by a Brazilian CAPES Scholarship

37th Int. Workshop on Unification (UNIF), FSCD 2023
Rome, July 2nd, 2023

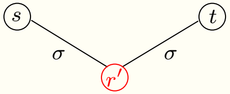
RISC Software GmbH

Outline

Motivation

Unification

Goal: find a substitution that identifies two expressions (terms).



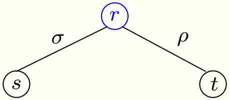
where $t\sigma \approx r' \approx s\sigma$.

Example 1

Identify the terms $h(g(a), y)$ and $h(g(z), f(w))$. Using the substitution $\sigma = \{y \mapsto f(w), z \mapsto a\}$ the expressions *unify* to $h(g(a), b)$.

Anti-unification

Goal: find the commonalities between two expressions (terms).
An expression with such commonalities is called a *generalization*.



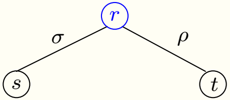
where $r\sigma \approx s$ and $r\rho \approx t$.

Example 2

Generalize the terms $h(g(a), y)$ and $h(g(z), f(w))$.

Anti-unification

Goal: find the commonalities between two expressions (terms).
An expression with such commonalities is called a *generalization*.

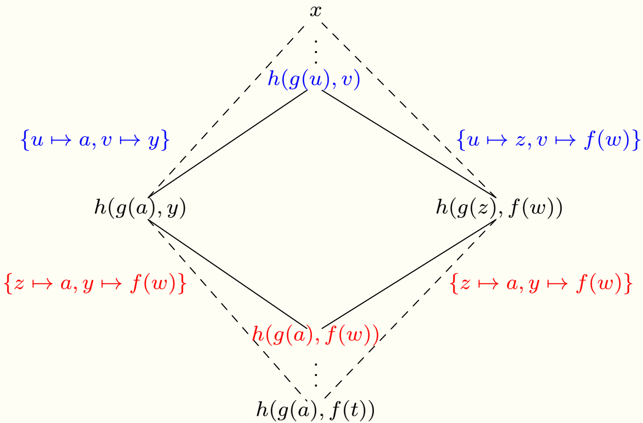


where $r\sigma \approx s$ and $r\rho \approx t$.

Example 2

Generalize the terms $h(g(a), y)$ and $h(g(z), f(w))$.
generalization: $h(g(u), v)$, with substitutions $\sigma = \{u \mapsto a, v \mapsto y\}$ and $\rho = \{u \mapsto z, v \mapsto f(w)\}$.

Unification and Anti-unification



Motivation

One interesting example of verbatim plagiarism:

- (Original sentence). All around the world, technology is continuing to become a part of everyday life, and its capabilities are progressing rapidly.
- (Possibly sentence with plagiarism). All over the world, technology became a part of our lives, and its capabilities are progressing very quickly.

Motivation

Then finding the common parts and the differences in the sentences:

- All around the world, technology is continuing to become a part of everyday life, and its capabilities are progressing rapidly.
- All over the world, technology became a part of our lives, and its capabilities are progressing very quickly.

All □ the world, technology □ a part of □, and its capabilities are progressing □.

Motivation

Applications of anti-unification include:

- searching parallel recursion schemes to transform sequential algorithms into parallel algorithms (Barwell et al. [BBH18]);
- preventing bugs and misconfigurations in software (Mehta et al. [MBK+20]);
- finding duplicate code and similarities;
- detecting code clones (i.e., plagiarism).

Absorption Theory

- The alphabet consists of a countable set of variables $\mathcal{V}$ and set $\mathcal{F}$ of function and with a special constant symbol $\star$ (The wild card).
- Terms over this alphabet, $\mathcal{T}(\mathcal{F}, \mathcal{V})(\mathcal{T})$ and $\mathcal{T}(\mathcal{F} \cup \{\star\}, \mathcal{V})(\mathcal{T}_\star)$, defined as usually:

$$t := x \mid f(t_1, \ldots, t_n)$$

- A finite set E that consists of equations $s \approx t$.
- A preorder $\preceq_E$, which states that $s \preceq_E t$ if there exists a substitution σ such that $s\sigma \approx_E t$.

Absorption Theory

Type of anti-unification problems

The type of an anti-unification modulo E problem is classified as below.

- *Nullary*(0): if there are terms s and t such that $\mathtt{mcsg}_E(s,t)$ does not exist. Also, called *type zero*.
- *Unitary*(1): if for all s and t, $\mathtt{mcsg}_E(s,t)$ has just one generalization.
- *Finitary*(ω): if for all s and t, $\mathtt{mcsg}_E(s,t)$ has more than one generalization.
- *Infinitary*(∞): there are terms s and t such that $\mathtt{mcsg}_E(s,t)$ is infinite.

Type of some Theories

Theory	Type	Authors and References	Procedure or Term
Syntactic ($\emptyset$)	1	G. Plotkin and [Plo70, Rey70] J. Reynolds	Dec, Sol, Rec
Associativity (A)	ω	M. Alpuente et al. [AEEM14]	A-left, A-right
Commutativity(C)	ω	M. Alpuente et al. [AEEM14]	C
Unital (U)	ω	D. Cerna [CK20a]	Start-C, Sat-C, M
Idempotency$_{\geq 1}$ (I)	∞	D. Cerna and T. Kutsia [CK20a]	M, Id-left,Id-right Id-both (1,2,3)
Unital$_{\geq 2}$ (U$_2$)	0	D. Cerna and T. Kutsia [CK20b]	$e_f \triangleq e_g$ $f(g(x,y),x)$

Type of some Theories

Theory	Type	Authors and References	Procedure or Term
AC, ACU	ω	M. Alpuente et al. [AEEM14]	AC-left, AC-right
AU_2, CU_2, ACU_2	0	D. Cerna and T. Kutsia [CK20b]	$e_f \triangleq e_g$ $f(g(x,y),x)$
$(UI)_2$, $(ACUI)_2$	0	D. Cerna and T. Kutsia [CK20b]	$e_f \triangleq e_g$ $f(g(f(x,y),e_f),x)$
Semirings (S), SC	0	D. Cerna [Cer20]	$e_f \triangleq e_g$ $\prod_{i=1}^{n} x$

Type of some Theories

Theory	Type	Authors and References	Procedure or Term
Absorption$_{\geq 1}$ ($\mathtt{Abs}$)	?	—	—
(ACU)$_2$, (ACU)$_2\mathtt{Abs}$	0	D. Cerna [Cer20]	$e_f \triangleq e_g$
			$\prod_{i=1}^{n} x$
Simply-typed λ-calculus	0	D. Cerna and M. Buran[BC22]	$\lambda xy.f(x) \triangleq \lambda xy.f(y)$
I$\mathtt{Abs}$,(UI)$_2\mathtt{Abs}$	$\emptyset, \infty$?	—	—

Absorption Theory

- An *anti-unification equation* (AUE) between s and t in a normal form is denoted by $s \overset{x}{\triangleq}_E t$, where x is called as label.
- A *valid set of AUEs* is a set of AUEs where all the labels are different.
- An AUE $s \overset{x}{\triangleq} t$ is *solved* if $head(s)$ and $head(t)$ are not related absorption symbols, where $s, t \in \mathcal{T}$.
- An AUE $s \overset{x}{\triangleq} t$ is *wild* if one of the terms is the wild card and the other belongs to $\mathcal{T}$.

Absorption Theory

Absorption Theory

Absorption is an important algebraic attribute in some magmas: for some function symbol f there is a constant ε_f such that

$$f(x, \varepsilon_f) \approx \varepsilon_f, \text{ or/and } f(\varepsilon_f, x) \approx \varepsilon_f$$

Equational theories with these equations are called an absorption theories (`Abs`).

Example 3

Let's find one generalization of the AUE $\varepsilon_f \triangleq_{\texttt{Abs}} f(f(a, b), c)$.

Anti-Unification Algorithm for Absorption Theory

The idea of the algorithm is to expand the ε_f to get the generalization:

$\varepsilon_f \overset{x}{\triangleq} f(f(a,b),c)$	x
$f(\varepsilon_f, c) \overset{x}{\triangleq} f(f(a,b),c)$	x
$\varepsilon_f \overset{y}{\triangleq} f(a,b), c \overset{z}{\triangleq} c$	$f(y,z)$
$f(\varepsilon_f, b) \overset{y}{\triangleq} f(a,b)$	$f(y,c)$
$\varepsilon_f \overset{u}{\triangleq} a, b \overset{v}{\triangleq} b$	$f(f(u,v),c)$
$\varepsilon_f \overset{u}{\triangleq} a$	$f(f(u,b),c)$

Anti-Unification Algorithm for Absorption Theory

Algorithm for absorption theory

To build the algorithm we consider a quadruple $\langle A; S; T; \theta \rangle$ as a *configuration* in each step of the procedure, 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 wild AUEs;
- θ is a *substitution* mapping the labels of the AUEs to the term of the generalization given by the rules.

Anti-Unification Algorithm for Absorption Theory

Inference Rules

Then we define the next rules

(Dec): **Decompose**

$$\langle \{ f(s_1, \dots, s_n) \overset{x}{\triangleq} f(t_1, \dots, t_n) \} \sqcup A; S; \theta \rangle$$

$$\overset{Dec}{\Longrightarrow} \langle \{ s_1 \overset{y_1}{\triangleq} t_1, \dots, s_n \overset{y_n}{\triangleq} t_n \} \cup A; S; \theta \{ x \mapsto f(y_1, \dots, y_n) \} \rangle$$

For f any function symbol, $n > 0$, and $y_1, \dots, y_n$ are fresh variables.

Anti-Unification Algorithm for Absorption Theory

Inference Rules

(Solve): **Solve**

$$\langle \{s \overset{x}{\triangleq} t\} \sqcup A; S; T; \theta \rangle \overset{Sol}{\Longrightarrow} \langle A; \{s \overset{x}{\triangleq} t\} \cup S; T; \theta \rangle$$

Where $head(s) \neq head(t)$ are not related absorption symbols.

(Mer): **Merge**

$$\langle \emptyset; \{s \overset{x}{\triangleq} t\} \cup \{s \overset{y}{\triangleq} t\} \cup S; \theta \rangle \overset{Mer}{\Longrightarrow} \langle \emptyset; \{s \overset{y}{\triangleq} t\} \cup S; \theta\{x \mapsto y\} \rangle$$

Anti-Unification Algorithm for Absorption Theory

Inference Rules

(ExpLA1): **Expansion for Absorption, Left 1**

$$\langle \{ \varepsilon_f \overset{x}{\triangleq} f(t_1, t_2) \} \sqcup A; S; T; \theta \rangle$$

$$\overset{\text{ExpLA1}}{\Longrightarrow} \langle \{ \varepsilon_f \overset{y_1}{\triangleq} t_1 \} \cup A; S; \{ \star \overset{y_2}{\triangleq} t_2 \} \cup T; \theta\{ x \mapsto f(y_1, y_2) \} \rangle$$

(ExpLA2): **Expansion for Absorption, Left 2**

$$\langle \{ \varepsilon_f \overset{x}{\triangleq} f(t_1, t_2) \} \sqcup A; S; T; \theta \rangle$$

$$\overset{\text{ExpLA2}}{\Longrightarrow} \langle \{ \varepsilon_f \overset{y_2}{\triangleq} t_2 \} \cup A; S; \{ \star \overset{y_1}{\triangleq} t_1 \} \cup T; \theta\{ x \mapsto f(y_1, y_2) \} \rangle$$

Anti-Unification Algorithm for Absorption Theory

Inference Rules

(ExpRA1): **Expansion for Absorption, Right 1**

$$\langle \{f(s_1, s_2) \overset{x}{\triangleq} \varepsilon_f\} \sqcup A; S; T; \theta \rangle$$

$$\overset{ExpRA1}{\Longrightarrow} \langle \{s_1 \overset{y_1}{\triangleq} \varepsilon_f\} \cup A; S; \{s_2 \overset{y_2}{\triangleq} \star\} \cup T; \theta\{x \mapsto f(y_1, y_2)\} \rangle$$

(ExpRA2): **Expansion for Absorption, Right 2**

$$\langle \{f(s_1, s_2) \overset{x}{\triangleq} \varepsilon_f\} \sqcup A; S; T; \theta \rangle$$

$$\overset{ExpRA2}{\Longrightarrow} \langle \{s_2 \overset{y_2}{\triangleq} \varepsilon_f\} \cup A; S; \{s_1 \overset{y_1}{\triangleq} \star\} \cup T; \theta\{x \mapsto f(y_1, y_2)\} \rangle$$

Algorithm ANT_UNIF

Algorithm ANT_UNIF

The algorithm ANT_UNIF is an exhaustive application of the inference rules 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.

Example 4

Apply ANT_UNIF to the anti-unification problem $g(\varepsilon_f, a) \triangleq g(f(h(\varepsilon_f), a), \varepsilon_f)$.

Algorithm ANT_UNIF

$$\langle\{g(\varepsilon_f, a) \overset{x}{\triangleq} g(f(h(\varepsilon_f), a), \varepsilon_f)\}; \emptyset; \emptyset; \iota\rangle \overset{Dec}{\Longrightarrow}$$

$$\langle\{\varepsilon_f \overset{w_1}{\triangleq} f(h(\varepsilon_f), a), a \overset{w_2}{\triangleq} \varepsilon_f\}; \emptyset; \emptyset; \{x \mapsto g(w_1, w_2)\}\rangle \overset{\text{ExpLA2}}{\Longrightarrow}$$

$$\langle\{\varepsilon_f \overset{u_2}{\triangleq} a, a \overset{w_2}{\triangleq} \varepsilon_f\}; \emptyset; \{\star \overset{u_1}{\triangleq} h(\varepsilon_f)\}; \{x \mapsto g(f(u_1, u_2), w_2)\}\rangle \overset{Sol}{\Longrightarrow}$$

$$\langle\{\varepsilon_f \overset{u_2}{\triangleq} a\}; \{a \overset{w_2}{\triangleq} \varepsilon_f\}; \{\star \overset{u_1}{\triangleq} h(\varepsilon_f)\}; \{x \mapsto g(f(u_1, u_2), w_2)\}\rangle \overset{Sol}{\Longrightarrow}$$

$$\langle\emptyset; \{\varepsilon_f \overset{u_2}{\triangleq} a, a \overset{w_2}{\triangleq} \varepsilon_f\}; \{\star \overset{u_1}{\triangleq} h(\varepsilon_f)\}; \{x \mapsto g(f(u_1, u_2), w_2)\}\rangle$$

Algorithm ANT_UNIF

$$g(f(u_1, u_2), w_2)$$

$$\sigma = \{u_1 \mapsto \star, u_2 \mapsto \varepsilon_f, w_2 \mapsto a\}$$

$$\rho = \{u_1 \mapsto h(\varepsilon_f), u_2 \mapsto a, w_2 \mapsto \varepsilon_f\}$$

$$g(f(\star, \varepsilon_f), a) \qquad g(f(h(\varepsilon), a), \varepsilon_f)$$

$$\approx_{\texttt{Abs}}$$

$$g(\varepsilon_f, a)$$

Then, $g(f(u_1, u_2), w_2)$ is a generalization with the substitutions σ and ρ.

Abstraction Set

Abstraction Set

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:

$$\uparrow(t, \sigma) := \{r \mid r\sigma \approx_{\texttt{Abs}} t,\ r \text{ is an } \texttt{Abs}\text{-normal form, and } \mathit{Var}(r) \subseteq \mathit{Dom}(\sigma)\}$$

Algorithm ANT_UNIF

Example 5

Find the abstraction set of $h(\varepsilon_f)$ with respect to $\rho = \{u_2 \mapsto a, w_2 \mapsto \varepsilon_f\}$:

$$\uparrow (h(\varepsilon_f), \rho) = \{h(\varepsilon_f), h(w_2), h(f(w_2, \star)), h(f(\star, w_2)), h(f(u_2, w_2)), \ldots\}$$

Where $\star$ could be replaced by a term whose variables are included in *Dom*(ρ). For example, $h(f(w_2, a))$ and $h(f(w_2, h(g(u_2, w_2))))$ belong to the abstraction set.

Algorithm ANT_UNIF

Continue with Example 4:

$$(\varepsilon_f, a) \triangleq g(f(h(\varepsilon_f), a), \varepsilon_f)$$

The final branch:

$$\langle \emptyset; \{\varepsilon_f \overset{u_2}{\triangleq} a, a \overset{w_2}{\triangleq} \varepsilon_f\}; \{\star \overset{u_1}{\triangleq} h(\varepsilon_f)\}; \{x \mapsto g(f(u_1, u_2), w_2)\}\rangle$$

To find a less general generalization, it is possible to replace the variable u_1 in the generalization $g(f(u_1, u_2), w_2)$ for one of the elements of the abstraction set $\uparrow (h(\varepsilon_f), \rho)$.

Algorithm ANT_UNIF

Then, the term $g(f(h(f(w_2, a)), u_2), w_2)$ is a generalization too.

$$g(f(h(f(w_2, a)), u_2), w_2)$$

$\sigma = \{u_2 \mapsto \varepsilon_f, w_2 \mapsto a\}$

$\rho = \{u_2 \mapsto a, w_2 \mapsto \varepsilon_f\}$

$$g(f(h(f(a, a)), \varepsilon_f), a) \quad\quad g(f(h(f(\varepsilon_f, a)), a), \varepsilon_f)$$

$$\approx_{\texttt{Abs}} \quad\quad \approx_{\texttt{Abs}}$$

$$g(\varepsilon_f, a) \quad\quad g(f(h(\varepsilon_f), a), \varepsilon_f)$$

Algorithm for ANT_UNIF

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 $\langle \emptyset; S'; T'; \theta'\rangle$.

Algorithm for ANT_UNIF

Soundness

If $\langle A_0; S_0; T_0; \theta_0 \rangle \Longrightarrow^* \langle \emptyset; S_n; T_n; \theta_n \rangle$ is a derivation to a final configuration, then for each $s \overset{x}{\triangleq} t \in A_0 \cup S_0 \cup T_0$:

- $x\theta_n$ is a generalization of s and t, and $x\theta_n\sigma_{\mathcal{D}} \approx_{\mathtt{Abs}} s$ and
- $x\theta_n\rho_{\mathcal{D}} \approx_{\mathtt{Abs}} t$.

Conclusions and Future work

- We design an algorithm for anti-unification in absorption theories.
  The algorithm is terminating and sound.
- We conjecture that the algorithm is complete.
  The complete set of least general generalizers can be built from the computed substitutions, the store, and the *abstraction set*.

References

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

Adam D. Barwell, Christopher Brown, and Kevin Hammond.
Finding parallel functional pearls: Automatic parallel recursion scheme detection in haskell functions via anti-unification.
Future Gener. Comput. Syst., 79:669–686, 2018.

Michal Buran and David M. Cerna.
One or nothing: Anti-unification over the simply-typed lambda calculus.
CoRR, abs/2207.08918, 2022.

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

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

David M. Cerna and Temur Kutsia.
Unital anti-unification: type algorithms.
5th International Conference on Formal Structures for Computation and Deduction, FSCD, 167(6):26:1–26:20, 2020.

Sonu Mehta, Ranjita Bhagwan, Rahul Kumar, Chetan Bansal, Chandra Maddila, B. Ashok, Sumit Asthana, Christian Bird, and Aditya Kumar.
Rex: Preventing bugs and misconfiguration in large services using correlated change analysis.
In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 435–448, 2020.

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

John C. Reynolds.
Transformational system and the algebric structure of atomic formulas.
Machine Intelligence 5, 5:135–151, 1970.