



**HAL**  
open science

## Semantic anti-unification in general algebras

Christian Antic

► **To cite this version:**

| Christian Antic. Semantic anti-unification in general algebras. 2023. hal-04207922v1

**HAL Id: hal-04207922**

**<https://hal.science/hal-04207922v1>**

Preprint submitted on 14 Sep 2023 (v1), last revised 22 Feb 2024 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SEMANTIC ANTI-UNIFICATION IN GENERAL ALGEBRAS

CHRISTIAN ANTIĆ

*christian.antic@icloud.com*  
*Vienna University of Technology*  
*Vienna, Austria*

**ABSTRACT.** Motivated by recent applications to analogical reasoning, this paper initiates a *semantic* theory of anti-unification within general algebras.

**Keywords:** Generalization, Universal Algebra, Tree Automata and Languages

## 1. INTRODUCTION AND PRELIMINARIES

Anti-unification (or generalization) (cf. Cerna & Kutsia, 2023) is the “dual” operation to the well-studied unification operation (cf. Baader & Snyder, 2001). More formally, given two terms  $s$  and  $t$ , unification searches for a most general unifier  $\sigma$  satisfying  $s\sigma = t\sigma$ , whereas syntactic anti-unification searches for a least general generalization  $u$  such that  $s = u\sigma_1$  and  $t = u\sigma_2$ , for some substitutions  $\sigma_1, \sigma_2$ . Notice the difference between the two operations: while unification computes a substitution, anti-unification computes a term. Therefore saying that the two operations are “dual” — as it is often done in the literature — is a bit misleading.

Syntactic anti-unification has been first studied by Plotkin (1970) and Reynolds (1970), and it has found numerous applications in theoretical computer science and artificial intelligence, as for example in inductive logic programming (Muggleton, 1991) (cf. Cropper & Morel, 2021; Cropper, 2022), programming by example (Gulwani, 2016), library learning and compression (Cao et al., 2023), and, in the form of  $E$ -generalization (i.e. anti-unification modulo theory) (Heinz, 1995; Burghardt, 2005), in analogy-making (Weller & Schmid, 2007; Schmidt et al., 2014) (for further applications, see e.g. Barwell et al., 2018; de Sousa et al., 2021; Vanhoof & Yernaux, 2019).

Motivated by recent applications to analogical reasoning (Antić, 2022, 2023a, 2023b), this paper initiates a *semantic* theory of anti-unification within general algebras. More formally, given an algebra  $\mathfrak{A}$  in the sense of universal algebra (cf. Burris & Sankappanavar, 2000, §II), we shall define the set of minimally general generalizations (or mggs) of two *elements* of the algebra (instead of terms). That is, given  $a$  and  $b$ , the set  $a \sqcap_{\mathfrak{A}} b$  of mggs in  $\mathfrak{A}$  will consist of all terms  $s$  such that  $a$  and  $b$  are within the range of the induced term function  $s^{\mathfrak{A}}$  and  $s$  is minimal with respect to a suitable semantic generalization ordering (see §2). Notice that this operation has *no* “dual” in the theory of unification since it makes no sense to try to “unify” two constant symbols  $a$  and  $b$  by finding a substitution  $\sigma$  such that  $a\sigma = b\sigma$  (which holds iff  $a = b$ ).

Semantic anti-unification as proposed in this paper is related to  $E$ -generalization or anti-unification modulo equational theory. In fact, if the underlying algebra  $\mathfrak{A}$  has an equational axiomatization  $E$ , then semantic and  $E$ -generalization are two sides of the same coin. This line of work is not pursued in this paper and therefore remains an interesting line of future research (see Problem 2).

After introducing the general framework in §2, we briefly study anti-unification in monounary algebras (§3), finite unary algebras (§4) where we establish a close connection to finite automata, and finite algebras (§6) where we establish a close connection to tree automata. We do not claim to obtain any deep results — the purpose of this paper is rather to introduce the framework of semantic anti-unification and to *initiate* its study.

**Preliminaries.** We expect the reader to be fluent in basic universal algebra as it is presented for example in Burris and Sankappanavar (2000, §II) and Baader and Nipkow (1998, §3).

A *language*  $L$  of algebras is a set of *function symbols*<sup>1</sup> together with a *rank function*  $r : L \rightarrow \mathfrak{N}$ , and a denumerable set  $X$  of *variables* distinct from  $L$ . Terms are formed as usual from variables and function symbols. We denote the set of variables occurring in a term  $s$  by  $X(s)$  and we say that  $s$  has *rank*  $k$  iff  $X(s) = \{x_1, \dots, x_k\}$ . We denote the rank of  $s$  by  $r(s)$ .

An  $L$ -*algebra*  $\mathfrak{A}$  consists of a non-empty set  $A$ , the *universe* of  $\mathfrak{A}$ , and for each function symbol  $f \in L$ , a function  $f^{\mathfrak{A}} : A^{r(f)} \rightarrow A$ , the *functions* of  $\mathfrak{A}$  (the *distinguished elements* of  $\mathfrak{A}$  are the 0-ary functions). We will not distinguish between distinguished elements and their 0-ary function symbols. Every term  $s$  induces a function  $s^{\mathfrak{A}}$  on  $\mathfrak{A}$  in the usual way. In this paper, we shall not distinguish between terms inducing the same function on the underlying algebra, which is common practice in mathematics where one does usually not distinguish, for example, between the terms  $2x^2$  and  $(1+1)x^2$ .

The *term algebra*  $\mathfrak{T}_L(X)$  over  $L$  and  $X$  is the algebra we obtain by interpreting each function symbol  $f \in L$  by

$$f^{\mathfrak{T}_L(X)} : \mathfrak{T}_L(X)^{r(f)} \rightarrow \mathfrak{T}_L(X) : (s_1, \dots, s_{r(f)}) \mapsto f(s_1, \dots, s_{r(f)}).$$

## 2. ANTI-UNIFICATION IN GENERAL ALGEBRAS

This is the main section of the paper. Here, we shall define what semantic anti-unification means and derive some first elementary observations.

In the rest of the paper, let  $\mathfrak{A}$  be an  $L$ -algebra and let  $(\mathfrak{A}, \mathfrak{B})$  be a pair of  $L$ -algebras. We shall always write  $\mathfrak{A}$  instead of  $(\mathfrak{A}, \mathfrak{A})$ .

Given an  $L$ -term  $s$ , define

$$\downarrow_{\mathfrak{A}} s := \{s^{\mathfrak{A}}(\mathbf{o}) \mid \mathbf{o} \in A^{r(s)}\}.$$

In case  $a \in \downarrow_{\mathfrak{A}} s$ , we say that  $a$  is an *instance* of  $s$ . Notice that any *distinguished* element  $a \in A$  is an instance of itself:

$$(1) \quad \downarrow_{\mathfrak{A}} a = \{a\}.$$

Now define the *generalization ordering* for two  $L$ -terms  $s$  and  $t$  in  $(\mathfrak{A}, \mathfrak{B})$  by

$$s \sqsubseteq_{(\mathfrak{A}, \mathfrak{B})} t \quad :\Leftrightarrow \quad \downarrow_{\mathfrak{A}} s \subseteq \downarrow_{\mathfrak{A}} t \quad \text{and} \quad \downarrow_{\mathfrak{B}} s \subseteq \downarrow_{\mathfrak{B}} t,$$

and

$$s \equiv_{(\mathfrak{A}, \mathfrak{B})} t \quad :\Leftrightarrow \quad s \sqsubseteq_{(\mathfrak{A}, \mathfrak{B})} t \quad \text{and} \quad t \sqsubseteq_{(\mathfrak{A}, \mathfrak{B})} s.$$

For example, we have  $0x \equiv_{(\mathbb{N}, \cdot, 0)} 0$ . In particular, for any distinguished element  $a \in A$ , we have by (1):

$$a \sqsubseteq_{\mathfrak{A}} s \quad \Leftrightarrow \quad a \in \downarrow_{\mathfrak{A}} s$$

and

$$s \sqsubseteq_{\mathfrak{A}} a \quad \Leftrightarrow \quad s^{\mathfrak{A}} \text{ is a constant function with value } a.$$

<sup>1</sup>We omit constant symbols as we identify constants with 0-ary functions.

For two distinguished elements  $a, b \in A$ , we have

$$a \sqsubseteq_{\mathfrak{A}} b \Leftrightarrow \downarrow_{\mathfrak{A}} a \subseteq \downarrow_{\mathfrak{A}} b \Leftrightarrow \{a\} \subseteq \{b\} \Leftrightarrow a = b.$$

**Remark 1.** Notice that our semantic generalization ordering differs from the usual one in *syntactic* anti-unification where a term  $s$  is said to be *more general than* a term  $t$  iff there is a substitution  $\sigma$  such that  $t = s\sigma$ .

**Fact 2.** The generalization ordering  $\sqsubseteq_{(\mathfrak{A}, \mathfrak{B})}$  is a pre-order between  $L$ -terms, for any pair of  $L$ -algebras  $(\mathfrak{A}, \mathfrak{B})$ , that is, it is reflexive and transitive.

Now we define the dual operation to the instantiation operation from above. Formally, for any term  $t \in T_L(X)$ , define

$$\uparrow_{(\mathfrak{A}, \mathfrak{B})} t := \{s \in T_L(X) \mid t \sqsubseteq_{(\mathfrak{A}, \mathfrak{B})} s\}.$$

In case  $s \in \uparrow_{(\mathfrak{A}, \mathfrak{B})} t$ , we say that  $s$  is a *generalization* of  $t$  in  $(\mathfrak{A}, \mathfrak{B})$ .

Recall that a *pre-filter*  $F$  on a pre-ordered set  $(P, \leq)$  is a subset of  $P$  satisfying:

- (1)  $F$  is non-empty.
- (2)  $F$  is downward directed, that is, for every  $a, b \in F$ , there is some  $c \in F$  such that  $c \leq a, b$ .
- (3)  $F$  is an upper set or upward closed, that is, for every  $a \in F$  and  $b \in P$ , if  $a \leq b$  then  $b \in F$ .

The smallest pre-filter containing an element  $a$  is a *principal pre-filter* and  $a$  is a *principal element* — it is given by

$$\uparrow_{(P, \leq)} a := \{b \in P \mid a \leq b\}.$$

The following observation motivates our notation:

**Fact 3.** The set  $\uparrow_{(\mathfrak{A}, \mathfrak{B})} t$  is the principal pre-filter with respect to the generalization ordering  $\sqsubseteq_{(\mathfrak{A}, \mathfrak{B})}$  generated by  $t$ .

We are now ready to introduce the main notion of the paper:

**Definition 4.** Given two elements  $a, b \in A$ , define

$$a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b := \min_{\sqsubseteq_{(\mathfrak{A}, \mathfrak{B})}}(a \uparrow_{(\mathfrak{A}, \mathfrak{B})} b),$$

with

$$a \uparrow_{(\mathfrak{A}, \mathfrak{B})} b := (\uparrow_{\mathfrak{A}} a) \cap (\uparrow_{\mathfrak{B}} b).$$

Here  $\min_{\sqsubseteq_{(\mathfrak{A}, \mathfrak{B})}}$  is the function computing the *set* of minimal elements with respect to  $\sqsubseteq_{(\mathfrak{A}, \mathfrak{B})}$  (not a single minimal element), which means that in case there are *no* minimal elements, the computed set is empty. That is,  $a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b$  contains the *minimally general generalizations* (or *mggs*) of  $a$  and  $b$  in  $(\mathfrak{A}, \mathfrak{B})$ . In case  $a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b = \{s\}$  contains a single term, we call  $s$  the *least general generalization* of  $a$  and  $b$  in  $(\mathfrak{A}, \mathfrak{B})$ . Notice that in case  $a \in A$  is a distinguished element, we always have

$$a \sqcap_{\mathfrak{A}} a = \{a\}.$$

The following definition is an adaptation of the definition in Cerna (2020, p.135):

**Definition 5.** We say that  $(\mathfrak{A}, \mathfrak{B})$  is:

- *Nullary* iff  $a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b = \emptyset$ , for some  $a \in A$  and  $b \in B$ .
- *Unitary* iff  $|a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b| = 1$ , for all  $a \in A$  and  $b \in B$ .
- *Finitary* iff
  - $|a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b| < \infty$ , for all  $a \in A$  and  $b \in B$ ,

- $|a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b| > 1$ , for some  $a \in A$  and  $b \in B$ .
- *Infinitary* iff  $|a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b| = \infty$ , for some  $a \in A$  and  $b \in B$ .
- *Trivial* iff  $a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b = T_L(X)$ , for all  $a \in A$  and  $b \in B$ .

Notice that in term algebras, syntactic and semantic generalization orderings (and thus anti-unification) coincide:

**Lemma 6.** *For any  $L$ -terms  $s, t \in T_L(X)$ , we have*

$$t \lesssim s \quad \Leftrightarrow \quad t \sqsubseteq_{\mathfrak{T}_L(X)} s.$$

*Proof.* We have the following equivalences:

$$\begin{aligned} t \lesssim s &\Leftrightarrow t = s(\mathbf{o}), \quad \text{for some } \mathbf{o} \in T_L(X)^{r(s)} \\ &\Leftrightarrow \downarrow_{\mathfrak{T}_L(X)} t \subseteq \downarrow_{\mathfrak{T}_L(X)} s \\ &\Leftrightarrow t \sqsubseteq_{\mathfrak{T}_L(X)} s. \end{aligned}$$

□

**Fact 7.** *Term algebras are unitary.*

*Proof.* A direct consequence of Lemma 6 and the well-known fact that term algebras are unitary by Plotkin's (1970) and Reynolds's (1970) classical results, and their original algorithms (plus Huet's (1976) simple algorithm) can be used to compute the least general generalization of two terms. □

**Fact 8.**  $\uparrow_{\mathfrak{A}} a = \{x\}$  *implies*  $a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b = \{x\}$ , *for all*  $b \in B$ .

**Fact 9.**  $a \uparrow_{(\mathfrak{A}, \mathfrak{B})} b = \{x\}$  *implies*  $a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b = a \uparrow_{(\mathfrak{A}, \mathfrak{B})} b$ .

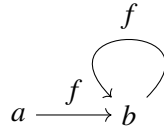
**Fact 10.** *In any pair of algebras  $(\mathfrak{A}, \mathfrak{B})$  so that  $A = \{a\}$  and  $B = \{b\}$ , we have  $a \sqcap_{(\mathfrak{A}, \mathfrak{B})} b = a \uparrow_{(\mathfrak{A}, \mathfrak{B})} b$ .*

*Proof.* Every generalization  $s \in a \uparrow_{(\mathfrak{A}, \mathfrak{B})} b$  has to satisfy  $\downarrow_{\mathfrak{A}} s = \{a\}$  and  $\downarrow_{\mathfrak{B}} s = \{b\}$ , which means that there cannot be a generalization  $t$  such that  $\downarrow_{(\mathfrak{A}, \mathfrak{B})} t \subsetneq \downarrow_{(\mathfrak{A}, \mathfrak{B})} s$ . □

**Fact 11.** *The algebra  $(A)$  consisting only of its universe is unitary and trivial.*

*Proof.* A direct consequence of  $a \sqcap_{(A)} b = \{x\}$ , for all  $a, b \in A$  (notice that  $a \sqcap_{(A)} a \neq \{a\}$  since  $a$  is not a distinguished element of  $(A)$  and thus cannot be used to form terms). □

**Example 12.** In the monounary algebra  $\mathfrak{A} = (\{a, b\}, f)$  given by



we have

$$\uparrow_{\mathfrak{A}} a = \{x\}$$

and thus by Fact 8 we have

$$a \sqcap_{\mathfrak{A}} b = \{x\}.$$

This shows that  $\mathfrak{A}$  is unitary.

## 3. MONOUNARY ALGEBRAS

In the rest of this section, let  $\mathfrak{A} = (A, f)$  be a monounary algebra with  $f : A \rightarrow A$  the only unary function. We define

$$f^k(x) \leq f^\ell(x) \Leftrightarrow k \leq \ell.$$

We have

$$k \leq \ell \text{ implies } f^\ell(x) \sqsubseteq f^k(x).$$

**Example 13.** Consider the monounary algebra  $\mathfrak{A} = (\{a\}, f)$  given by



We have

$$\uparrow_{\mathfrak{A}} a = \{f^k(x) \mid k \geq 0\}.$$

Since

$$\downarrow_{\mathfrak{A}} f^k(x) = \downarrow_{\mathfrak{A}} f^\ell(x) = \{a\}, \text{ for all } k, \ell \geq 0,$$

we have

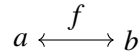
$$f^k(x) \equiv_{\mathfrak{A}} f^\ell(x), \text{ for all } k, \ell \geq 0,$$

and thus

$$a \sqcap_{\mathfrak{A}} a = \{f^k(x) \mid k \geq 0\}.$$

This shows that  $\mathfrak{A}$  is infinitary and trivial.

Similarly, the monounary algebra  $\mathfrak{B} = (\{a, b\}, f)$  given by



is infinitary and trivial since

$$\uparrow_{\mathfrak{B}} a = \{f^k(x) \mid k \geq 0\} = \uparrow_{\mathfrak{B}} b$$

and

$$\downarrow_{\mathfrak{B}} f^k(x) = \downarrow_{\mathfrak{B}} f^\ell(x) = \{a, b\}, \text{ for all } k, \ell \geq 0,$$

implies

$$f^k(x) \equiv_{\mathfrak{B}} f^\ell(x), \text{ for all } k, \ell \geq 0,$$

and thus

$$a \sqcap_{\mathfrak{B}} a = b \sqcap_{\mathfrak{B}} b = a \sqcap_{\mathfrak{B}} b = \{f^k(x) \mid k \geq 0\}.$$

We define

$$m(a) := \begin{cases} \max \{k \mid f^k(x) \in \uparrow_{\mathfrak{A}} a\} & \text{if the maximum exists,} \\ \infty & \text{otherwise,} \end{cases}$$

and

$$m(a, b) := \min(m(a), m(b)) \in \mathbb{N} \cup \{\infty\}.$$

**Theorem 14.** Let  $\mathfrak{A} := (A, f)$  be a monounary algebra. For any  $a, b \in A$ , we have

$$a \sqcap_{\mathfrak{A}} b = \begin{cases} \{f^{m(a,b)}(x)\} & m(a, b) < \infty \\ \emptyset & \text{otherwise.} \end{cases}$$

*Proof.* Suppose  $m(a, b) < \infty$ , which means that either  $m(a) < \infty$  or  $m(b) < \infty$ , that is, there is some maximal  $k$  such that  $f^k(x) \in \uparrow_{\mathfrak{A}} a$  or  $f^k(x) \in \uparrow_{\mathfrak{A}} b$ . We then have

$$f^{k+\ell}(x) \notin (\uparrow_{\mathfrak{A}} a) \cap (\uparrow_{\mathfrak{A}} b) = a \uparrow b, \quad \text{for all } \ell \geq 1,$$

which implies  $a \sqcap_{\mathfrak{A}} b = \{f^k(x)\}$  — notice that  $k = m(a, b)$  by definition.

Now suppose  $m(a, b) = \infty$ , which means that  $m(a) = m(b) = \infty$ . In that case, we clearly have  $\max_{\leq}(a \uparrow_{\mathfrak{A}} b) = \emptyset$ .  $\square$

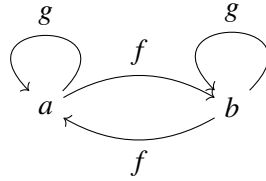
Let  $(\mathbb{N}, S)$  be the infinite monounary algebra where  $S(a) := a + 1$  denotes the *successor function*.

**Corollary 15.** For any  $a, b \in \mathbb{N}$ , we have  $a \sqcap_{(\mathbb{N}, S)} b = \{S^{\min(a,b)}(x)\}$ . This means that  $(\mathbb{N}, S)$  is unitary.

#### 4. FINITE UNARY ALGEBRAS

Let  $\mathfrak{A} = (A, F := \{f_i \mid i \in I\})$  be a finite unary algebra, that is,  $A$  and  $I$  are finite. In this section, we summarize some first observations regarding semantic anti-unification in  $\mathfrak{A}$ . We assume the reader to be familiar with the basics of formal language and automata theory as presented for instance in Sipser (2006).

In Antić (2023a), we have seen that in every *finite* unary algebra  $\mathfrak{A}$ , the sets  $\uparrow_{\mathfrak{A}} a$  and  $a \uparrow_{\mathfrak{A}} b$  can be computed by some finite automata  $\mathcal{A}(a)$  and  $\mathcal{A}(a, b)$ , respectively, which we will repeat here. For example, consider the unary algebra  $\mathfrak{A}$  given by



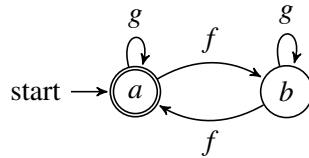
consisting of two unary functions  $f$  and  $g$ . We want to compute the set of generalizations  $\uparrow_{\mathfrak{A}} a$ . Observe that

$$\uparrow_{\mathfrak{A}} a = \{s(x) \in T_L(\{x\}) \mid a = s^{\mathfrak{A}}(a)\} \cup \{t(x) \in T_L(\{x\}) \mid a = t^{\mathfrak{A}}(b)\}.$$

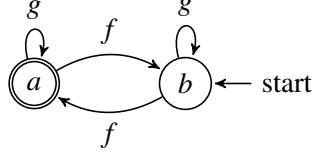
Moreover, we can identify each term in  $\mathfrak{A}$  with a word over the alphabet  $\{f, g, x\}$ : for example, the term  $f(g(f(x)))$  can be identified with the word  $fgfx$ . This means that we can identify the set  $\uparrow_{\mathfrak{A}} a$  with

$$\begin{aligned} \uparrow_{\mathfrak{A}} a = & \{wx \in \{f, g\}^* \{x\} \mid \text{if we start in } a \text{ and follow the word } w \text{ in } \mathfrak{A} \text{ we end up in } a\} \\ & \cup \{ux \in \{f, g\}^* \{x\} \mid \text{if we start in } b \text{ and follow the word } v \text{ in } \mathfrak{A} \text{ we end up in } a\}. \end{aligned}$$

Now define the automaton  $\mathcal{A}_{a \rightarrow a}$  by



and the automaton  $\mathcal{A}_{b \rightarrow a}$  by



Hence,

$$\uparrow_{\mathfrak{A}} a = \|\mathcal{A}_{a \rightarrow a}\|\{x\} \cup \|\mathcal{A}_{b \rightarrow a}\|\{x\},$$

where  $\|\mathcal{A}\|$  denotes the *behavior* of  $\mathcal{A}$ .

We can easily generalize the above reasoning, that is, given any *finite*<sup>2</sup> unary algebra  $\mathfrak{A} = (A, f_1, \dots, f_n)$ , we have

$$(2) \quad \uparrow_{\mathfrak{A}} a = \bigcup_{b \in A} (\|\mathcal{A}_{b \rightarrow a}\|\{x\}),$$

where  $\mathcal{A}_{b \rightarrow a}$  is the automaton induced by the functions  $f_1, \dots, f_n$  of the algebra  $\mathfrak{A}$  with start state  $b$  and final state  $a$  as above. Hence, computing

$$a \uparrow_{\mathfrak{A}} b = (\uparrow_{\mathfrak{A}} a) \cap (\uparrow_{\mathfrak{A}} b)$$

amounts to computing the intersection of two regular languages by applying standard techniques from automata theory.

We have

$$\uparrow_{\mathfrak{A}} a = \bigcup_{b \in A} \{wx \mid a = wb\} = \bigcup_{b \in A} \{w \in F^* \mid a = wb\}x.$$

That is,  $\uparrow_{\mathfrak{A}} a$  is a formal language over  $F$  concatenated with the variable  $x$  (and thus, formally, a formal language over  $F \cup \{x\}$  of a very specific form). Moreover, we have

$$\downarrow_{\mathfrak{A}} wx = \{wa \mid a \in A\}$$

and thus

$$wx \sqsubseteq_{\mathfrak{A}} ux \quad \Leftrightarrow \quad \{wa \mid a \in A\} \subseteq \{ua \mid a \in A\}.$$

Hence, to decide whether  $wx$  is  $\sqsubseteq_{\mathfrak{A}}$ -minimal means to decide whether there is some  $ux$  such that

$$\{ua \mid a \in A\} \subsetneq \{wa \mid a \in A\}.$$

In a finite algebra, this is clearly a computable relation (see Problem 4).

## 5. THE $(k, \ell)$ -FRAGMENTS

Since computing the set of *all* generalizations is rather difficult in general, it is reasonable to study fragments of the framework. For this, we introduce in this section the  $(k, \ell)$ -fragments. Formally, let  $X_k := \{x_1, \dots, x_k\}$ , for some  $k, \ell \in \mathbb{N} \cup \{\infty\}$  so that  $X_\infty = X$ . Define

$$\uparrow_{\mathfrak{A}}^{(k, \ell)} a := (\uparrow_{\mathfrak{A}} a) \cap \{s(x_1, \dots, x_k) \in T_L(X_k) \mid \text{each of the } k \text{ variables in } X_k \text{ occurs at most } \ell \text{ times in } s\}.$$

We write  $k$  instead of  $(k, \infty)$  so that

$$\uparrow_{\mathfrak{A}}^k a = (\uparrow_{\mathfrak{A}} a) \cap T_L(X_k).$$

<sup>2</sup>Finiteness is required since regular languages are not closed under *infinite* union.



**Example 16.** The simplest fragment — namely, the (1, 1)-fragment — contains only monolinear generalizations containing exactly one occurrence of a single variable  $x$ . We denote the monolinear generalization and instantiation operations in  $\mathfrak{A}$  by  $\uparrow_{\mathfrak{A}}^m$  and  $\downarrow_{\mathfrak{A},m}$ , respectively, and the so-obtained anti-unification relation by  $\sqcap_{\mathfrak{A},m}$ .

We now compute monolinear anti-unification in the set domain.

First, we work in  $(2^U, \cup, 2^U)$  and omit the explicit reference to the algebra. We have

$$A \uparrow^m B = \{Z \cup C \mid \emptyset \subseteq C \subseteq A \cap B\} \cup \{A \mid A = B\}$$

and

$$\downarrow_m(Z \cup C) = [C, U].$$

This implies

$$Z \cup C \subseteq Z \cup D \Leftrightarrow [C, U] \subseteq [D, U] \Leftrightarrow D \subseteq C.$$

Hence

$$A \sqcap_m B = \begin{cases} \{Z \cup (A \cap B)\} & A \neq B \\ \{A\} & A = B. \end{cases}$$

Second, we work in  $(2^U, \cap, 2^U)$ . We have

$$A \uparrow^m B = \{Z \cap C \mid A, B \subseteq C\} \cup \{A \mid A = B\}$$

and

$$\downarrow_m(Z \cap C) = [\emptyset, C].$$

This implies

$$Z \cap C \subseteq Z \cap D \Leftrightarrow [\emptyset, C] \subseteq [\emptyset, D] \Leftrightarrow C \subseteq D.$$

Hence

$$A \sqcap_m B = \begin{cases} \{Z \cap (A \cup B)\} & A \neq B \\ \{A\} & A = B. \end{cases}$$

Third, we work in  $(2^U, c, 2^U)$ . We have

$$A \uparrow^m B = \{Z, Z^c\} \cup \{A \mid A = B\}$$

and

$$\downarrow_m Z^c = 2^U = \downarrow Z.$$

This implies

$$Z^c \equiv Z.$$

Hence

$$A \sqcap_m B = \begin{cases} \{Z, Z^c\} & A \neq B \\ \{A\} & A = B. \end{cases}$$

**Remark 17.** In the full set algebra  $(2^U, \cup, \cap, c, 2^U)$ , it already appears challenging to compute  $A \sqcap_m B$ , for given sets  $A, B \in 2^U$  (see Problem 6).

## 6. FINITE ALGEBRAS

In this section, let  $\mathfrak{A}$  be a *finite algebra* which means that its underlying universe  $A$  is a finite set. For  $k \geq 1$ , let  $X_k := \{x_1, \dots, x_k\}$ .

Recall that a (*frontier-to-root*) *tree automaton* (cf. Gécseg & Steinby, 2015; Comon et al., 2008)

$$\mathcal{T}_{k,\alpha,A'}(\mathfrak{A}) := (\mathfrak{A}, L, X_k, \alpha, A')$$

consists of

- a *finite L-algebra*  $\mathfrak{A}$ ,
- an *initial assignment*  $\alpha : X_k \rightarrow A$ , and
- a set  $A' \subseteq A$  of *final states*.

The *regular tree language* recognized by  $\mathcal{T}_{k,\alpha,A'}(\mathfrak{A})$  is given by

$$\|\mathcal{T}_{k,\alpha,A'}(\mathfrak{A})\| := \{s \in T_L(X_k) \mid s^{\mathfrak{A}}(\alpha) \in A'\}.$$

We have

$$\uparrow_{\mathfrak{A}}^k a = \bigcup_{\alpha \in A^{X_k}} \|\mathcal{T}_{k,\alpha,\{a\}}(\mathfrak{A})\|.$$

Since  $A^{X_k}$  is a finite set and tree automata are closed under finite union, the set  $\uparrow_{\mathfrak{A}}^k a$  is a regular tree language. Moreover, since tree automata are closed under finite intersection, there is some tree automaton  $\mathfrak{T}_{k,a,b}(\mathfrak{A})$  such that

$$a \uparrow_{\mathfrak{A}}^k b = (\uparrow_{\mathfrak{A}}^k a) \cap (\uparrow_{\mathfrak{A}}^k b) = \|\mathfrak{T}_{k,a,b}(\mathfrak{A})\|.$$

Since the underlying algebra is finite, checking the  $b$ - $\sqsubseteq$ -minimality of  $a \uparrow_{\mathfrak{A}}^k b$  can be done in time linear to the size of the algebra. In total, we have thus derived an algorithm for the computation of  $a \sqcap_{\mathfrak{A},k} b^3$  in finite algebras (see Problem 5).

## 7. PROBLEMS

This section lists problems which remained unsolved in this paper and appear to be interesting lines of future research.

**Problem 1.** The main open problem is to provide an algorithm for the computation of  $a \sqcap_{\mathfrak{A}} b$  in an arbitrary general algebra  $\mathfrak{A}$  in case the problem is decidable, or to prove that the problem is undecidable.

**Problem 2.** Study the relationship between semantic anti-unification as introduced in this paper and anti-unification modulo equational theory (Heinz, 1995; Burghardt, 2005).

**Problem 3.** Study semantic anti-unification in finitely representable infinite algebras, most importantly in automatic structures (Blumensath & Grädel, 2000, 2004).

**Problem 4.** Give an explicit algorithm for the computation of mggs in finite unary algebras.

**Problem 5.** Implement the algorithm of §6 for the computation of  $k$ -mggs in finite algebras.

**Problem 6.** Characterize monilinear set anti-unification in the full set algebra (see Example 16).

---

<sup>3</sup>By  $a \sqcap_{\mathfrak{A},k} b$  we mean, of course, the set of minimally general  $k$ -generalizations of  $a$  and  $b$  in  $\mathfrak{A}$ .

## 8. CONCLUSION

Motivated by recent applications to analogical reasoning, this paper initiated the study of *semantic anti-unification* in general algebras.

## CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

## DATA AVAILABILITY STATEMENT

The manuscript has no data associated.

## REFERENCES

- Antić, C. (2022). Analogical proportions. *Annals of Mathematics and Artificial Intelligence*, 90(6), 595–644.
- Antić, C. (2023a). Generalization-based similarity. submitted to *Journal of Logic and Computation*, <https://arxiv.org/pdf/2302.10096.pdf>.
- Antić, C. (2023b). Logic program proportions. *Annals of Mathematics and Artificial Intelligence*, accepted. <https://arxiv.org/pdf/1809.09938.pdf>.
- Baader, F., & Nipkow, T. (1998). *Term Rewriting and All That*. Cambridge University Press, Cambridge UK.
- Baader, F., & Snyder, W. (2001). Unification theory. In *Handbook of Automated Reasoning*. Elsevier.
- Barwell, A. D., Brown, C., & Hammond, K. (2018). Finding parallel functional pearls: Automatic parallel recursion scheme detection in Haskell functions via anti-unification. *Future Generation Computer Systems*, 79(2), 669–686.
- Blumensath, A., & Grädel, E. (2000). Automatic structures. In *LICS 2000*, pp. 51–62. IEEE Computer Society.
- Blumensath, A., & Grädel, E. (2004). Finite presentations of infinite structures: Automata and interpretations. *Theory of Computing Systems*, 37, 641–674.
- Burghardt, J. (2005). *E*-generalization using grammars. *Artificial Intelligence*, 165(1), 1–35.
- Burris, S., & Sankappanavar, H. (2000). *A Course in Universal Algebra*. <http://www.math.hawaii.edu/~ralph/Classes/619/univ-algebra.pdf>.
- Cao, D., Kunkel, R., Nandi, C., Willsey, M., Tatlock, Z., & Polikarpova, N. (2023). babble: Learning better abstractions with e-graphs and anti-unification. *Proceedings of the ACM on Programming Languages*, 7(POPL), 396–424.
- Cerna, D. M. (2020). Anti-unification and the theory of semirings. *Theoretical Computer Science*, 848, 133–139.
- Cerna, D. M., & Kutsia, T. (2023). Anti-unification and generalization: a survey. <https://arxiv.org/pdf/2302.00277.pdf>.
- Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Tison, S., & Tommasi, M. (2008). *Tree Automata Techniques and Applications*. <https://hal.inria.fr/hal-03367725>.
- Cropper, A. (2022). Inductive logic programming at 30. *Machine Learning*, 111(1), 147–172.
- Cropper, A., & Morel, R. (2021). Learning programs by learning from failures. *Machine Learning*, 110(4), 801–856.
- de Sousa, R. R., Soares, G., Gheyi, R., Barik, T., & D’Antoni, L. (2021). Learning quick fixes from code repositories. In *SBES 2021*, pp. 74–83.

- Gécseg, F., & Steinby, M. (2015). *Tree Automata* (2 edition). <https://arxiv.org/pdf/1509.06233.pdf>.
- Gulwani, S. (2016). Programming by examples — and its applications in data wrangling. In *NATO Science for Peace and Security Series*, Vol. 45, pp. 137–158. IOS Press.
- Heinz, B. (1995). *Anti-Unifikation modulo Gleichungstheorie und deren Anwendung zur Lemmagerierung*. Ph.D. thesis, TU Berlin.
- Huet, G. (1976). *Résolution d'équations dans des langages d'ordre 1,2,..., $\omega$* . Ph.D. thesis, Université Paris VII.
- Muggleton, S. (1991). Inductive logic programming. *New Generation Computing*, 8(4), 295–318.
- Plotkin, G. D. (1970). A note on inductive generalization. *Machine Intelligence*, 5, 153–163.
- Reynolds, J. C. (1970). Transformational systems and the algebraic structure of atomic formulas. *Machine Intelligence*, 5(1), 135–151.
- Schmidt, M., Krumnack, U., Gust, H., & Kühnberger, K.-U. (2014). Heuristic-driven theory projection: an overview. In Prade, H., & Richard, G. (Eds.), *Computational Approaches to Analogical Reasoning: Current Trends*, Vol. 548 of *Studies in Computational Intelligence*, pp. 163–194. Springer-Verlag, Berlin/Heidelberg.
- Sipser, M. (2006). *Introduction to the Theory of Computation* (2 edition). Thomson Course Technology, Boston.
- Vanhoof, W., & Yernaux, G. (2019). Generalization-driven semantic clone detection in CLP. In Gabrielli, M. (Ed.), *LOPSTR 2019*, pp. 228–242. Springer-Verlag.
- Weller, S., & Schmid, U. (2007). Solving proportional analogies using E-generalization. In Freksa, C., Kohlhase, M., & Schmill, K. (Eds.), *KI 2006, LNAI 4314*, pp. 64–75. Springer-Verlag.