When First-order Unification Calls Itself

David M. Cerna

July 18^{th} 2021

JOHANNES KEPLER
UNIVERSITY LINZ

Czech Academy
of Sciences

The Origin of the Problem

- Consider schematically defined clause sets such as

$$P(x,0),\ P(x,1),\ \cdots,\ P(x,n)$$

$$\neg P(x,0),\ \neg P(y,0),\ \neg s(y) \leq x$$

$$\neg P(x,1),\ \neg P(y,1),\ \neg s(y) \leq x$$

$$\vdots$$

$$\neg P(x,n),\ \neg P(y,n),\ \neg s(y) \leq x$$

$$m(x,y) \leq z, x \leq z$$

$$m(x,y) \leq z, y \leq z$$

- Each instance can be handled individually.
- We want to construct a finite representation of all instance resolution refutations of such clause sets.

The Origin of the Problem

- The formal representation of such refutations is discussed in

  "Schematic Refutations of Formula Schemata", David M. Cerna, Alexander Leitsch, and Anela Lolic

- The goal of that work is inductive proof transformation and extraction of Schematic Herbrand Sequents.
- The bulk of the information is stored in a schematic unifier extracted from the formal representation of the refutation.
- Such unifiers need to be constructed during the formalization process.
- In this talk, we discuss a small step towards algorithmically constructing schematic unifiers.

Schematic Unification Problems - Extension

- Consider the following unification problem:

$$h(h(x_2, h(x_1, x_2)), \hat{a}) \stackrel{?}{=} h(\hat{b}, h(y_1, h(y_2, y_1)))$$

- Here $\hat{a}$ and $\hat{b}$ denote recursion variables and are associated with so called extension operators. $\mathrm{ex}_{\hat{a}}^{s}(\cdot)$ and $\mathrm{ex}_{\hat{b}}^{s'}(\cdot)$ where s and s' are some terms.
- The convention taken in this work is that $h(h(x_2, h(x_1, x_2)), \hat{a})$ is an extendable term and

$$\mathrm{ex}_{\hat{a}}^{s}(\hat{a}) = h(h(x_2, h(x_1, x_2)), \hat{a}) = s$$

- Each extendable term has an associated extension operator and a unique recursion variable.

Schematic Unification Problems - Shifting

- Now consider the variables

$$h(h(x_2, h(x_1, x_2)), \hat{a}) \stackrel{?}{=} h(\hat{b}, h(y_1, h(y_2, y_1)))$$

- Variables used in our schematic unification problem belong to variable classes.
- A variable class is a pair $(Z, <)$ where $Z \subset \mathcal{V}$ and $<$ is a strict well-founded total linear order of Z. Each class has a successor function $Suc^Z_<(\cdot)$ with following properties over Z:
  - $x < Suc^Z_<(x)$
  - $Suc^Z_<(x) < Suc^Z_<(y) \Longrightarrow x < y$
- We define a shift operator $s^Z(\cdot)$ which replaces all variables in Z by their successors with respect to the variable class $(Z, <)$.

Application of the Operators

The shift operator is applied recursively to terms as follows:

- $\mathsf{s}^Z(f(t_1, \cdots, t_n)) = f(\mathsf{s}^Z(t_1), \cdots, \mathsf{s}^Z(t_n))$
- for $z \in V_{\mathbb{N}}^{x}$, s.t. $x \in X$, $\mathsf{s}^Z(z) = Suc(z)$
- for $\hat{a} \in S$, $\mathsf{s}^Z(\hat{a}) = \hat{a}$

The extension operator is applied recursively to terms as follows:

- $\mathrm{ex}_{\hat{a}}^{s}(f(t_1, \cdots, t_n)) = f(\mathrm{ex}_{\hat{a}}^{s}(t_1), \cdots, \mathrm{ex}_{\hat{a}}^{s}(t_n))$
- for $z \in Z$, $\mathrm{ex}_{\hat{a}}^{s}(z) = z$
- $\mathrm{ex}_{\hat{a}}^{s}(\hat{a}) = s$
- for $\hat{b} \in \mathcal{R}$ such that $\hat{a} \neq \hat{b}$, $\mathrm{ex}_{\hat{a}}^{s}(\hat{b}) = \hat{b}$

Both may be applied to unifiers, but the extension operator only changes the right-hand side term.

- $\mathrm{ex}_{\hat{a}}^{s}(\{x \to t\}) = \{x \to \mathrm{ex}_{\hat{a}}^{s}(t)\}$

Schematic Unification Problems - the Sequence

- Let $s = h(h(x_2, h(x_1, x_2)), \hat{a})$ and $t = h(\hat{b}, h(y_1, h(y_2, y_1)))$. Repeated application of extend and shift results in the following sequence:

$$s \stackrel{?}{=} t$$

$$\mathsf{ex}^{s}_{\hat{a}}(\mathsf{s}^{X}(s)) \stackrel{?}{=} \mathsf{ex}^{t}_{\hat{b}}(\mathsf{s}^{Y}(t))$$

$$\mathsf{ex}^{s}_{\hat{a}}(\mathsf{s}^{X}(\cdots \mathsf{ex}^{s}_{\hat{a}}(\mathsf{s}^{X}(s))\cdots)) \stackrel{?}{=} \mathsf{ex}^{t}_{\hat{b}}(\mathsf{s}^{Y}(\cdots \mathsf{ex}^{t}_{\hat{b}}(\mathsf{s}^{Y}(t))\cdots))$$

- Loop unification, denoted by $s \stackrel{\circlearrowleft}{=} t$, requires finding a solution to all instance unification problems.
- We focus on left semiloop unification, i.e. only the left side is extendable.
  - The right side is the same in every instance.
  - The right semiloop unification is symmetrically defined.
- Loops are denoted by $\langle s, t \rangle$ ($\langle s, t |$ for left semiloops).

Extensions of Loops and Semiloops

- Instance problems are derived from extensions of loops.
- The *n-extension* of the loop $\langle s, t\rangle$, denoted by $\langle s, t\rangle_n$ is defined recursively as follows:
  - $\langle s, t\rangle_0 = \langle \hat{a}, \hat{b}\rangle$
  - $\langle s, t\rangle_{n+1} = \langle \mathrm{ex}^s_{\hat{a}}(\mathsf{s}(s')), \mathrm{ex}^t_{\hat{b}}(\mathsf{s}(t'))\rangle$ where $\langle s, t\rangle_n = \langle s', t'\rangle$.
- Semiloops, only the left side will be shifted and extended.
- $\langle s, t|_{n+1} = \langle \mathrm{ex}^s_{\hat{a}}(\mathsf{s}(s')), t|$ where $\langle s, t|_n = \langle s', t|$.
- When we say $s \overset{\circlearrowleft}{=} t$ is loop unifiable, we mean for all n, such that $\langle s, t|_n = \langle s', t'|$, $s' \overset{?}{=} t'$ is unifiable.

Relationship to Existing Work

- Notice that this construction has similarities with both Narrowing methods and Primal grammars.
- The following example illustrates the differences:

$$h(h(x_2, h(x_1, x_2)), \hat{a}) \overset{\circlearrowleft}{=} h(y_1, y_1)$$

Solving the above problem requires solving both:

$$h(h(x_2, h(x_1, x_2)), \hat{a}) \overset{?}{=} h(y_1, y_1)$$

$$h(h(x_3, h(x_2, x_3)), h(x_2, h(x_1, x_2))) \overset{?}{=} h(y_1, y_1)$$

- Notice that the second problem in the sequence introduces fresh variables as well as "used" variables.
- While this construction can be expressed using the above mentioned methods, few if any existing results are applicable.

Types of (Semi)loop Unifiability

- There are two ways $\langle s, t|_n$ can unify depending on whether $\hat{a}$ occurs in the unifier σ derived from the solved form.
  (1) $\hat{a}$ occurs in the domain of σ.
  (2) $\hat{a}$ does not occur in the domain of σ
- When there is a choice, we assume a preference for $\hat{a}$ occurring in the domain, i.e. $\hat{a} \stackrel{?}{=} x_i$.
- We refer to extensions of type (1) as extendably unifiable.
- The intuition being that if $\hat{a} \stackrel{?}{=} t^*$ occurs in the solved form, the unification problem may be extended.
- If every extension of a loop $\langle s, t|$ is extendably unifiable then we say $\langle s, t|$ is infinitely loop unifiable.
- Otherwise we say $\langle s, t|$ is finitely loop unifiable.

Halting and Infinite Loop Unifiable

- When an extension $\langle s, t|_k$ is Type (2) unifiable, a unifier can be built for all extensions $\langle s, t|_r$, for $r \geq k$.
- Concerning extensions $\langle s, t|_k$ which are Type (1) unifiable it is more complex.
- The title of this talk comes from the Type (1) unifiability problem:

  function $\textsc{LoopUnif}(\langle s, t|, k)$
      if $\langle s, t|_k$ is extendably unifiable **then**
          $\textsc{LoopUnif}(\langle s, t|, k+1)$
      end if
  end function

- It is currently open if a decision procedure exists concerning the halting of the above procedure.
- We will present a sufficient condition for infinite/finite loop unifiability of semiloops.

Example Finite Loop Unifiability

- Let us consider the semiloop $\langle s, t| =$

$$\langle h(h(h(x_2, h(x_1, x_1)), x_3), \hat{a}),\ \ h(h(y_4, y_3), h(y_1, y_2))|$$

- The first extension is unifiable by

$$\{y_3 \mapsto x_3\ ,\ \ y_4 \mapsto h(x_2, h(x_1, x_1))\ ,\ \ \hat{a} \mapsto h(y_1, y_2)\}.$$

- Thus, $\langle s, t|_1$ is extendably unifiable. What about the $\langle s, t|_2$?

$$\langle h(h(h(x_2, x_1), h(x_2, x_3)), h(h(h(x_2, x_1), h(x_2, x_3)), \hat{a}))\ ,\ \ t|$$

- It is unifiable by $\{y_3 \mapsto x_4,\ \ y_4 \mapsto h(x_3, h(x_2, x_2)),$

$$y_1 \mapsto h(h(x_2, h(x_1, x_1)), x_3),\ \ y_2 \mapsto \hat{a}\}$$

- This extension is not extendably unifiable.
- It is trivial to show finite unifiability of $\langle s, t|$.

Example Not Loop Unifiable

- Let us consider the semiloop $\langle s, t| =$

$$\langle h(h(h(x_2, x_1), h(x_2, x_3)), \hat{a}) \ , \ h(h(y_3, y_1), h(y_4, y_4))|$$

- The first extension is unifiable by

$$\{y_3 \mapsto h(x_2, x_1) \ , \ y_1 \mapsto h(x_2, x_3) \ , \ \hat{a} \mapsto h(y_4, y_4)\}.$$

- Thus, $\langle s, t|_1$ is extendably unifiable. What about $\langle s, t|_2$?

$$\langle h(h(h(x_3, x_2), h(x_3, x_4)), h(h(h(x_2, x_1), h(x_2, x_3)), \hat{a})) \ , \ t|$$

- It is unifiable by

$$\begin{array}{l}\{y_3 \mapsto h(x_3, x_2), \ y_4 \mapsto h(h(x_2, x_1), h(x_2, x_3)), \\ y_1 \mapsto h(x_3, x_4), \ \hat{a} \mapsto h(h(x_2, x_1), h(x_2, x_3))\}.\end{array}$$

- Thus $\langle s, t|_2$ is also extendably unifiable.

Example Not Loop Unifiable

- What about $\langle s, t|_3$?

$$\langle h(h(h(h(x_4, x_3), h(x_4, x_5)), h(h(h(x_3, x_2), h(x_3, x_4)),$$

$$h(h(h(x_2, x_1), h(x_2, x_3)), \hat{a}))) \ , \ t|$$

- Notice the solved form contains an occurrence check:

$$\{y_3 \stackrel{?}{=} h(x_4, x_3), \ y_4 \stackrel{?}{=} h(h(x_3, x_2), h(x_3, x_4)),$$
$$y_1 \stackrel{?}{=} h(x_4, x_5), \ \hat{a} \stackrel{?}{=} h(x_3, x_4), \ x_3 \stackrel{?}{=} h(x_2, x_1),$$
$$x_2 \stackrel{?}{=} h(x_2,, x_3)\}.$$

Example Infinite Loop Unifiable

- One of the simplest infinite loop unifiable semiloops is

$$\langle s, t| = \langle h(h(x_1, x_1), \hat{a})\ ,\ h(y_1, y_1)|$$

- The solved form of every extension contains $\hat{a} \stackrel{?}{=} h(x_1, x_1)$.
- A more complex example would be $\langle s, t| =$

$$\langle h(\hat{a}, h(h(h(x_1, x_1), x_1), x_1)),\ h(h(h(h(y_1, y_1), y_1), y_1), y_1)|$$

- Using the following abbreviation:

$$t(n) = h(h(h(x_{n+1}, x_{n+1}), x_{n+1}), x_{n+1})$$

- We can categorize the unifiers of each extension.

Example Infinite Loop Unifiable

- The solved form of $\langle s, t|_{3n}$ will contain

$$\hat{a} \stackrel{?}{=} h(h(t(1), t(1)), t(1)),$$

- The solved form of $\langle s, t|_{3n+1}$ will contain

$$\hat{a} \stackrel{?}{=} h(h(h(h(x_2, x_2), x_2), x_2), h(h(h(x_2, x_2), x_2), x_2)),$$

- The solved form of $\langle s, t|_{3n+2}$ will contain

$$\hat{a} \stackrel{?}{=} h(h(h(x_2, x_2), x_2), x_2).$$

- This pattern repeats for all m-extensions where $m > 2$.

Sufficient Condition for Finite Unifiability

- It is not enough to find a non-extendably unifiable extension.
- In addition, every variable of the unifier must be large enough not to cause occurrence checks.

Theorem

Let $k > 0$. Then if $\langle s, t|_k = \langle s', t|$ is unifiable by σ and the following conditions hold:

(1) $\hat{a} \notin Dom(\sigma)$

(2) *for all* $z \in Dom(\sigma)$ *s.t.* $\hat{a} \in var(z\sigma)$, $|\mathrm{s}(z)| > m$, *where* $m = \max_{z \in var(s\theta)} |j|$ *and* $\theta = \mathrm{s}(\sigma)$.

Then for all $j \geq k$, $\langle s, t|_j$ *is unifiable and* $s \stackrel{\circlearrowleft}{=} t$ *is finitely unifiable.*

- The following example illustrates why (2) is necessary.
- Note, $|x_i| = i$.

Finite Loop Unifiability Fails for Small variables

- Consider the following example:

$$\langle s, t| = \langle h(x_2, h(x_4, \hat{a})), h(y_1, y_1)|$$

- The unifier of $\langle s, t|_1$ is as follows:

$$\{y_1 \mapsto h(x_4, \hat{a}) \ , \ x_2 \mapsto h(x_4, \hat{a})\}$$

- We can generate the unifier for $\langle s, t|_2$ from the above unifier:

$$\{y_1 \mapsto h(x_5, h(x_2, h(x_4, \hat{a}))), \ x_3 \mapsto h(x_5, h(x_2, h(x_4, \hat{a})))\}$$

- However, generating the unifier for $\langle s, t|_3$ this way fails:

$$\begin{array}{l}\{y_1 \mapsto h(x_6, h(x_3, h(x_5, h(x_2, h(x_4, \hat{a}))))), \\ {\color{red}x_4} \mapsto h({\color{blue}x_6}, h(x_3, h({\color{blue}x_5}, h(x_2, h({\color{red}x_4}, \hat{a})))))\}\end{array}$$

- Extending the unifier results in an occurrence check.

Decomposing Unifiers

- Given enough information about the extensions of $\langle s, t|$ one can decomposed the unifier of $\langle s, t|_k$.

Lemma

Let $k > 1$. If $\langle s, t|_1$, $\langle s, t|_k$, and $\langle s, t|_{k+1}$ are extendably unifiable by σ_1, σ_k, σ_{k+1}, then $\sigma_{k+1} = sh^k(\theta)\sigma$ where:

- $sh^k(\cdot) = \overbrace{\mathsf{s}(\cdots \mathsf{s}(\cdot)\cdots)}^{k}$
- $\sigma_1 = \theta\{\hat{a} \mapsto t'\}$ *where* $\hat{a} \notin Dom(\theta)$
- σ *is the m.g.u of* $s_k\mu \stackrel{?}{=} sh^k(t')$ *where* $\mu = sh^k(\theta)$, *and* $\langle s, t|_k = \langle s_k, t|$.

- Iterating this decomposition gives us the follow complete decomposition of σ_{k+1}.

Complete Decomposition

Lemma

Let $k \geq 1$. If for all $0 \leq j \leq k$, $\langle s, t|_j$ is extendably unifiable by σ_j, then $\sigma_k = D(Id, s, t, k)$ where D is defined as follows:

$$D(\sigma, s, t, n+1) \equiv sh^n(\theta) D(\mathsf{s}^V(\sigma\theta), s, \mathsf{s}^V(t'), n)$$

$$D(\sigma, s, t, 0) \equiv \{\hat{a} \mapsto t\}$$

where $\theta\{\hat{a} \mapsto t'\}$ is the m.g.u. of $s\sigma \stackrel{?}{=} t$,$\hat{a} \notin Dom(\theta)$, and $V = V^x_{\mathbb{N}}$.

- At each step we unify the left term of the semiloop with the term paired with $\hat{a}$ in the solved form.
- The substitution $\mathsf{s}^V(\sigma\theta)$ can be restricted to relevant variables, Those occurring in s.

Example Decomposition

- Consider the following: $\langle s, t| = \langle h(t(0), \hat{a})\ ,\ h(y_1, h(y_2, y_1))|$ where $t(n) = h(x_{n+6}, h(x_{n+1}, x_{n+6}))$.
- $\langle s, t|_3 = \langle h(t(2), h(t(1), h(t(0), \hat{a})))\ ,\ h(y_1, h(y_2, y_1))\rangle$
- The solved form of $h(t(2), h(t(1), h(t(0), \hat{a}))) \stackrel{?}{=} t$ is

$$\{y_1 \stackrel{?}{=} h(x_8, h(x_3, x_8)),\ y_2 \stackrel{?}{=} h(x_7, h(x_2, x_7))$$
$$x_8 \stackrel{?}{=} h(x_6, h(x_1, x_6)),\ \hat{a} \stackrel{?}{=} h(x_3, h(x_6, h(x_1, x_6))\}$$

- The unifier of $\langle s, t|_3$ can be written as

$$D(Id, h(t(0), \hat{a}), h(y_1, h(y_2, y_1)), 3) =$$
$$sh^2(\sigma^2)D(sh^1(\sigma^2), s, h(y_2, t(1))), 2) =$$
$$sh^2(\sigma^2)sh^1(\sigma^1)D(sh^1(\sigma^1), s, t(2), 1) =$$
$$sh^2(\sigma^2)sh^1(\sigma^1)\sigma^0 D(sh^1(\sigma^0), s, h(x_4, t(1)), 0) =$$
$$sh^2(\sigma^2)sh^1(\sigma^1)\sigma^0\{\hat{a} \mapsto h(x_3, h(h(x_6, h(x_1, x_6))\}$$

Example Decomposition

- which is equivalent to the unifier:

$$\{y_1 \mapsto h(x_8, h(x_3, x_8))\ ,\ y_2 \mapsto h(x_7, h(x_2, x_7))\ ,$$
$$x_8 \mapsto h(x_6, h(x_1, x_6))\ ,\ \hat{a} \mapsto h(x_3, h(h(x_6, h(x_1, x_6))\}$$

- where

$$\sigma^2 = \{y_1 \mapsto h(x_6, h(x_1, x_6))\}$$
$$\sigma^1 = \{y_2 \mapsto h(x_6, h(x_1, x_6))\}$$
$$\sigma^0 = \{x_8 \mapsto h(x_6, h(x_1, x_6))\}$$

- Surprisingly, this loop is not infinitely unifiable as the 14-extension is not unifiable.

Sufficient Condition - Cyclicity

- The second and fourth argument of D doe not directly influence the construction of the unifier.
- This leaves the substitution $\mathrm{s}^V(\sigma\theta)$ and $\mathrm{s}^V(t')$ such that $\hat{a} \stackrel{?}{=} t'$ occurs in the solved form of some extension.
- When a unifier for a large enough extension decomposes as follows:

$$D'(Id, s, t, r+1) = \Theta(r+1)D'(\sigma_1^{\Delta}, s, t_1, r)$$

$$\vdots$$

$$D'(\sigma_{r-i+1}, s, t_{r-i+1}, i) = \Theta(i)D'(\sigma^*, s, t^*, i-1)$$

$$\vdots$$

$$D'(\sigma_{r-j+1}, s, t_{r-j+1}, j) = \Theta(j)D'(\sigma^*, s, t^*, j-1)$$

- We can construct a unifier for any extension.

Example with a Cycle

- Consider the semiloop

$$\langle s, t| = \langle h(\hat{a}, h(h(x_1, x_1), x_1)) \ , \ \ h(h(h(y_1, y_1), y_1), y_1)|,$$

  and we define $t(n) = h(h(x_{n+1}, x_{n+1}), x_{n+1}))$.
- Now consider the decomposition of $\langle s, t|_5$:

$$D(Id, s, t, 5) =$$

$$sh^4(\sigma_4)D(Id, s, h(h(t(1), t(1)), t(1)), 4) =$$

$$sh^4(\sigma_4)sh^3(\sigma_3)D'(\mathsf{Id}, s, h(t(1)), t(1))), 3) =$$

$$sh^4(\sigma_4)sh^3(\sigma_3)sh^2(\sigma_2)D'(Id, s, t(1), 2) =$$

$$sh^4(\sigma_4)sh^3(\sigma_3)sh^2(\sigma_2)sh^1(\sigma_1)D(\mathsf{Id}, s, h(t(1), t(1)), 1) =$$

$$sh^4(\sigma_4)sh^3(\sigma_3)sh^2(\sigma_2)sh^1(\sigma_1)\sigma_0\{\hat{f} \mapsto h(h(x_2, x_2), x_2)\}$$

Example with a Cycle

Where the substitutions σ_i are as follows:

$$\begin{aligned}
\sigma_4 =& \{y_2 \mapsto h(h(x_1, x_1), x_1)\} \\
\sigma_3 =& \{x_2 \mapsto x_1\} \\
\sigma_2 =& \{x_2 \mapsto x_1\} \\
\sigma_1 =& \{x_2 \mapsto h(h(x_1, x_1), x_1)\} \\
\sigma_0 =& \{x_2 \mapsto x_1\}
\end{aligned}$$

- Cycle repeats within the unifiers of large extensions of $\langle s, t|$
- If a cycle is found within the decomposition of a large enough extension, then the semiloop is infinite loop unifiable.

When Extension is not Large Enough

- Consider $\langle s, t|$ where:

$$s = h(h(x_1, h(x_{16}, h(x_{32}, h(x_1, h(x_{16}, x_{32}))))), \hat{f})$$

$$t = h(y_1, h(y_2, h(y_3, h(y_1, h(y_2, y_3)))))$$

- The unifier of $\langle s, t|_{11}$ decomposes such that the procedure reaches the following steps:

$$D(Id, s, h(x_4, h(x_{19}, h(x_{35}, h(x_4, h(x_{19}, x_{35}))))), 9)$$

$$D(Id, s, h(x_4, h(x_{19}, h(x_{35}, h(x_4, h(x_{19}, x_{35}))))), 4).$$

- This fits the cycle requirement, yet, $\langle s, t|_{28}$ is not unifiable.
- In this case, large enough means 64-extension.

Conclusion

- The results presented here provide a sufficient condition for semiloop unification.
- Evidence points to this condition being necessary, but we do not have a proof.
- Additionally, this concerning only a fragment of the loop unification problem.
- We are currently investigating how to extend these result to full loop unification with a restricted number of variable classes.