Proving and Solving in Unranked Theories

Part 2: Solving

Temur Kutsia

RISC, Johannes Kepler University, Linz, Austria
kutsia@risc.jku.at

Plan

First-order logic

From ranked to unranked languages

Resolution and unranked unification

Unranked matching and transformations

Unranked anti-unification and generalization

Plan

First-order logic

From ranked to unranked languages

Resolution and unranked unification

Unranked matching and transformations

Unranked anti-unification and generalization

Unification

Syntactic unification:

Given: Two (unranked) terms s and t.

Find: A substitution σ such that $\sigma(s) = \sigma(t)$.

- σ: a unifier of s and t.
- σ: a solution of the equation $s \doteq^? t$.

Example

$x \doteq^? f(y)$: infinitely many unifiers

$$\{x \mapsto f(y)\}, \{x \mapsto f(a), y \mapsto a\}, \ldots$$

Some solutions are better than the others: $\{x \mapsto f(y)\}$ is more general than $\{x \mapsto f(a), y \mapsto a\}$

Instantiation Quasi-Ordering

A substitution σ is more general than ϑ, written $\sigma \lesssim \vartheta$, if there exists η such that $\eta\sigma = \vartheta$.

ϑ is called an instance of σ.

The relation $\lesssim$ is reflexive and transitive binary relation, called instantiation quasi-ordering.

$\simeq$ is the equivalence relation corresponding to $\lesssim$, i.e., the relation $\lesssim \cap \gtrsim$.

Instantiation Quasi-Ordering

Example

Let $\sigma = \{x \mapsto y\}$, $\rho = \{x \mapsto a, y \mapsto a\}$, $\vartheta = \{y \mapsto x\}$.

- $\sigma \lesssim \rho$, because $\{y \mapsto a\}\sigma = \rho$.
- $\sigma \lesssim \vartheta$, because $\{y \mapsto x\}\sigma = \vartheta$.
- $\vartheta \lesssim \sigma$, because $\{x \mapsto y\}\vartheta = \sigma$.
- $\sigma \simeq \vartheta$.

Instantiation Quasi-Ordering

Example

Let $\sigma = \{\overline{x} \mapsto (\overline{y}, \overline{z}),\ \overline{z} \mapsto ()\}$, $\rho = \{\overline{x} \mapsto \overline{y}, \overline{z} \mapsto ()\}$, $\vartheta = \{\overline{y} \mapsto \overline{x},\ \overline{z} \mapsto ()\}$.

- $\sigma \lesssim \rho$, because $\{\overline{z} \mapsto ()\}\sigma = \rho$.
- $\sigma \lesssim \vartheta$, because $\vartheta\sigma = \vartheta$.
- $\rho \lesssim \vartheta$, because $\{\overline{y} \mapsto \overline{x}\}\rho = \vartheta$.
- $\vartheta \lesssim \rho$, because $\{\overline{x} \mapsto \overline{y}\}\vartheta = \rho$.
- $\vartheta \simeq \rho$.

Unification problem, unifier

Unification problem:

A finite set of equations $\Gamma = \{s_1 \doteq^? t_1, \ldots, s_n \doteq^? t_n\}$.

Unification problem, unifier

Unification problem:
A finite set of equations $\Gamma = \{s_1 \doteq^? t_1, \ldots, s_n \doteq^? t_n\}$.

Unifier or solution of Γ:
A substitution σ such that $\sigma(s_i) = \sigma(t_i)$ for all $1 \leqslant i \leqslant n$.

Unification problem, unifier

Unification problem:
A finite set of equations $\Gamma = \{s_1 \doteq^? t_1, \ldots, s_n \doteq^? t_n\}$.

Unifier or solution of Γ:
A substitution σ such that $\sigma(s_i) = \sigma(t_i)$ for all $1 \leqslant i \leqslant n$.

$\mathcal{U}(\Gamma)$: The set of all unifiers of Γ. Γ is unifiable iff $\mathcal{U}(\Gamma) \neq \emptyset$.

Minimal complete set of unifiers

A complete set of unifiers of a unification problem Γ is a set S of substitutions with the following properties:

- (Correctness) Each element of S is an unifier of Γ.
- (Completeness) For each unifier ϑ of Γ there exists $\sigma \in S$ with $\sigma \lesssim \vartheta$.

S is a minimal complete set of unifiers of Γ if it satisfies an additional property:

- (Minimality) No two distinct elements of S are comparable with respect to $\lesssim$.

Notation: $mcsu(\Gamma)$.

Minimal complete set of unifiers

$$\Gamma = \{\mathsf{f}(\overline{\mathsf{x}}, \mathsf{x}, \overline{\mathsf{y}}) \doteq^{?} \mathsf{f}(\mathsf{f}(\overline{\mathsf{x}}), \mathsf{x}, \mathsf{a}, \mathsf{b})\}$$

$$mcsu(\Gamma) = \{\{\overline{\mathsf{x}} \mapsto (),\ \mathsf{x} \mapsto \mathsf{f},\ \overline{\mathsf{y}} \mapsto (\mathsf{f}, \mathsf{a}, \mathsf{b})\}\}$$

Minimal complete set of unifiers

$$\Gamma = \{f(\overline{x}, x, \overline{y}) \doteq^{?} f(f(\overline{x}), x, a, b)\}$$
$$mcsu(\Gamma) = \{\{\overline{x} \mapsto (),\ x \mapsto f,\ \overline{y} \mapsto (f, a, b)\}\}$$

$$\Gamma = \{f(\overline{x}, a) \doteq^{?} f(b, \overline{y})\}$$
$$mcsu(\Gamma) = \{\{\overline{x} \mapsto (b, \overline{x}),\ \overline{y} \mapsto (\overline{x}, a)\}\}$$

Minimal complete set of unifiers

$$\begin{aligned}\Gamma &= \{f(\overline{x}, x, \overline{y}) \doteq^? f(f(\overline{x}), x, a, b)\} \\ mcsu(\Gamma) &= \{\{\overline{x} \mapsto (),\ x \mapsto f,\ \overline{y} \mapsto (f, a, b)\}\}\end{aligned}$$

$$\begin{aligned}\Gamma &= \{f(\overline{x}, a) \doteq^? f(b, \overline{y})\} \\ mcsu(\Gamma) &= \{\{\overline{x} \mapsto (b, \overline{x}),\ \overline{y} \mapsto (\overline{x}, a)\}\}\end{aligned}$$

$$\begin{aligned}\Gamma &= \{f(\overline{x}, x, \overline{y}, x, \overline{z}) \doteq^? f(a, b, c, a, b)\} \\ mcsu(\Gamma) &= \{\{\overline{x} \mapsto (),\ x \mapsto a,\ \overline{y} \mapsto (b, c),\ \overline{z} \mapsto b\}, \\ &\qquad \{\overline{x} \mapsto a,\ x \mapsto b,\ \overline{y} \mapsto (c, a),\ \overline{z} \mapsto ()\}\}\end{aligned}$$

Minimal complete set of unifiers

$$\Gamma = \{f(\overline{x}, x, \overline{y}) \doteq^? f(f(\overline{x}), x, a, b)\}$$
$$mcsu(\Gamma) = \{\{\overline{x} \mapsto (),\ x \mapsto f,\ \overline{y} \mapsto (f, a, b)\}\}$$

$$\Gamma = \{f(\overline{x}, a) \doteq^? f(b, \overline{y})\}$$
$$mcsu(\Gamma) = \{\{\overline{x} \mapsto (b, \overline{x}),\ \overline{y} \mapsto (\overline{x}, a)\}\}$$

$$\Gamma = \{f(\overline{x}, x, \overline{y}, x, \overline{z}) \doteq^? f(a, b, c, a, b)\}$$
$$\begin{aligned} mcsu(\Gamma) = \{&\{\overline{x} \mapsto (),\ x \mapsto a,\ \overline{y} \mapsto (b, c),\ \overline{z} \mapsto b\},\\ &\{\overline{x} \mapsto a,\ x \mapsto b,\ \overline{y} \mapsto (c, a),\ \overline{z} \mapsto ()\}\} \end{aligned}$$

$$\Gamma = \{f(\overline{x}, a) \doteq^? f(a, \overline{x})\}$$
$$mcsu(\Gamma) = \{\{\overline{x} \mapsto ()\}, \{\overline{x} \mapsto a\}, \{\overline{x} \mapsto (a, a)\}, \ldots\}$$

Minimal complete set of unifiers

Goal: design a procedure to compute minimal complete sets of unifiers for unranked unification problems.

Minimal complete set of unifiers

Goal: design a procedure to compute minimal complete sets of unifiers for unranked unification problems.

A rule-based procedure.

Rules transform pairs $\Gamma; \sigma$, where Γ is a unification problem and σ is a unifier computed so far.

Unification rules

T: **Trivial**

$$\{t \doteq^? t\} \cup \Gamma; \sigma \Longrightarrow \Gamma; \sigma.$$

S: **Solve**

$$\{x \doteq^? t\} \cup \Gamma; \sigma \Longrightarrow \vartheta(\Gamma); \vartheta\sigma.$$

where $x \notin \mathcal{V}_{\mathsf{ind}}(t)$ and $\vartheta = \{x \mapsto t\}$.

O: **Orient**

$$\{t \doteq^? x\} \cup \Gamma; \sigma \Longrightarrow \{x \doteq^? t\} \cup \Gamma; \sigma.$$

where $t \notin \mathcal{V}_{\mathsf{ind}}$.

Unification rules

LD: **Lazy decomposition**

$$\{f(t, s_1, \ldots, s_n) \doteq^? f(r, q_1, \ldots, q_m)\} \cup \Gamma; \sigma \Longrightarrow$$
$$\{t \doteq^? r, f(s_1, \ldots, s_n) \doteq^? f(q_1, \ldots, q_m)\} \cup \Gamma; \sigma,$$

where $t \notin \mathcal{V}_{\mathsf{seq}}$ and $r \notin \mathcal{V}_{\mathsf{seq}}$.

SVO: **Sequence variable orientation**

$$\{f(t, s_1, \ldots, s_n) \doteq^? f(\overline{x}, q_1, \ldots, q_m)\} \cup \Gamma; \sigma \Longrightarrow$$
$$\{f(\overline{x}, q_1, \ldots, q_m) \doteq^? f(t, s_1, \ldots, s_n)\} \cup \Gamma; \sigma,$$

where $t \notin \mathcal{V}_{\mathsf{seq}}$.

SVD: **Sequence variable deletion**

$$\{f(\overline{x}, s_1, \ldots, s_n) \doteq^? f(\overline{x}, q_1, \ldots, q_m)\} \cup \Gamma; \sigma \Longrightarrow$$
$$\{f(s_1, \ldots, s_n) \doteq^? f(q_1, \ldots, q_m)\} \cup \Gamma; \sigma.$$

Unification rules

SVE: **Sequence variable elimination**

$$\{f(\overline{x}, s_1, \dots, s_n) \doteq^? f(t, q_1, \dots, q_m)\} \cup \Gamma; \sigma \Longrightarrow$$
$$\vartheta(\{f(s_1, \dots, s_n) \doteq^? f(q_1, \dots, q_m)\} \cup \Gamma); \vartheta\sigma,$$

where $\overline{x} \notin \mathcal{V}_{\mathsf{seq}}(t)$ and $\vartheta = \{\overline{x} \mapsto t\}$.

SVW-1: **Sequence variable widening 1**

$$\{f(\overline{x}, s_1, \dots, s_n) \doteq^? f(t, q_1, \dots, q_m)\} \cup \Gamma; \sigma \Longrightarrow$$
$$\{f(\overline{x}, \vartheta(s_1), \dots, \vartheta(s_n)) \doteq^? \vartheta(f(q_1, \dots, q_m))\} \cup \vartheta(\Gamma); \vartheta\sigma,$$

where $\overline{x} \notin \mathcal{V}_{\mathsf{seq}}(t)$ and $\vartheta = \{\overline{x} \mapsto (t, \overline{x})\}$.

SVW-2: **Sequence variable widening 2**

$$\{f(\overline{x}, s_1, \dots, s_n) \doteq^? f(\overline{y}, q_1, \dots, q_m)\} \cup \Gamma; \sigma \Longrightarrow$$
$$\{\vartheta(f(s_1, \dots, s_n)) \doteq^? f(\overline{y}, \vartheta(q_1), \dots, \vartheta(q_m))\} \cup \vartheta(\Gamma); \vartheta\sigma,$$

where $\vartheta = \{\overline{y} \mapsto (\overline{x}, \overline{y})\}$.

Unification procedure

Steps to solve a unification problem Γ:

1. Let $\Gamma_1, \ldots, \Gamma_n$ be all the unification problems obtained by replacing some sequence variables in Γ by the empty sequence (erasing some sequence variables).
2. Let σ_i be the substitution giving Γ_i from Γ: $\Gamma_i = \sigma(\Gamma)$.
3. Starting from each $\Gamma_i; \sigma_i$, apply the unification rules iteratively. If a selected equation can be transformed by several rules, apply them in parallel.
4. Whenever a pair of the form $\emptyset; \vartheta$ is obtained, return ϑ.

Unification procedure: example 1

Let the unification problem be $\{\mathsf{f}(\overline{\mathsf{x}}, \mathsf{a}) \doteq^? \mathsf{f}(\mathsf{a}, \overline{\mathsf{y}})\}$.

After the projection step, we obtain four pairs:

1. $\{\mathsf{f}(\mathsf{a}) \doteq^? \mathsf{f}(\mathsf{a})\}; \{\overline{\mathsf{x}} \mapsto (), \overline{\mathsf{y}} \mapsto ()\}$.
2. $\{\mathsf{f}(\overline{\mathsf{x}}, \mathsf{a}) \doteq^? \mathsf{f}(\mathsf{a})\}; \{\overline{\mathsf{y}} \mapsto ()\}$.
3. $\{\mathsf{f}(\mathsf{a}) \doteq^? \mathsf{f}(\mathsf{a}, \overline{\mathsf{y}})\}; \{\overline{\mathsf{x}} \mapsto ()\}$.
4. $\{\mathsf{f}(\overline{\mathsf{x}}, \mathsf{a}) \doteq^? \mathsf{f}(\mathsf{a}, \overline{\mathsf{y}})\}; Id$

Unification procedure: example 1

Solving each of the obtained pairs.

1. $\{f(a) \doteq^? f(a)\}; \{\overline{x} \mapsto (), \overline{y} \mapsto ()\} \Longrightarrow_{\mathsf{T}}$
   $\emptyset; \{\overline{x} \mapsto (), \overline{y} \mapsto ()\}$.

Hence, the substitution $\{\overline{x} \mapsto (), \overline{y} \mapsto ()\}$ is one computed result.

Unification procedure: example 1

Solving each of the obtained pairs.

2. $\{f(\overline{x}, a) \doteq^? f(a)\}; \{\overline{y} \mapsto ()\} \Longrightarrow_{\mathsf{SVE}}$
   $\{f(a) \doteq^? f()\}; \{\overline{x} \mapsto a, \overline{y} \mapsto ()\}$.

No rule can be applied. This branch fails.

Unification procedure: example 1

Solving each of the obtained pairs.

2. $\{f(\overline{x}, a) \doteq^? f(a)\}; \{\overline{y} \mapsto ()\} \Longrightarrow_{\mathsf{SVE}}$
   $\{f(a) \doteq^? f()\}; \{\overline{x} \mapsto a, \overline{y} \mapsto ()\}$.

No rule can be applied. This branch fails.

Alternative:

2. $\{f(\overline{x}, a) \doteq^? f(a)\}; \{\overline{y} \mapsto ()\} \Longrightarrow_{\mathsf{SVW1}}$
   $\{f(\overline{x}, a) \doteq^? f()\}; \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto ()\}$.

No rule can be applied. This branch also fails.

Unification procedure: example 1

Solving each of the obtained pairs.

3. $\{f(a) \doteq^? f(a, \overline{y})\}; \{\overline{x} \mapsto ()\} \Longrightarrow_{\mathsf{O}}$

   $\{f(a, \overline{y}) \doteq^? f(a)\}; \{\overline{x} \mapsto ()\} \Longrightarrow_{\mathsf{LD}}$

   $\{a \doteq^? a, f(\overline{y}) \doteq^? f()\}; \{\overline{x} \mapsto ()\} \Longrightarrow_{\mathsf{T}}$

   $\{f(\overline{y}) \doteq^? f()\}; \{\overline{x} \mapsto ()\}.$

No rule can be applied. This case fails.

Unification procedure: example 1

Solving each of the obtained pairs.

The fourth one can be transformed either by SVE or by SVW1.

Applying SVE:

4. $\{\mathsf{f}(\overline{\mathsf{x}}, \mathsf{a}) \doteq^? \mathsf{f}(\mathsf{a}, \overline{\mathsf{y}})\}; Id \Longrightarrow_{\mathsf{SVE}}$

$\{\mathsf{f}(\mathsf{a}) \doteq^? \mathsf{f}(\overline{\mathsf{y}})\}; \{\overline{\mathsf{x}} \mapsto \mathsf{a}\} \Longrightarrow_{\mathsf{O}}$

$\{\mathsf{f}(\overline{\mathsf{y}}) \doteq^? \mathsf{f}(\mathsf{a})\}; \{\overline{\mathsf{x}} \mapsto \mathsf{a}\} \Longrightarrow_{\mathsf{SVE}}$

$\{\mathsf{f}() \doteq^? \mathsf{f}()\}; \{\overline{\mathsf{x}} \mapsto \mathsf{a}, \overline{\mathsf{y}} \mapsto \mathsf{a}\} \Longrightarrow_{\mathsf{T}}$

$\emptyset; \{\overline{\mathsf{x}} \mapsto \mathsf{a}, \overline{\mathsf{y}} \mapsto \mathsf{a}\}$

The second computed result: $\{\overline{\mathsf{x}} \mapsto \mathsf{a}, \overline{\mathsf{y}} \mapsto \mathsf{a}\}$.

Unification procedure: example 1

Solving each of the obtained pairs.

The fourth one can be transformed either by SVE or by SVW1.

Applying SVE:

4. $\{f(\overline{x}, a) \doteq^? f(a, \overline{y})\}; Id \Longrightarrow_{\mathsf{SVE}}$
$\{f(a) \doteq^? f(\overline{y})\}; \{\overline{x} \mapsto a\} \Longrightarrow_{\mathsf{O}}$
$\{f(\overline{y}) \doteq^? f(a)\}; \{\overline{x} \mapsto a\} \Longrightarrow_{\mathsf{SVE}}$
$\{f() \doteq^? f()\}; \{\overline{x} \mapsto a, \overline{y} \mapsto a\} \Longrightarrow_{\mathsf{T}}$
$\emptyset; \{\overline{x} \mapsto a, \overline{y} \mapsto a\}$

The second computed result: $\{\overline{x} \mapsto a, \overline{y} \mapsto a\}$.

$\{f(\overline{y}) \doteq^? f(a)\}; \{\overline{x} \mapsto a\} \Longrightarrow_{\mathsf{SVW1}} \{f(\overline{y}) \doteq^? f()\}; \{\overline{x} \mapsto a, \overline{y} \mapsto (a, \overline{y})\}$ fails.

Unification procedure: example 1

Solving each of the obtained pairs.

The fourth one cane be transformed either by SVE or by SVW1.

Applying SVW1:

4. $\{\mathsf{f}(\overline{\mathsf{x}}, \mathsf{a}) \doteq^? \mathsf{f}(\mathsf{a}, \overline{\mathsf{y}})\}; Id \Longrightarrow_{\mathsf{SVW1}}$

$\{\mathsf{f}(\overline{\mathsf{x}}, \mathsf{a}) \doteq^? \mathsf{f}(\overline{\mathsf{y}})\}; \{\overline{\mathsf{x}} \mapsto (\mathsf{a}, \overline{\mathsf{x}})\}.$

The new pair can be transformed in three ways.

Unification procedure: example 1

The new pair $\{f(\overline{x}, a) \doteq^? f(\overline{y})\}; \{\overline{x} \mapsto (a, \overline{x})\}$ can be transformed in three ways, by rules SVE, SVW1, and SVW2.

The first two of them fail:

$$\{f(\overline{x}, a) \doteq^? f(\overline{y})\}; \{\overline{x} \mapsto (a, \overline{x})\} \Longrightarrow_{\text{SVE}}$$
$$\{f(a) \doteq^? f()\}; \{\overline{x} \mapsto (a, \overline{y})\}$$ No rule applies, fail.

$$\{f(\overline{x}, a) \doteq^? f(\overline{y})\}; \{\overline{x} \mapsto (a, \overline{x})\} \Longrightarrow_{\text{SVW1}}$$
$$\{f(\overline{x}, a) \doteq^? f()\}; \{\overline{x} \mapsto (a, \overline{y}, \overline{x})\}$$ No rule applies, fail.

Unification procedure: example 1

The new pair $\{f(\overline{x}, a) \doteq^? f(\overline{y})\}; \{\overline{x} \mapsto (a, \overline{x})\}$ can be transformed in three ways, by rules SVE, SVW1, and SVW2.

The third one, by SVW2, proceeds as follows:

$$\{f(\overline{x}, a) \doteq^? f(\overline{y})\}; \{\overline{x} \mapsto (a, \overline{x})\} \Longrightarrow_{\mathsf{SVW2}}$$
$$\{f(a) \doteq^? f(\overline{y})\}; \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, \overline{y})\} \Longrightarrow_{\mathsf{O}}$$
$$\{f(\overline{y}) \doteq^? f(a)\}; \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, \overline{y})\}$$

Two alternatives to continue from

$$\{f(\overline{y}) \doteq^? f(a)\}; \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, \overline{y})\}.$$

Unification procedure: example 1

Two alternatives (SVE and SVW1) to continue from

$$\{f(\overline{y}) \doteq^? f(a)\}; \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, \overline{y})\}.$$

The first one, by SVE:

$$\{f(\overline{y}) \doteq^? f(a)\}; \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, \overline{y})\} \Longrightarrow_{\mathsf{SVE}}$$
$$\{f() \doteq^? f()\}; \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, a)\} \Longrightarrow_{\mathsf{T}}$$
$$\emptyset; \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, a)\}.$$

We got the third computed result $\{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, a)\}$.

Unification procedure: example 1

The second alternative, by SVW1:

$$\{f(\overline{y}) \doteq^? f(a)\}; \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, \overline{y})\} \Longrightarrow_{\mathsf{SVW1}}$$
$$\{f(\overline{y}) \doteq^? f()\}; \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, a, \overline{y})\}$$

No rule applies. This alternative fails.

Unification procedure: example 1

Summary:

For the unification problem be $\{f(\overline{x}, a) \doteq^{?} f(a, \overline{y})\}$ the procedure stops and returns three solutions:

$$\vartheta_1 = \{\overline{x} \mapsto (), \overline{y} \mapsto ()\}.$$

$$\vartheta_2 = \{\overline{x} \mapsto a, \overline{y} \mapsto a\}.$$

$$\vartheta_3 = \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, a)\}.$$

Unification procedure: example 1

Summary:

For the unification problem be $\{f(\overline{x}, a) \doteq^? f(a, \overline{y})\}$ the procedure stops and returns three solutions:

$$\vartheta_1 = \{\overline{x} \mapsto (), \overline{y} \mapsto ()\}.$$
$$\vartheta_2 = \{\overline{x} \mapsto a, \overline{y} \mapsto a\}.$$
$$\vartheta_3 = \{\overline{x} \mapsto (a, \overline{x}), \overline{y} \mapsto (\overline{x}, a)\}.$$

Note that $\vartheta_2 \lesssim \vartheta_2$: $\{\overline{x} \mapsto ()\}\vartheta_3 = \vartheta_2$.

Hence, the set $\{\vartheta_1, \vartheta_2, \vartheta_3\}$ is not minimal, but “almost minimal”: two substitutions can be related only via erasing substitutions.

Unification procedure: example 2

$$\{f(\overline{x}, a) \doteq^{?} f(a, \overline{x})\}$$

Unification procedure: example 2

$$
\begin{tikzcd}[column sep=tiny]
& \{f(\overline{x}, a) \doteq^{?} f(a, \overline{x})\} \arrow[dl] \arrow[dr] & \\
\{f(a) \doteq^{?} f(a)\}; \{\overline{x} \mapsto ()\} & & \{f(\overline{x}, a) \doteq^{?} f(a, \overline{x})\}; Id
\end{tikzcd}
$$

Unification procedure: example 2

$$\begin{array}{ccc} & \{f(\overline{x}, a) \doteq^? f(a, \overline{x})\} & \\ \swarrow & & \searrow \\ \{f(a) \doteq^? f(a)\}; \{\overline{x} \mapsto ()\} & & \{f(\overline{x}, a) \doteq^? f(a, \overline{x})\}; Id \\ \downarrow & & \\ \emptyset; \{\overline{x} \mapsto ()\} & & \end{array}$$

Unification procedure: example 2

$$\{f(\overline{x}, a) \doteq^{?} f(a, \overline{x})\}$$

$$\{f(a) \doteq^{?} f(a)\}; \{\overline{x} \mapsto ()\} \qquad \{f(\overline{x}, a) \doteq^{?} f(a, \overline{x})\}; Id$$

$$\emptyset; \{\overline{x} \mapsto ()\} \qquad \{f(a) \doteq^{?} f(a)\}; \{\overline{x} \mapsto a\}$$

Unification procedure: example 2

$$\begin{tikzcd}
 & \{\mathsf{f}(\overline{x}, a) \doteq^? \mathsf{f}(a, \overline{x})\} \arrow[dl] \arrow[dr] & \\
\{\mathsf{f}(a) \doteq^? \mathsf{f}(a)\}; \{\overline{x} \mapsto ()\} \arrow[d] & & \{\mathsf{f}(\overline{x}, a) \doteq^? \mathsf{f}(a, \overline{x})\}; Id \arrow[dl] \\
\emptyset; \{\overline{x} \mapsto ()\} & \{\mathsf{f}(a) \doteq^? \mathsf{f}(a)\}; \{\overline{x} \mapsto a\} \arrow[d] & \\
 & \emptyset; \{\overline{x} \mapsto a\} &
\end{tikzcd}$$

Unification procedure: example 2

$$
\begin{tikzcd}[column sep=small]
 & & \{f(\overline{x}, a) \doteq^? f(a, \overline{x})\} \arrow[dll] \arrow[dr] & \\
\{f(a) \doteq^? f(a)\}; \{\overline{x} \mapsto ()\} \arrow[d] & & & \{f(\overline{x}, a) \doteq^? f(a, \overline{x})\}; Id \arrow[dll] \arrow[ddd] \\
\emptyset; \{\overline{x} \mapsto ()\} & \{f(a) \doteq^? f(a)\}; \{\overline{x} \mapsto a\} \arrow[d] & & \\
 & \emptyset; \{\overline{x} \mapsto a\} & & \\
 & & & \{f(\overline{x}, a) \doteq^? f(a, \overline{x})\}; \{\overline{x} \mapsto (a, \overline{x})\}
\end{tikzcd}
$$

Unification procedure: example 2

$$
\begin{tikzcd}[column sep=small]
 & \{f(\overline{x}, a) \doteq^? f(a, \overline{x})\} \arrow[dl] \arrow[dr] & \\
\{f(a) \doteq^? f(a)\}; \{\overline{x} \mapsto ()\} \arrow[d] & & \{f(\overline{x}, a) \doteq^? f(a, \overline{x})\}; Id \arrow[dl] \arrow[dd] \\
\emptyset; \{\overline{x} \mapsto ()\} & \{f(a) \doteq^? f(a)\}; \{\overline{x} \mapsto a\} \arrow[d] & \\
 & \emptyset; \{\overline{x} \mapsto a\} & \{f(\overline{x}, a) \doteq^? f(a, \overline{x})\}; \{\overline{x} \mapsto (a, \overline{x})\} \arrow[dl] \\
 & \{f(a, a) \doteq^? f(a, a)\}; \{\overline{x} \mapsto (a, a)\} \arrow[d] & \\
 & \emptyset; \{\overline{x} \mapsto (a, a)\} &
\end{tikzcd}
$$

Unification procedure: example 2

$$
\begin{tikzcd}[column sep=small]
 & \{f(\overline{x}, a) \doteq^{?} f(a, \overline{x})\} \arrow[dl] \arrow[dr] & \\
\{f(a) \doteq^{?} f(a)\}; \{\overline{x} \mapsto (\,)\} \arrow[d] & & \{f(\overline{x}, a) \doteq^{?} f(a, \overline{x})\}; \mathit{Id} \arrow[dl] \arrow[ddd] \\
\emptyset; \{\overline{x} \mapsto (\,)\} & \{f(a) \doteq^{?} f(a)\}; \{\overline{x} \mapsto a\} \arrow[d] & \\
 & \emptyset; \{\overline{x} \mapsto a\} & \\
 & & \{f(\overline{x}, a) \doteq^{?} f(a, \overline{x})\}; \{\overline{x} \mapsto (a, \overline{x})\} \arrow[dl] \arrow[ddd] \\
 & \{f(a, a) \doteq^{?} f(a, a)\}; \{\overline{x} \mapsto (a, a)\} \arrow[d] & \\
 & \emptyset; \{\overline{x} \mapsto (a, a)\} & \\
 & & \{f(\overline{x}, a) \doteq^{?} f(a, \overline{x})\}; \{\overline{x} \mapsto (a, a, \overline{x})\} \\
 & & \cdots
\end{tikzcd}
$$

Unification procedure: example 2

The procedure computes infinitely many substitutions for $\{f(\overline{x}, a) \doteq^{?} f(a, \overline{x})\}$.

Unification procedure: example 3

$\{f(\overline{x}, a, \overline{x}) \doteq^? f(a, \overline{x}, a)\}$ has a single solution $\{\overline{x} \mapsto a\}$.

The procedure returns it, but still does not stop.

Generates infinitely many unsolvable problems
$\{f(\overline{x}, a, \ldots, a, \overline{x}) \doteq^? f(a, \overline{x}, a)\}$.

Unification procedure: example 3

$\{f(\overline{x}, a, \overline{x}) \doteq^? f(a, \overline{x}, a)\}$ has a single solution $\{\overline{x} \mapsto a\}$.

The procedure returns it, but still does not stop.

Generates infinitely many unsolvable problems
$\{f(\overline{x}, a, \ldots, a, \overline{x}) \doteq^? f(a, \overline{x}, a)\}$.

It can be prevented by combining decision and unification procedures.

Unification procedure: example 3

$\{f(\overline{x}, a, \overline{x}) \doteq^? f(a, \overline{x}, a)\}$ has a single solution $\{\overline{x} \mapsto a\}$.

The procedure returns it, but still does not stop.

Generates infinitely many unsolvable problems
$\{f(\overline{x}, a, \ldots, a, \overline{x}) \doteq^? f(a, \overline{x}, a)\}$.

It can be prevented by combining decision and unification procedures.

Decidability of unranked unification reduces to decidability of word equations (Makanin 1977).

Unification procedure: properties

The procedure enumerates a complete set of unifiers of the given unification problems.

The computed set, in general, is not minimal, but two substitutions may be related to each other only by variable erasing substitutions.

With the decision procedure, the procedure terminates when the minimal complete set of unifiers is finite.

Unification procedure: terminating cases

Restricting the form of unification problems, we obtain terminating fragments of unranked unification.

Unification procedure: terminating cases

Restricting the form of unification problems, we obtain terminating fragments of unranked unification.

Linear fragment: no variable appears more than once.

Example: $f(\overline{x}, f(\overline{y}, \overline{z})) \doteq^? f(a, b, f(x, c))$.

Solutions:

$$\{\overline{x} \mapsto (a, b), \overline{y} \mapsto (), \overline{y} \mapsto (x, c)\}$$

$$\{\overline{x} \mapsto (a, b), \overline{y} \mapsto x, \overline{y} \mapsto c\}$$

$$\{\overline{x} \mapsto (a, b), \overline{y} \mapsto (x, c), \overline{y} \mapsto ()\}$$

Unification procedure: terminating cases

KIF fragment: sequence variables appear only in the last argument positions.

Example: $f(f(a, \overline{x}), g(x, y, \overline{x}), \overline{y}) = f(f(x, a, \overline{y}), g(a, \overline{z}), \overline{u})$.

Single solution:

$$\{x \mapsto a, \overline{x} \mapsto (a, \overline{u}), \overline{z} \mapsto (y, a, \overline{u}), \overline{y} \mapsto \overline{u}\}.$$

Unification procedure: terminating cases

Matching fragment: variables appear only in one side of unification equation.

Example: $f(\overline{x}, f(a, \overline{y}, \overline{z})) = f(f(a, b)$.

Solutions:

$$\{\overline{x} \mapsto (), \overline{y} \mapsto (), \overline{z} \mapsto b\}.$$
$$\{\overline{x} \mapsto (), \overline{y} \mapsto b, \overline{z} \mapsto ()\}.$$

(does not need to be linear)

Matching algorithm

For matching, we do not need all the rules of the unification procedure.

T, S, LD and a modified version of SVE suffice.

SVEM: **Sequence variable elimination modified**

$$\{f(\overline{x}, s_1, \ldots, s_n) \doteq^? f(q_1, \ldots, q_i, \ldots, q_m)\} \cup \Gamma; \sigma \Longrightarrow$$
$$\vartheta(\{f(s_1, \ldots, s_n) \doteq^? f(q_{i+1}, \ldots, q_m)\} \cup \Gamma); \vartheta\sigma,$$

where $0 \leqslant i \leqslant m$ and $\vartheta = \{\overline{x} \mapsto (q_1, \ldots, q_i)\}$.

Matching algorithm

For matching, we do not need all the rules of the unification procedure.

T, S, LD and a modified version of SVE suffice.

SVEM: **Sequence variable elimination modified**

$$\{f(\overline{x}, s_1, \ldots, s_n) \doteq^? f(q_1, \ldots, q_i, \ldots, q_m)\} \cup \Gamma; \sigma \Longrightarrow$$
$$\vartheta(\{f(s_1, \ldots, s_n) \doteq^? f(q_{i+1}, \ldots, q_m)\} \cup \Gamma); \vartheta\sigma,$$

where $0 \leqslant i \leqslant m$ and $\vartheta = \{\overline{x} \mapsto (q_1, \ldots, q_i)\}$.

Unranked matching is the main computational mechanism in ρLog calculus.

ρLog, very briefly

ρLog is a calculus for strategy-based conditional transformations of term sequences.

Rules have the form

$$\mathit{strategy} :: \mathit{sequence}_1 \rightarrow \mathit{sequence}_2 \textbf{ if } \mathit{condition}$$

Meaning: $\mathit{strategy}$ transforms $\mathit{sequence}_1$ to $\mathit{sequence}_2$ if $\mathit{condition}$ holds.

ρLog, very briefly

Example: find a duplicated element in a sequence and remove one of the two copies.

$$merge\_doubles :: (\overline{x}, x, \overline{y}, x, \overline{z}) \to (\overline{x}, x, \overline{y}, \overline{z}).$$

The condition is empty (true).

ρLog, very briefly

Example: find a duplicated element in a sequence and remove one of the two copies.

$$merge\_doubles :: (\overline{x}, x, \overline{y}, x, \overline{z}) \to (\overline{x}, x, \overline{y}, \overline{z}).$$

The condition is empty (true).

Apply the rule to merge doubles in $(1, 2, 3, 2, 1)$:

$$merge\_doubles :: (1, 2, 3, 2, 1) \to \overline{u}.$$

A computed answer is $\quad \overline{u} \mapsto (1, 2, 3, 2)$.

Another answer is $\quad \overline{u} \mapsto (1, 2, 3, 1)$.

ρLog, very briefly

Example: find a duplicated element in a sequence and remove one of the two copies.

$$merge\_doubles :: (\overline{x}, x, \overline{y}, x, \overline{z}) \to (\overline{x}, x, \overline{y}, \overline{z}).$$

The condition is empty (true).

Apply the rule to merge doubles in $(1, 2, 3, 2, 1)$:

$$merge\_doubles :: (1, 2, 3, 2, 1) \to \overline{u}.$$

A computed answer is $\quad \overline{u} \mapsto (1, 2, 3, 2)$.

Another answer is $\quad \overline{u} \mapsto (1, 2, 3, 1)$.

How does it work?

ρLog, very briefly

Example: remove all duplicates.

$$
\begin{aligned}
&\mathit{merge\_all\_doubles} :: \bar{x} \to \bar{y} \textbf{ if} \\
&\qquad \textbf{nf}(\mathit{merge\_doubles}) :: \bar{x} \to \bar{y}.
\end{aligned}
$$

nf is the ρLog strategy for normal form computation.

nf$(\mathit{merge\_doubles}) :: \bar{x} \to \bar{y}$ applies $\mathit{merge\_doubles}$ to $\bar{x}$ as long as possible and returns the computed result in $\bar{y}$.

ρLog, very briefly

Example: remove all duplicates.

$$\mathit{merge\_all\_doubles} :: \overline{x} \to \overline{y} \textbf{ if}$$
$$\textbf{nf}(\mathit{merge\_doubles}) :: \overline{x} \to \overline{y}.$$

nf is the ρLog strategy for normal form computation.

$\textbf{nf}(\mathit{merge\_doubles}) :: \overline{x} \to \overline{y}$ applies $\mathit{merge\_doubles}$ to $\overline{x}$ as long as possible and returns the computed result in $\overline{y}$.

Apply the rule to merge all doubles in $(1, 2, 3, 2, 1)$:

$$\mathit{merge\_all\_doubles} :: (1, 2, 3, 2, 1) \to \overline{u}.$$

Computed answer is $\overline{u} \mapsto (1, 2, 3)$.

ρLog, very briefly

Example: remove all duplicates.

$$\mathit{merge\_all\_doubles} :: \overline{x} \to \overline{y} \textbf{ if}$$
$$\textbf{nf}(\mathit{merge\_doubles}) :: \overline{x} \to \overline{y}.$$

nf is the ρLog strategy for normal form computation.

$\textbf{nf}(\mathit{merge\_doubles}) :: \overline{x} \to \overline{y}$ applies $\mathit{merge\_doubles}$ to $\overline{x}$ as long as possible and returns the computed result in $\overline{y}$.

Apply the rule to merge all doubles in $(1, 2, 3, 2, 1)$:

$$\mathit{merge\_all\_doubles} :: (1, 2, 3, 2, 1) \to \overline{u}.$$

Computed answer is $\overline{u} \mapsto (1, 2, 3)$.

How does it work?

The anti-unification problem

Given: Two terms t_1 and t_2.

Find: Their generalization, a term t such that both t_1 and t_2 are instances of t under some substitutions.

The anti-unification problem

Given: Two terms t_1 and t_2.

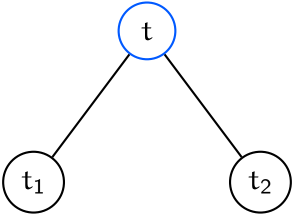
Find: Their generalization, a term t such that both t_1 and t_2 are instances of t under some substitutions.

t_1 $\qquad$ t_2

The anti-unification problem

Given: Two terms t_1 and t_2.

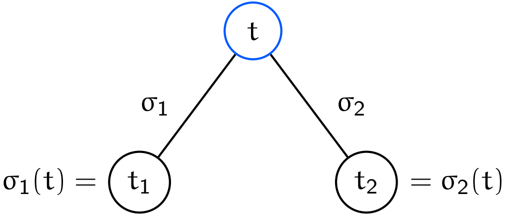
Find: Their generalization, a term t such that both t_1 and t_2 are instances of t under some substitutions.



The anti-unification problem

Given: Two terms t_1 and t_2.

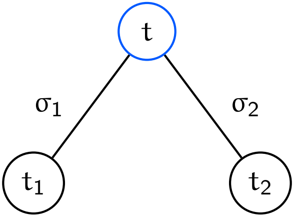
Find: Their generalization, a term t such that both t_1 and t_2 are instances of t under some substitutions.



The anti-unification problem

Given: Two terms t_1 and t_2.

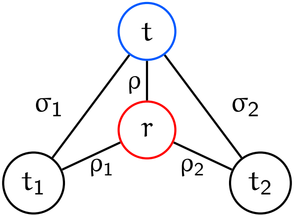
Find: Their least general generalization t.



The anti-unification problem

Given: Two terms t_1 and t_2.

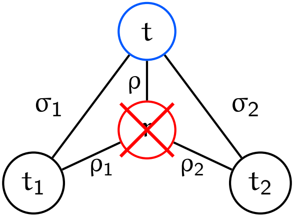
Find: Their least general generalization t.



The anti-unification problem

Given: Two terms t_1 and t_2.

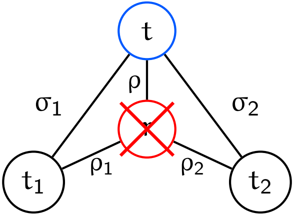
Find: Their least general generalization t.



The anti-unification problem

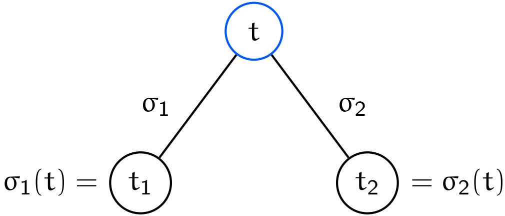
Given: Two terms t_1 and t_2.

Find: Their least general generalization t.

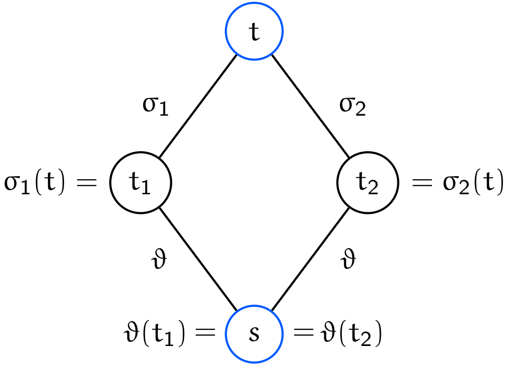


- $f(x, x)$ is the lgg of $f(a, a)$ and $f(b, b)$.
- $f(x, y)$ is not.

Anti-unification and unification

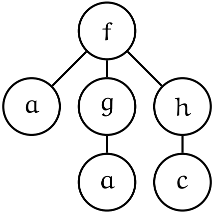


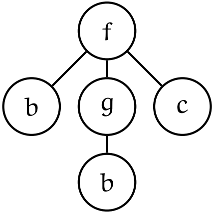
Anti-unification and unification



Example: generalizing ranked terms

Given terms:

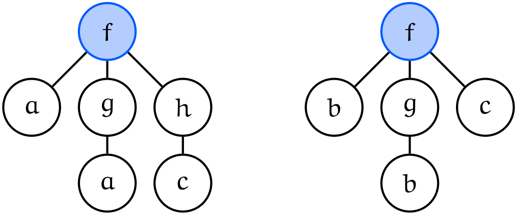




Example: generalizing ranked terms

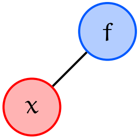
Generalization:

Given terms:

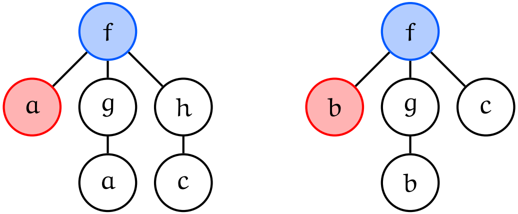


Example: generalizing ranked terms

Generalization:

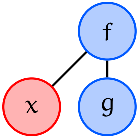


Given terms:

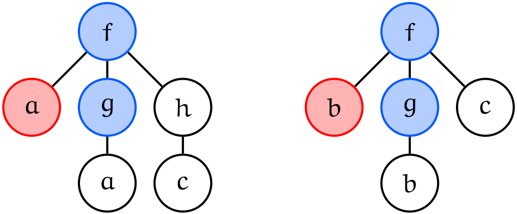


Example: generalizing ranked terms

Generalization:

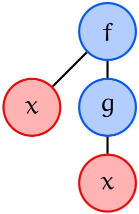


Given terms:

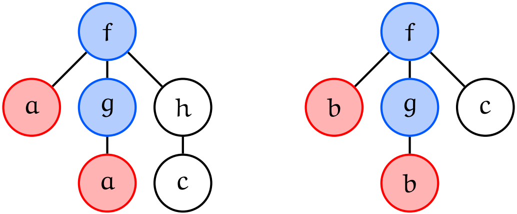


Example: generalizing ranked terms

Generalization:



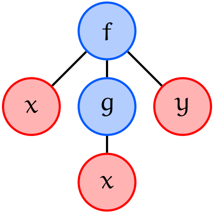
Given terms:



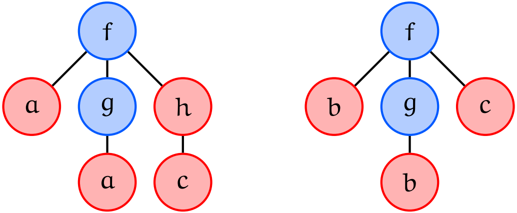
Example: generalizing ranked terms

Generalization:

$$x : a \triangleq b$$
$$y : h(c) \triangleq c$$

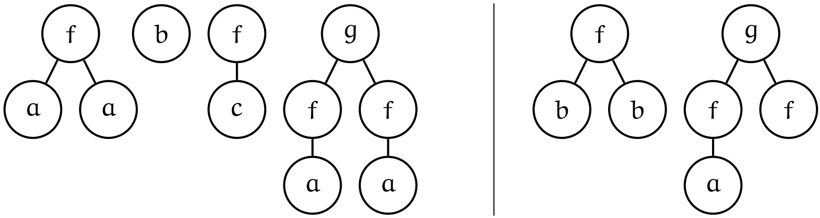


Given terms:



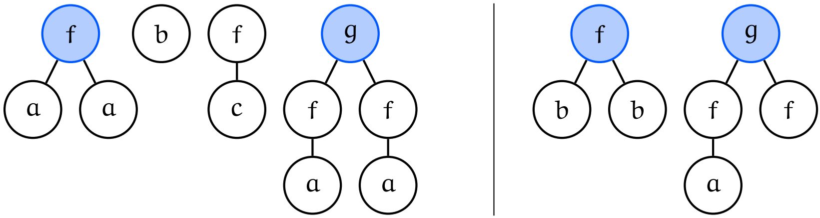
Example: generalizing unranked term sequences

Given sequences:



Example: generalizing unranked term sequences

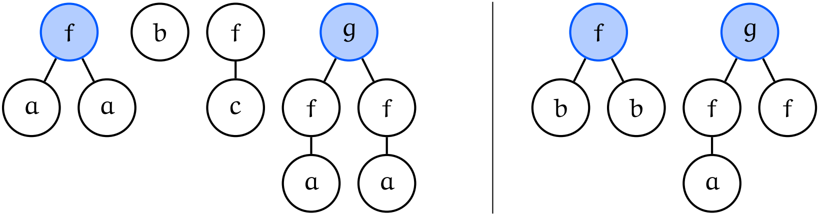
Given sequences:



Example: generalizing unranked term sequences

Generalization:

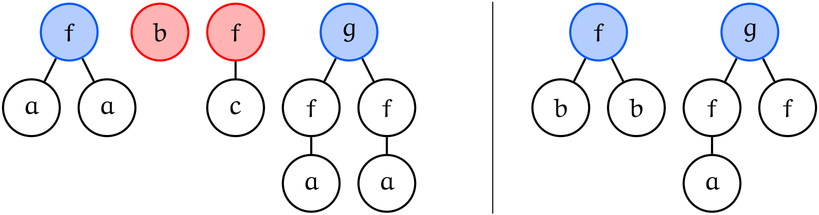
Given sequences:



Example: generalizing unranked term sequences

Generalization:

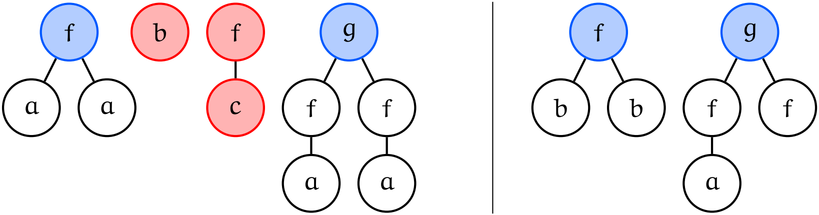
Given sequences:



Example: generalizing unranked term sequences

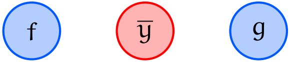
Generalization:

Given sequences:

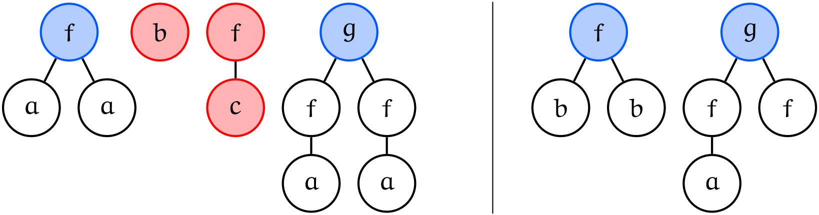


Example: generalizing unranked term sequences

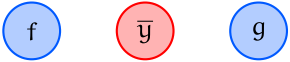
Generalization:



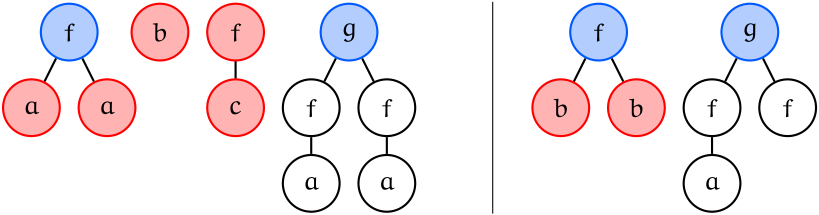
Given sequences:



Example: generalizing unranked term sequences

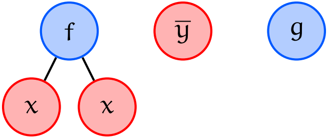
Generalization: 

Given sequences:

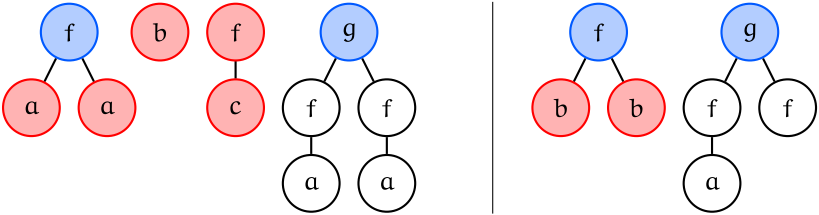


Example: generalizing unranked term sequences

Generalization:

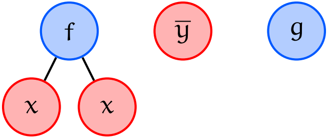


Given sequences:

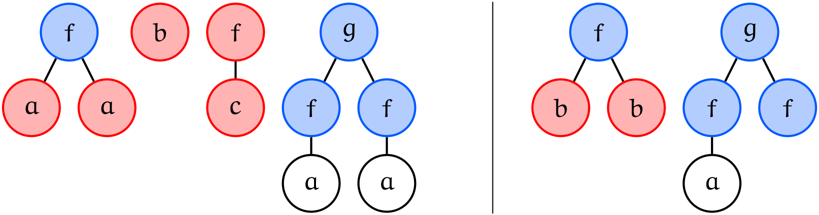


Example: generalizing unranked term sequences

Generalization:

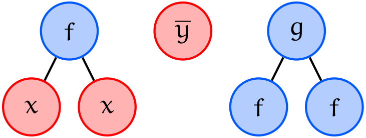


Given sequences:

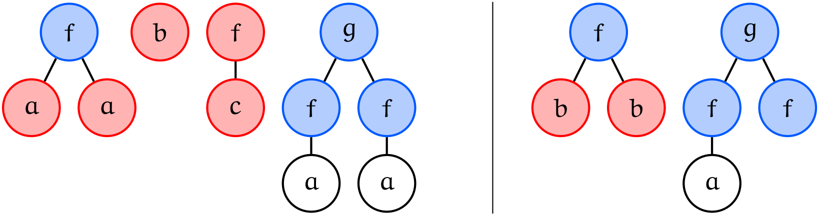


Example: generalizing unranked term sequences

Generalization:

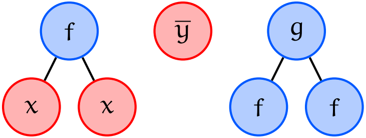


Given sequences:

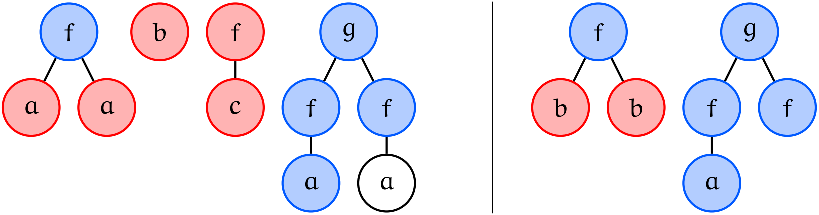


Example: generalizing unranked term sequences

Generalization:

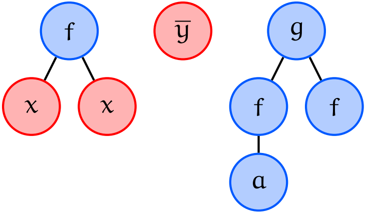


Given sequences:

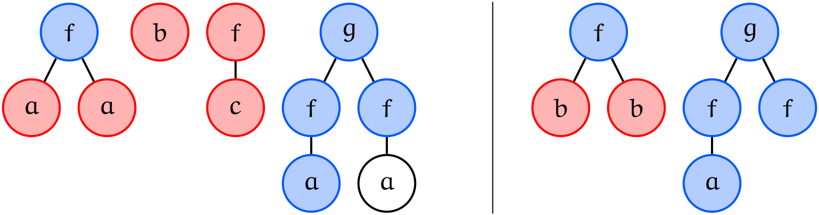


Example: generalizing unranked term sequences

Generalization:

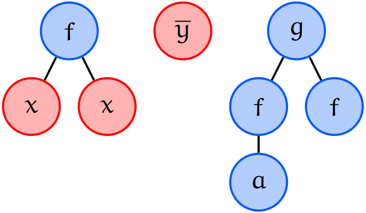


Given sequences:

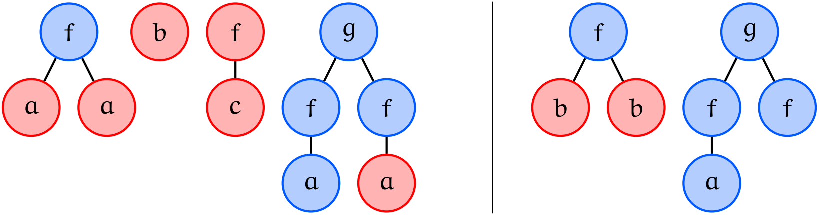


Example: generalizing unranked term sequences

Generalization:

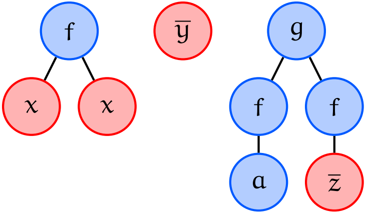


Given sequences:

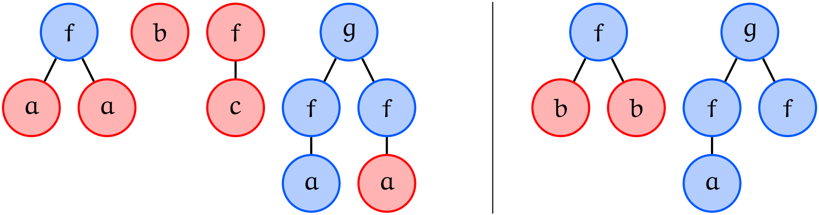


Example: generalizing unranked term sequences

Generalization:



Given sequences:



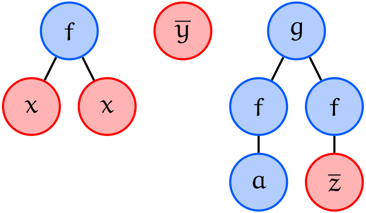
Example: generalizing unranked term sequences

Generalization:

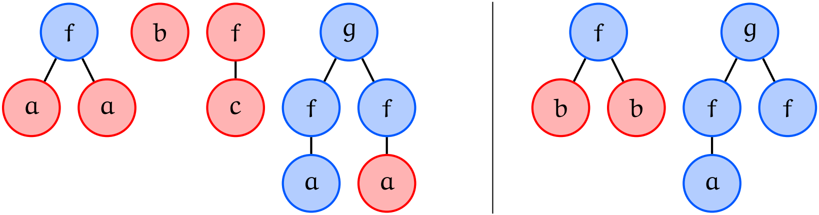
$x \;:\; a \triangleq b$

$\overline{y} \;:\; (b, f(c)) \triangleq ()$

$\overline{z} \;:\; a \triangleq ()$



Given sequences:



Applications of anti-unification

- Inductive logic programming
- Automatic bug fixing
- Software code clone detection and refactoring
- Analogy making
- indexing and compression
- Chatbots
- ...

Software

An open-source unification and anti-unification library:

https://www.risc.jku.at/projects/stout/library.html

Summary

We discussed

- syntax and semantics of an unranked language with sequence variables
- inference system, problems cause by sequence variables
- unification, needed in the inference system
- complete and almost minimal non-terminating unification procedure
- terminating cases of unification
- matching, the ρLog calculus for sequence transformations
- the notion of generalization and anti-unification

References

Literature on the Web page of the project SToUT:
https://www.risc.jku.at/projects/stout/

Information technology—Common Logic (CL): A framework for a family of logic-based languages. International Standard ISO/IEC 24707:2018.

G. S. Makanin. The problem of solvability of equations in a free semi-group, *Math. USSR Sbornik* 32(2): 129–198 (1977).

Ch. Menzel. Knowledge representation, the World Wide Web, and the evolution of logic. *Synthese* 182(2): 269–295 (2011).