Nominal Anti-Unification with an Algorithm to Decide Equivariance

Alexander Baumgartner
Temur Kutsia
Jordi Levy
Mateu Villaret

Anti-Unification Problem

- Given two terms t_1, t_2.
- Find a generalization term t such that t_1, t_2 are instances of t.
- Interesting generalizations are the least general ones (lggs).

Anti-Unification Problem

- Given two terms t_1, t_2.
- Find a generalization term t such that t_1, t_2 are instances of t.
- Interesting generalizations are the least general ones (lggs).

Input terms	$f(a, g(b), b)$ and $f(a, g(c), c)$
Generalization	$f(a, X, Y)$
Lgg	$f(a, g(X), X)$

Nominal Term

- Function symbols (f, g, h)
- Atoms (a, b, c, d)
- Variables (X, Y, Z)

Variables can be instantiated and atoms can be bound.

Nominal Term

Grammar: $t ::= f(t_1, \ldots, t_n) \mid a \mid a.t \mid \pi \cdot X$

- Application
- Atom
- Abstraction
- Suspension

Nominal Term

Grammar: $t ::= f(t_1, \ldots, t_n) \mid a \mid a.t \mid \pi \cdot X$

- Application
- Atom
- Abstraction
- Suspension
  - π is a permutation, represented as a list of swappings.
  - A swapping $(a\ b)$ is a pair of atoms.

Nominal Term

Grammar: $t ::= f(t_1, \ldots, t_n) \mid a \mid a.t \mid \pi \cdot X$

- Application
- Atom
- Abstraction
- Suspension
  - π is a permutation, represented as a list of swappings.
  - A swapping $(a\ b)$ is a pair of atoms.
    - $(a\ b) \cdot a = b; \quad (a\ b) \cdot b = a$
    - $(a\ b) \cdot f(a, b) = f(b, a)$
    - $(a\ b) \cdot a.a = b.b$
    - $(a\ b)(c\ d) \cdot t = (a\ b) \cdot ((c\ d) \cdot t)$

Nominal Anti-Unification – Example

Input terms	$f(a, b, c)$ and $f(a, c, d)$
Generalization	$f(a, X, Y)$
Lgg	$f(a, X, (b\ d)(b\ c) \cdot X)$

Nominal Anti-Unification – Example

Input terms	$f(a, b, c)$ and $f(a, c, d)$
Generalization	$f(a, X, Y)$
Lgg	$f(a, X, (b\ d)(b\ c) \cdot X)$

Input terms	$f(a.a, b)$ and $f(a.X, (a\ b) \cdot X)$
Generalization	$f(a.Y, Z)$
Lgg	$f(a.Y, (a\ b) \cdot Y)$

Freshness

Definition (Freshness constraint)

A *freshness constraint* is a pair of the form $a\#X$ stating that the instantiation of X cannot contain free occurrences of a.

Definition (Freshness context)

A *freshness context* ∇ is a finite set of freshness constraints.

Definition (Term-in-context)

A *term-in-context* is a pair $\langle\nabla, t\rangle$ of a freshness context and a term.

Definition of α-Equivalence

$$\frac{}{\nabla \vdash a \approx a}(\approx\text{-atom}) \qquad \frac{\nabla \vdash t \approx t'}{\nabla \vdash a.t \approx a.t'}(\approx\text{-abs-1})$$

$$\frac{a \neq a' \quad \nabla \vdash t \approx (a\,a') \cdot t' \quad \nabla \vdash a\#t'}{\nabla \vdash a.t \approx a'.t'}(\approx\text{-abs-2})$$

$$\frac{a\#X \in \nabla \text{ for all } a \text{ such that } \pi \cdot a \neq \pi' \cdot a}{\nabla \vdash \pi \cdot X \approx \pi' \cdot X}(\approx\text{-susp.})$$

$$\frac{\nabla \vdash t_1 \approx t_1' \quad \cdots \quad \nabla \vdash t_n \approx t_n'}{\nabla \vdash f(t_1, \ldots t_n) \approx f(t_1', \ldots, t_n')}(\approx\text{-application})$$

where the freshness predicate $\#$ is defined by

$$\frac{a \neq a'}{\nabla \vdash a\#a'}(\#\text{-atom}) \qquad \frac{(\pi^{-1} \cdot a\#X) \in \nabla}{\nabla \vdash a\#\pi \cdot X}(\#\text{-susp.})$$

$$\frac{\nabla \vdash a\#t_1 \quad \cdots \quad \nabla \vdash a\#t_n}{\nabla \vdash a\#f(t_1, \ldots t_n)}(\#\text{-application})$$

$$\frac{}{\nabla \vdash a\#a.t}(\#\text{-abst-1}) \qquad \frac{a \neq a' \quad \nabla \vdash a\#t}{\nabla \vdash a\#a'.t}(\#\text{-abst-2})$$

Substitution Application to ∇

- We say that a substitution σ *respects* a freshness constraint ∇, if for all X, $\mathrm{FA}^{\text{-s}}(X\sigma) \cap \{a \mid a\#X \in \nabla\} = \emptyset$.

Substitution Application to ∇

- We say that a substitution σ *respects* a freshness constraint ∇, if for all X, $\mathrm{FA}^{\text{-s}}(X\sigma) \cap \{a \mid a\#X \in \nabla\} = \emptyset$.
- Rule based system FC works on tuples $F;\ \nabla$ and transforms $F = \{a_1\#t_1, \ldots, a_n\#t_n\}$ into a freshness context ∇, if possible:
  - Del-FC: $\{a\#b\} \uplus F;\ \nabla \Longrightarrow F;\ \nabla$, if $a \neq b$
  - Abs-FC1: $\{a\#a.t\} \uplus F;\ \nabla \Longrightarrow F;\ \nabla$
  - Abs-FC2: $\{a\#b.t\} \uplus F;\ \nabla \Longrightarrow \{a\#t\} \cup F;\ \nabla$, if $a \neq b$
  - Dec-FC: $\{a\#f(t_1, \ldots, t_n)\} \uplus F;\ \nabla \Longrightarrow \{a\#t_1, \ldots, a\#t_n\} \cup F;\ \nabla$
  - Sus-FC: $\{a\#\pi \cdot X\} \uplus F;\ \nabla \Longrightarrow F;\ \{\pi^{-1} \cdot a\#X\} \cup \nabla$

Substitution Application to ∇

- We say that a substitution σ *respects* a freshness constraint ∇, if for all X, $\mathrm{FA}^{\text{-s}}(X\sigma) \cap \{a \mid a\#X \in \nabla\} = \emptyset$.
- Rule based system FC works on tuples $F;\ \nabla$ and transforms $F = \{a_1\#t_1, \ldots, a_n\#t_n\}$ into a freshness context ∇, if possible:
  - Del-FC: $\{a\#b\} \mathbin{\dot\cup} F;\ \nabla \Longrightarrow F;\ \nabla$, if $a \neq b$
  - Abs-FC1: $\{a\#a.t\} \mathbin{\dot\cup} F;\ \nabla \Longrightarrow F;\ \nabla$
  - Abs-FC2: $\{a\#b.t\} \mathbin{\dot\cup} F;\ \nabla \Longrightarrow \{a\#t\} \cup F;\ \nabla$, if $a \neq b$
  - Dec-FC: $\{a\#f(t_1, \ldots, t_n)\} \mathbin{\dot\cup} F;\ \nabla \Longrightarrow \{a\#t_1, \ldots, a\#t_n\} \cup F;\ \nabla$
  - Sus-FC: $\{a\#\pi \cdot X\} \mathbin{\dot\cup} F;\ \nabla \Longrightarrow F;\ \{\pi^{-1} \cdot a\#X\} \cup \nabla$
- Given a freshness context ∇ and a substitution σ, we define $\nabla\sigma = \mathsf{FC}(\{a\#X\sigma \mid a\#X \in \nabla\})$.

Definition of More General

Definition (More general)

We say that a term-in-context $\langle \nabla_1, t_1 \rangle$ is *more general* than a term-in-context $\langle \nabla_2, t_2 \rangle$, written $\langle \nabla_1, t_1 \rangle \preceq \langle \nabla_2, t_2 \rangle$, if there exists a substitution σ, which respects ∇_1, such that $\nabla_1 \sigma \subseteq \nabla_2$ and $\nabla_2 \vdash t_1 \sigma \approx t_2$.

Definition of More General

Definition (More general)

We say that a term-in-context $\langle \nabla_1, t_1 \rangle$ is *more general* than a term-in-context $\langle \nabla_2, t_2 \rangle$, written $\langle \nabla_1, t_1 \rangle \preceq \langle \nabla_2, t_2 \rangle$, if there exists a substitution σ, which respects ∇_1, such that $\nabla_1 \sigma \subseteq \nabla_2$ and $\nabla_2 \vdash t_1 \sigma \approx t_2$.

- $\langle \{a \# X\}, f(a) \rangle \simeq \langle \emptyset, f(a) \rangle$

Definition of More General

Definition (More general)

We say that a term-in-context $\langle \nabla_1, t_1 \rangle$ is *more general* than a term-in-context $\langle \nabla_2, t_2 \rangle$, written $\langle \nabla_1, t_1 \rangle \preceq \langle \nabla_2, t_2 \rangle$, if there exists a substitution σ, which respects ∇_1, such that $\nabla_1 \sigma \subseteq \nabla_2$ and $\nabla_2 \vdash t_1 \sigma \approx t_2$.

- $\langle \{a \# X\}, f(a) \rangle \simeq \langle \emptyset, f(a) \rangle$
- $\langle \emptyset, f(X) \rangle \preceq \langle \{a \# X\}, f(X) \rangle$

Definition of More General

Definition (More general)

We say that a term-in-context $\langle\nabla_1, t_1\rangle$ is *more general* than a term-in-context $\langle\nabla_2, t_2\rangle$, written $\langle\nabla_1, t_1\rangle \preceq \langle\nabla_2, t_2\rangle$, if there exists a substitution σ, which respects ∇_1, such that $\nabla_1\sigma \subseteq \nabla_2$ and $\nabla_2 \vdash t_1\sigma \approx t_2$.

- $\langle\{a\#X\}, f(a)\rangle \simeq \langle\emptyset, f(a)\rangle$
- $\langle\emptyset, f(X)\rangle \preceq \langle\{a\#X\}, f(X)\rangle$
- $\langle\emptyset, f(X)\rangle \preceq \langle\{a\#Y\}, f(Y)\rangle$

Definition of More General

Definition (More general)

We say that a term-in-context $\langle \nabla_1, t_1 \rangle$ is *more general* than a term-in-context $\langle \nabla_2, t_2 \rangle$, written $\langle \nabla_1, t_1 \rangle \preceq \langle \nabla_2, t_2 \rangle$, if there exists a substitution σ, which respects ∇_1, such that $\nabla_1 \sigma \subseteq \nabla_2$ and $\nabla_2 \vdash t_1 \sigma \approx t_2$.

- $\langle \{a\#X\}, f(a) \rangle \simeq \langle \emptyset, f(a) \rangle$
- $\langle \emptyset, f(X) \rangle \preceq \langle \{a\#X\}, f(X) \rangle$
- $\langle \emptyset, f(X) \rangle \preceq \langle \{a\#Y\}, f(Y) \rangle$
- $\langle \{a\#X\}, f(X) \rangle \not\preceq \langle \emptyset, f(Y) \rangle$

Definition of More General

Definition (More general)

We say that a term-in-context $\langle \nabla_1, t_1 \rangle$ is *more general* than a term-in-context $\langle \nabla_2, t_2 \rangle$, written $\langle \nabla_1, t_1 \rangle \preceq \langle \nabla_2, t_2 \rangle$, if there exists a substitution σ, which respects ∇_1, such that $\nabla_1 \sigma \subseteq \nabla_2$ and $\nabla_2 \vdash t_1 \sigma \approx t_2$.

- $\langle \{a\#X\}, f(a) \rangle \simeq \langle \emptyset, f(a) \rangle$
- $\langle \emptyset, f(X) \rangle \preceq \langle \{a\#X\}, f(X) \rangle$
- $\langle \emptyset, f(X) \rangle \preceq \langle \{a\#Y\}, f(Y) \rangle$
- $\langle \{a\#X\}, f(X) \rangle \not\preceq \langle \emptyset, f(Y) \rangle$
- $\langle \{a\#X\}, f(X) \rangle \not\preceq \langle \{a\#X\}, f(a) \rangle$

Definition of More General

Definition (More general)

We say that a term-in-context $\langle \nabla_1, t_1 \rangle$ is *more general* than a term-in-context $\langle \nabla_2, t_2 \rangle$, written $\langle \nabla_1, t_1 \rangle \preceq \langle \nabla_2, t_2 \rangle$, if there exists a substitution σ, which respects ∇_1, such that $\nabla_1 \sigma \subseteq \nabla_2$ and $\nabla_2 \vdash t_1 \sigma \approx t_2$.

- $\langle \{a\#X\}, f(a) \rangle \simeq \langle \emptyset, f(a) \rangle$
- $\langle \emptyset, f(X) \rangle \preceq \langle \{a\#X\}, f(X) \rangle$
- $\langle \emptyset, f(X) \rangle \preceq \langle \{a\#Y\}, f(Y) \rangle$
- $\langle \{a\#X\}, f(X) \rangle \not\preceq \langle \emptyset, f(Y) \rangle$
- $\langle \{a\#X\}, f(X) \rangle \not\preceq \langle \{a\#X\}, f(a) \rangle$
- $\langle \{b\#X\}, (a\, b) \cdot X \rangle \simeq \langle \{c\#X\}, (a\, c) \cdot X \rangle$

Input / Output

- **Given:** Two nominal terms t and s of the same sort, a freshness context ∇, and a finite set of atoms A such that t, s, and ∇ are based on A.
- **Find:** A nominal term r and a freshness context Γ, such that the term-in-context $\langle \Gamma, r \rangle$ is an A-based least general generalization of the terms-in-context $\langle \nabla, t \rangle$ and $\langle \nabla, s \rangle$.

Input terms	$f(a, b)$ and $f(b, c)$
Input context	$\emptyset$
Atoms A	$\{a, b, c, d\}$
No lgg	$\langle \emptyset, f(Y, (a\, b)(b\, c) \cdot Y) \rangle$
A-based lgg	$\langle \{c\#Y, d\#Y\}, f(Y, (a\, b)(b\, c) \cdot Y) \rangle$

The Anti-Unification Algorithm

- Rule-based formulation:
  - 4 transformation rules.
- Works on tuples P; S; ∇; Atoms; Γ; σ:
  - P and S are sets of AUEs of the form $X : t \triangleq s$;
  - Global freshness context ∇;
  - Global set Atoms is a finite set of atoms;
  - Computed freshness context Γ;
  - Computed substitution σ.

Illustration of Anti-Unification Algorithm

We illustrate the algorithm on the input:
$f(a, b)$ and $f(b, c)$, $\nabla = \emptyset$, Atoms $= \{a, b, c, d\}$:

$$\{X : f(a, b) \triangleq f(b, c)\}; \emptyset; \emptyset; \varepsilon$$

Illustration of Anti-Unification Algorithm

We illustrate the algorithm on the input:
$f(a,b)$ and $f(b,c)$, $\nabla = \emptyset$, Atoms $= \{a,b,c,d\}$:

$$\{X : f(a,b) \triangleq f(b,c)\}; \emptyset; \emptyset; \varepsilon$$
$$\Longrightarrow_{\mathsf{Dec}} \{Y : a \triangleq b, Z : b \triangleq c\}; \emptyset; \emptyset; \{X \mapsto f(Y,Z)\}$$

Illustration of Anti-Unification Algorithm

We illustrate the algorithm on the input:
$f(a, b)$ and $f(b, c)$, $\nabla = \emptyset$, Atoms $= \{a, b, c, d\}$:

$$\begin{aligned}
& \{X : f(a, b) \triangleq f(b, c)\}; \emptyset; \emptyset; \varepsilon \\
\Longrightarrow_{\mathsf{Dec}}\ & \{Y : a \triangleq b, Z : b \triangleq c\}; \emptyset; \emptyset; \{X \mapsto f(Y, Z)\} \\
\Longrightarrow_{\mathsf{Sol}}\ & \{Z : b \triangleq c\}; \{Y : a \triangleq b\}; \\
& \{c\#Y, d\#Y\};\ \{X \mapsto f(Y, Z)\}
\end{aligned}$$

Illustration of Anti-Unification Algorithm

We illustrate the algorithm on the input:
$f(a,b)$ and $f(b,c)$, $\nabla = \emptyset$, Atoms $= \{a,b,c,d\}$:

$$\{X : f(a,b) \triangleq f(b,c)\}; \emptyset; \emptyset; \varepsilon$$

$$\Longrightarrow_{\mathsf{Dec}} \{Y : a \triangleq b, Z : b \triangleq c\}; \emptyset; \emptyset; \{X \mapsto f(Y,Z)\}$$

$$\Longrightarrow_{\mathsf{Sol}} \{Z : b \triangleq c\}; \{Y : a \triangleq b\};$$
$$\{c\#Y, d\#Y\};\ \{X \mapsto f(Y,Z)\}$$

$$\Longrightarrow_{\mathsf{Sol}} \emptyset; \{Y : a \triangleq b, Z : b \triangleq c\};$$
$$\{c\#Y, d\#Y, a\#Z, d\#Z\};\ \{X \mapsto f(Y,Z)\}$$

Illustration of Anti-Unification Algorithm

We illustrate the algorithm on the input:
$f(a,b)$ and $f(b,c)$, $\nabla = \emptyset$, Atoms $= \{a,b,c,d\}$:

$$
\begin{aligned}
& \{X : f(a,b) \triangleq f(b,c)\}; \emptyset; \emptyset; \varepsilon \\
\Longrightarrow_{\mathsf{Dec}}\ & \{Y : a \triangleq b, Z : b \triangleq c\}; \emptyset; \emptyset; \{X \mapsto f(Y,Z)\} \\
\Longrightarrow_{\mathsf{Sol}}\ & \{Z : b \triangleq c\}; \{Y : a \triangleq b\}; \\
& \{c\#Y, d\#Y\};\ \{X \mapsto f(Y,Z)\} \\
\Longrightarrow_{\mathsf{Sol}}\ & \emptyset; \{Y : a \triangleq b, Z : b \triangleq c\}; \\
& \{c\#Y, d\#Y, a\#Z, d\#Z\};\ \{X \mapsto f(Y,Z)\} \\
\Longrightarrow_{\mathsf{Mer}}\ & \emptyset; \{Y : a \triangleq b\}; \\
& \{c\#Y, d\#Y\}; \{X \mapsto f(Y, (a\,b)(b\,c) \cdot Y)\}
\end{aligned}
$$

Illustration of Anti-Unification Algorithm

We illustrate the algorithm on the input:
$f(a,b)$ and $f(b,c)$, $\nabla = \emptyset$, Atoms $= \{a,b,c,d\}$:

$$
\begin{aligned}
& \{X : f(a,b) \triangleq f(b,c)\}; \emptyset; \emptyset; \varepsilon \\
\Longrightarrow_{\mathsf{Dec}}\ & \{Y : a \triangleq b, Z : b \triangleq c\}; \emptyset; \emptyset; \{X \mapsto f(Y,Z)\} \\
\Longrightarrow_{\mathsf{Sol}}\ & \{Z : b \triangleq c\}; \{Y : a \triangleq b\}; \\
& \{c\#Y, d\#Y\};\ \{X \mapsto f(Y,Z)\} \\
\Longrightarrow_{\mathsf{Sol}}\ & \emptyset; \{Y : a \triangleq b, Z : b \triangleq c\}; \\
& \{c\#Y, d\#Y, a\#Z, d\#Z\};\ \{X \mapsto f(Y,Z)\} \\
\Longrightarrow_{\mathsf{Mer}}\ & \emptyset; \{Y : a \triangleq b\}; \\
& \{c\#Y, d\#Y\}; \{X \mapsto f(Y, (a\,b)(b\,c) \cdot Y)\}
\end{aligned}
$$

The Atoms-based lgg is $\langle \{c\#Y, d\#Y\}, f(Y, (a\,b)(b\,c) \cdot Y \rangle$.

The Equivariance Problem

- Mer: **Merging**

  $P;\ \{X : t_1 \triangleq s_1, Y : t_2 \triangleq s_2\} \mathbin{\dot\cup} S;\ \Gamma;\ \sigma \Longrightarrow$
  $\quad P;\ \{X : t_1 \triangleq s_1\} \cup S;\ \Gamma\{Y \mapsto \pi \cdot X\};\ \sigma\{Y \mapsto \pi \cdot X\},$

  where π : Atoms $\longrightarrow$ Atoms is a permutation such that $\nabla \vdash \pi \cdot t_1 \approx t_2$, and $\nabla \vdash \pi \cdot s_1 \approx s_2$.

The Equivariance Problem

- Mer: **Merging**
  $P;\ \{X : t_1 \triangleq s_1, Y : t_2 \triangleq s_2\} \cup S;\ \Gamma;\ \sigma \Longrightarrow$
  $P;\ \{X : t_1 \triangleq s_1\} \cup S;\ \Gamma\{Y \mapsto \pi \cdot X\};\ \sigma\{Y \mapsto \pi \cdot X\},$
  where π : Atoms $\longrightarrow$ Atoms is a permutation such that $\nabla \vdash \pi \cdot t_1 \approx t_2$, and $\nabla \vdash \pi \cdot s_1 \approx s_2$.

- **Given:** Pairs of nominal terms (t_1, t_2) and (s_1, s_2) of the same sort, a freshness context ∇, and a finite set of atoms A.
- **Find:** An A-based permutation π, such that $\nabla \vdash \pi \cdot t_1 \approx t_2$, and $\nabla \vdash \pi \cdot s_1 \approx s_2$, if it exists.

Deciding Equivariance

- Rule-based algorithm $\mathfrak{E}$ works in two phases:
  - Simplification phase.
  - Permutation computation.

Deciding Equivariance

- Rule-based algorithm $\mathfrak{E}$ works in two phases:
  - Simplification phase.
  - Permutation computation.
- Tuples of the form E; ∇; A; π:
  - E is a set of equivariance equations of the form $\langle t_1, \rho_1 \rangle \approx \langle t_2, \rho_2 \rangle$.
    - t_1, t_2 are nominal terms;
    - ρ_1, ρ_2 are permutations.

Deciding Equivariance

- Rule-based algorithm $\mathfrak{E}$ works in two phases:
  - Simplification phase.
  - Permutation computation.
- Tuples of the form $E; \nabla; A; \pi$:
  - E is a set of equivariance equations of the form $\langle t_1, \rho_1 \rangle \approx \langle t_2, \rho_2 \rangle$.
    - t_1, t_2 are nominal terms;
    - ρ_1, ρ_2 are permutations.
  - ∇ is a freshness context.
  - A is a finite set of atoms.
  - π is a A-based permutation.

Deciding Equivariance

- Rule-based algorithm $\mathfrak{E}$ works in two phases:
  - Simplification phase.
  - Permutation computation.
- Tuples of the form $E; \nabla; A; \pi$:
  - E is a set of equivariance equations of the form $\langle t_1, \rho_1 \rangle \approx \langle t_2, \rho_2 \rangle$.
    - t_1, t_2 are nominal terms;
    - ρ_1, ρ_2 are permutations.
  - ∇ is a freshness context.
  - A is a finite set of atoms.
  - π is a A-based permutation.
- Two final states:
  - Success state: π holds the computed permutation.
  - Failure state $\bot$.

Simplification Phase of $\mathfrak{E}$

- Dec-E: **Decomposition**

$$\{\langle f(t_1,\dots,t_m),\rho_1\rangle \approx \langle f(s_1,\dots,s_m),\rho_2\rangle\} \mathbin{\dot\cup} E;\ \nabla;\ A;\ Id \Longrightarrow$$
$$\{\langle t_1,\rho_1\rangle \approx \langle s_1,\rho_2\rangle,\dots,\langle t_m,\rho_1\rangle \approx \langle s_m,\rho_2\rangle\} \cup E; \nabla;\ A;\ Id.$$

Simplification Phase of $\mathfrak{E}$

- Dec-E: **Decomposition**
  $$\{\langle f(t_1,\ldots,t_m),\rho_1\rangle \approx \langle f(s_1,\ldots,s_m),\rho_2\rangle\} \mathbin{\dot\cup} E;\ \nabla;\ A;\ Id \Longrightarrow$$
  $$\{\langle t_1,\rho_1\rangle \approx \langle s_1,\rho_2\rangle,\ldots,\langle t_m,\rho_1\rangle \approx \langle s_m,\rho_2\rangle\} \cup E; \nabla;\ A;\ Id.$$
- Alp-E: **Alpha Equivalence**
  $$\{\langle a.t,\rho_1\rangle \approx \langle b.s,\rho_2\rangle\} \mathbin{\dot\cup} E;\ \nabla;\ A;\ Id \Longrightarrow$$
  $$\{\langle t,\rho_3\rangle \approx \langle s,\rho_4\rangle\} \cup E;\ \nabla \cup \{\acute{c}\#X \mid X \in \mathrm{Vars}(t,s)\};\ A;\ Id,$$
  where $\acute{c}$ is a fresh atom and $a, b, \acute{c}$ are of the same sort.
  $\rho_3 = (\rho_1{\cdot}a\ \ \acute{c})\rho_1$ and $\rho_4 = (\rho_2{\cdot}b\ \ \acute{c})\rho_2$.

Permutation Computation of $\mathfrak{E}$

- Rem-E: **Remove**
  $\{\langle a, \rho_1\rangle \approx \langle b, \rho_2\rangle \cup E;\ \nabla;\ A;\ \pi \Longrightarrow E;\ \nabla;\ A \setminus \{\rho_2 \cdot b\};\ \pi,$
  where $\pi\rho_1 \cdot a = \rho_2 \cdot b$.

Permutation Computation of $\mathfrak{E}$

- Rem-E: **Remove**
  $\{\langle a, \rho_1\rangle \approx \langle b, \rho_2\rangle \uplus E;\ \nabla;\ A;\ \pi \Longrightarrow E;\ \nabla;\ A \setminus \{\rho_2 \cdot b\};\ \pi,$
  where $\pi\rho_1 \cdot a = \rho_2 \cdot b$.
- Sol-E: **Solve**
  $\{\langle a, \rho_1\rangle \approx \langle b, \rho_2\rangle\} \uplus E;\ \nabla;\ A;\ \pi \Longrightarrow E;\ \nabla;\ A \setminus \{d\};\ (c\, d)\pi,$
  where $c, d \in A$, $c = \pi\rho_1 \cdot a$, $d = \rho_2 \cdot b$, and $c \neq d$.

Permutation Computation of $\mathfrak{E}$

- Rem-E: **Remove**
  $\{\langle a, \rho_1 \rangle \approx \langle b, \rho_2 \rangle \cup E;\ \nabla;\ A;\ \pi \Longrightarrow E;\ \nabla;\ A \setminus \{\rho_2 \cdot b\};\ \pi,$
  where $\pi\rho_1 \cdot a = \rho_2 \cdot b$.
- Sol-E: **Solve**
  $\{\langle a, \rho_1 \rangle \approx \langle b, \rho_2 \rangle\} \cup E;\ \nabla;\ A;\ \pi \Longrightarrow E;\ \nabla;\ A \setminus \{d\};\ (c\ d)\pi,$
  where $c, d \in A$, $c = \pi\rho_1 \cdot a$, $d = \rho_2 \cdot b$, and $c \neq d$.
- Sus-E: **Suspension**
  $\{\langle \pi_1 \cdot X, \rho_1 \rangle \approx \langle \pi_2 \cdot X, \rho_2 \rangle\} \cup E;\ \nabla;\ A;\ \pi \Longrightarrow$
  $\{\langle \pi^{-1} \cdot a, Id \rangle \approx \langle b, Id \rangle \mid (a, b) \in \mathsf{dp}(\pi\rho_1\pi_1, \rho_2\pi_2, \nabla, X)\} \cup E;$
  $\nabla;\ A \cap (\mathsf{da}(\pi\rho_1\pi_1, \rho_2\pi_2, \nabla, X) \cup \{\pi\rho_1\pi_1 \cdot a \mid a\#X \in \nabla\});\ \pi.$

Demonstration of $\mathfrak{E}$

Consider the term-pairs (a, a) and $(a.(a\ b)(c\ d) \cdot X,\ b.X)$.
Atoms $= \{a, b, c, d\}$, and $\nabla = \{a\#X\}$.

$$\{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle a.(a\ b)(c\ d) \cdot X, Id\rangle \approx \langle b.X, Id\rangle\};$$
$$\{a\#X\};\ \{a, b, c, d\};\ Id$$

Demonstration of $\mathfrak{E}$

Consider the term-pairs (a, a) and $(a.(a\ b)(c\ d) \cdot X,\ b.X)$.
Atoms $= \{a, b, c, d\}$, and $\nabla = \{a\#X\}$.

$$\{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle a.(a\ b)(c\ d) \cdot X, Id\rangle \approx \langle b.X, Id\rangle\};$$
$$\{a\#X\};\ \{a, b, c, d\};\ Id$$
$$\Longrightarrow_{\text{Alp-E}} \{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle (a\ b)(c\ d) \cdot X, (a\ \acute{e})\rangle \approx \langle X, (b\ \acute{e})\rangle\};$$
$$\{a\#X, \acute{e}\#X\};\ \{a, b, c, d\};\ Id$$

Demonstration of $\mathfrak{E}$

Consider the term-pairs (a, a) and $(a.(a\ b)(c\ d) \cdot X,\ b.X)$.
Atoms $= \{a, b, c, d\}$, and $\nabla = \{a\#X\}$.

$$\{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle a.(a\ b)(c\ d) \cdot X, Id\rangle \approx \langle b.X, Id\rangle\};$$
$$\{a\#X\};\ \{a, b, c, d\};\ Id$$

$$\Longrightarrow_{\text{Alp-E}} \{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle (a\ b)(c\ d) \cdot X, (a\ é)\rangle \approx \langle X, (b\ é)\rangle\};$$
$$\{a\#X, é\#X\};\ \{a, b, c, d\};\ Id$$

$$\Longrightarrow_{\text{Rem-E}} \{\langle (a\ b)(c\ d) \cdot X, (a\ é)\rangle \approx \langle X, (b\ é)\rangle\};$$
$$\{a\#X, é\#X\};\ \{b, c, d\};\ Id$$

Demonstration of $\mathfrak{E}$

Consider the term-pairs (a, a) and $(a.(a\ b)(c\ d) \cdot X,\ b.X)$.
Atoms $= \{a, b, c, d\}$, and $\nabla = \{a\#X\}$.

$$\{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle a.(a\ b)(c\ d)\cdot X, Id\rangle \approx \langle b.X, Id\rangle\};$$
$$\{a\#X\};\ \{a, b, c, d\};\ Id$$
$$\Longrightarrow_{\text{Alp-E}} \{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle (a\ b)(c\ d)\cdot X, (a\ \acute{e})\rangle \approx \langle X, (b\ \acute{e})\rangle\};$$
$$\{a\#X, \acute{e}\#X\};\ \{a, b, c, d\};\ Id$$

$$\Longrightarrow_{\text{Rem-E}} \{\langle (a\ b)(c\ d)\cdot X, (a\ \acute{e})\rangle \approx \langle X, (b\ \acute{e})\rangle\};$$
$$\{a\#X, \acute{e}\#X\};\ \{b, c, d\};\ Id$$
$$\Longrightarrow_{\text{Sus-E}} \{\langle c, Id\rangle \approx \langle d, Id\rangle, \langle d, Id\rangle \approx \langle c, Id\rangle\};$$
$$\{a\#X, \acute{e}\#X\};\ \{b, c, d\};\ Id$$

Demonstration of $\mathfrak{E}$

Consider the term-pairs (a, a) and $(a.(a\ b)(c\ d) \cdot X,\ b.X)$.
Atoms $= \{a, b, c, d\}$, and $\nabla = \{a\#X\}$.

$$\{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle a.(a\ b)(c\ d) \cdot X, Id\rangle \approx \langle b.X, Id\rangle\};$$
$$\{a\#X\};\ \{a, b, c, d\};\ Id$$
$$\Longrightarrow_{\text{Alp-E}} \{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle (a\ b)(c\ d) \cdot X, (a\ \acute{e})\rangle \approx \langle X, (b\ \acute{e})\rangle\};$$
$$\{a\#X, \acute{e}\#X\};\ \{a, b, c, d\};\ Id$$

$$\Longrightarrow_{\text{Rem-E}} \{\langle (a\ b)(c\ d) \cdot X, (a\ \acute{e})\rangle \approx \langle X, (b\ \acute{e})\rangle\};$$
$$\{a\#X, \acute{e}\#X\};\ \{b, c, d\};\ Id$$
$$\Longrightarrow_{\text{Sus-E}} \{\langle c, Id\rangle \approx \langle d, Id\rangle, \langle d, Id\rangle \approx \langle c, Id\rangle\};$$
$$\{a\#X, \acute{e}\#X\};\ \{b, c, d\};\ Id$$
$$\Longrightarrow_{\text{Sol-E}} \{\langle d, Id\rangle \approx \langle c, Id\rangle\};\ \{a\#X, \acute{e}\#X\};\ \{b, c\};\ (c\ d)$$

Demonstration of $\mathfrak{E}$

Consider the term-pairs (a, a) and $(a.(a\ b)(c\ d) \cdot X,\ b.X)$.
Atoms $= \{a, b, c, d\}$, and $\nabla = \{a\#X\}$.

$$\{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle a.(a\ b)(c\ d) \cdot X, Id\rangle \approx \langle b.X, Id\rangle\};$$
$$\{a\#X\};\ \{a, b, c, d\};\ Id$$
$$\Longrightarrow_{\text{Alp-E}} \{\langle a, Id\rangle \approx \langle a, Id\rangle, \langle (a\ b)(c\ d) \cdot X, (a\ é)\rangle \approx \langle X, (b\ é)\rangle\};$$
$$\{a\#X, é\#X\};\ \{a, b, c, d\};\ Id$$

$$\Longrightarrow_{\text{Rem-E}} \{\langle (a\ b)(c\ d) \cdot X, (a\ é)\rangle \approx \langle X, (b\ é)\rangle\};$$
$$\{a\#X, é\#X\};\ \{b, c, d\};\ Id$$
$$\Longrightarrow_{\text{Sus-E}} \{\langle c, Id\rangle \approx \langle d, Id\rangle, \langle d, Id\rangle \approx \langle c, Id\rangle\};$$
$$\{a\#X, é\#X\};\ \{b, c, d\};\ Id$$
$$\Longrightarrow_{\text{Sol-E}} \{\langle d, Id\rangle \approx \langle c, Id\rangle\};\ \{a\#X, é\#X\};\ \{b, c\};\ (c\ d)$$
$$\Longrightarrow_{\text{Rem-E}} \emptyset;\ \{a\#X, é\#X\};\ \{b\};\ (c\ d).$$

Demonstration

http://www.risc.jku.at/projects/stout/software/nequiv.php
http://www.risc.jku.at/projects/stout/software/nau.php