Nominal Disunification

Seminários Teoria da Computação - UnB

Deivid Vale, Advisor: Daniele Nantes

November 9, 2018

Department of Mathematics - University of Brasilia

Table of contents

1. Nominal Terms and Syntax
2. Freshness and Equality
3. Nominal Unification
4. Nominal Disunification

Nominal Terms and Syntax

Nominal Terms

Consider countable disjoint sets of *meta-variables* (or just *variables*) $\mathbb{X} = \{X, Y, Z, \dots\}$ and atoms $\mathbb{A} = \{a, b, c, d, \dots\}$.

Nominal Terms

Definition (Nominal Terms)

The set $T(\Sigma, \mathbb{A}, \mathbb{X})$ of all nominal terms is inductively generated by the following grammar:

$$s, t, u, v \quad ::= \quad a \;\mid\; \pi \cdot X \;\mid\; [a]\,t \;\mid\; f(t_1, \ldots, t_n)$$

Terms are called, respectively, **atoms**, **moderated variable**, **abstractions**, and **function application**.

Permutation Action

Definition

The **object-level** action of a permutation π on a term t is defined by induction on the structure of t, as follows:

$$\pi \cdot a \equiv \pi(a) \quad \pi \cdot (\gamma \cdot X) \equiv (\pi \cdot \gamma) \cdot X \quad \pi \cdot [a]\, t \equiv [\pi \cdot a]\, (\pi \cdot t)$$
$$\pi \cdot f(t_1, \ldots, t_n) \equiv f(\pi \cdot t_1, \ldots, \pi \cdot t_n)$$

Substitutions

Definition

The **meta-level** action of a substitution σ on a term t, denoted as $t\sigma$, is inductively defined by:

$$a\sigma \equiv a \quad (\pi \cdot X)\sigma \equiv \pi \cdot (X\sigma) \quad ([a]\,t)\sigma \equiv [a]\,(t\sigma)$$

$$f(t_1, \ldots, t_n)\sigma \equiv f(t_1\sigma, \ldots, t_n\sigma)$$

Permutation and Substitution Action

Lemma (Commutation Lemma)

$\pi \cdot (t\sigma) \equiv (\pi \cdot t)\sigma.$

Freshness and Equality

Freshness Constraints and Contexts

Definition (Contexts)

A *freshness constraint* is a pair $a\#t$ of an atom a and a term t. Call freshness constraints $a\#a$ and $a\#X$ primitive and write Δ and ∇ for finite sets of primitive freshness constraints and call them **freshness contexts**.

We say a context Δ is consistent iff it does not contain any freshness constraint of the form $a\#a$.

Derivation Rules for Freshness

$$\frac{}{a\#b}\ (\#ab) \qquad \frac{\pi^{-1}(a)\#X}{a\#\pi\cdot X}\ (\#X)\ (\pi \neq \mathtt{id})$$

$$\frac{}{a\#[a]\,t}\ (\#a)$$

$$\frac{a\#t}{a\#[b]\,t}\ (\#b) \qquad \frac{a\#t_1 \cdots a\#t_n}{a\#f(t_1,\ldots,t_n)}\ (\#f)$$

Table 1: Derivation rules for freshness

Derivation Rules for α-equivalence

$$\frac{}{a \approx_\alpha a}\ (\#ab) \qquad \frac{\Delta \vdash \mathtt{ds}(\pi, \gamma)\#X}{\pi \cdot X \approx_\alpha \gamma \cdot X}\ (\mathrm{Ds})$$

$$\frac{t_1 \approx_\alpha u_1 \cdots t_n \approx_\alpha u_n}{f(t_1, \ldots, t_n) \approx_\alpha f(u_1, \ldots, u_n)}\ (\mathrm{F})$$

$$\frac{t \approx_\alpha u}{[a]\, t \approx_\alpha [a]\, u}\ (\text{Abs-a}) \qquad \frac{(a\ \ b) \cdot t \approx_\alpha u \quad \Delta \vdash b\#t}{[a]\, t \approx_\alpha [b]\, u}\ (\text{Abs-b})$$

Table 2: Derivation rules for α-equivalence

Judgment Forms

Nominal algebra ([3]) has two judgment forms:

1. A *freshness judgment form* $\Delta \vdash a\#t$.
2. An *equality judgment form* $\Delta \vdash t \approx_\alpha u$.

Derivability

We define a *notion of derivability* by the natural deduction rules given in Tables 1 and 2.

1. We say $\Delta \vdash a\#t$ is a valid judgment when there exists a derivation of $a\#t$ using the elements of Δ as assumptions.
2. We also say $\Delta \vdash t \approx_\alpha u$ is a valid judgment when there exists a derivation of $t \approx_\alpha u$ using elements of Δ as assumptions.

Nominal Unification

Unification Problems

Definition

A *unification problem* P is a finite set

$$P = \left\langle a_1 \overset{?}{\#} u_1, \cdots, a_n \overset{?}{\#} u_j \mid\mid s_1 \overset{?}{\approx}_\alpha t_1, \cdots, s_n \overset{?}{\approx}_\alpha t_n \right\rangle$$

of *freshness problems* $a_i \overset{?}{\#} u_i$ and *equational problems* $s_k \overset{?}{\approx}_\alpha t_k$.

Simplification Rules on Problems

$$
\begin{aligned}
a\#b, P &\Longrightarrow P \\
a\#\pi \cdot X, P &\Longrightarrow \pi^{-1}(a)\#X, P \quad \pi \neq \mathtt{id} \\
a\#[a]\,t, P &\Longrightarrow P \\
a\#[b]\,t, P &\Longrightarrow a\#t, P \\
a\#f(t_1, \cdots, f_n), P &\Longrightarrow a\#t_1, \cdots, a\#t_n, P
\end{aligned}
$$

Table 3: Simplification rules for freshness

Simplification Rules on Problems

$$
\begin{aligned}
a \approx_\alpha a, P &\implies P \\
\pi \cdot X \approx_\alpha \gamma \cdot X, P &\implies \mathtt{ds}(\pi, \gamma) \# X, P \\
f(s_1, \cdots, s_n) \approx_\alpha f(t_1, \cdots, t_n), P &\implies s_1 \approx_\alpha t_1, \cdots, s_n \approx_\alpha t_n, P \\
[a]\, t \approx_\alpha [a]\, u, P &\implies t \approx_\alpha u, P \\
[b]\, l \approx_\alpha [a]\, r, P &\implies (a \;\; b) \cdot l \approx_\alpha r, a \# l, P
\end{aligned}
$$

Table 4: Simplification rules for α-equivalence

Properties of Simplification Rules

Lemma

The relation $\Longrightarrow$ *is confluent and strongly normalizing.*

Unification Normal Forms

We write the *unique* normal form of an unification problem P as $\langle P \rangle \downarrow$. So the problem P can be written as

$$\langle P \rangle \downarrow = \left\langle \Delta \mid\mid s'_1 \overset{?}{\approx}_\alpha t'_1, \cdots, s'_l \overset{?}{\approx}_\alpha t'_l \right\rangle$$

We call Δ an **initial context** and the equational predicates $s_i \overset{?}{\approx}_\alpha t_i$ the equations that remains to be solved.

From now on we always consider problems in normal forms.

Unification Problems

A solution of an unification problem P is a pair (Γ, σ) of a consistent context Γ and a substitution σ such that:

1. $\Gamma \vdash \Delta\sigma$.
2. $\Gamma \vdash s_i\sigma \overset{?}{\approx}_\alpha t_i\sigma$, for all $1 \leq i \leq n$.

We write $\mathcal{U}(Pr)$ for the set of all solutions of P.

An Algorithm for Nominal Unification

We now derive an *algorithm* for nominal unification by enriching the *simplification rules* with the two following additional rules: (as done in [2]).

$$\pi \cdot X \overset{?}{\approx}_\alpha u, P \xrightarrow{X \mapsto \pi^{-1} \cdot u} P[X \mapsto \pi^{-1} \cdot u] \quad (X \notin \mathtt{vars}(u))$$
$$u \overset{?}{\approx}_\alpha \pi \cdot X, P \xrightarrow{X \mapsto \pi^{-1} \cdot u} P[X \mapsto \pi^{-1} \cdot u] \quad (X \notin \mathtt{vars}(u))$$

We denote this algorithm as $\mathtt{Unif}\,(P)$ or if we need to be more specific $\mathtt{Unif}\,(\Delta, s_1 \approx_\alpha t_1, \cdots, s_n \approx_\alpha t_n)$.

Instantiation

We order solutions by the *instantiation order:* We say (Γ_2, σ_2) is an *instance* of (Γ_1, σ_1) iff there exists a substitution δ such that for all variables X the following conditions hold:

1. $\Gamma_2 \vdash X\sigma_2 \approx_\alpha X\sigma_1\delta$.
2. $\Gamma_2 \vdash \Gamma_1\delta$.

We denote this fact by writing:

$$(\Gamma_1, \sigma_1) \leq_\delta (\Gamma_2, \sigma_2)$$

Instantiation of Solutions

Lemma

The instantiation order is a partial order on $\mathcal{U}(P)$. *([2])*

Nominal Disunification

Disunification Problems

Now we discuss an extension to the nominal syntax of the work on First Order Disunification done by W. Buntine and H. Bürckert in [1].

Disunification Problems

Definition

A *nominal disunification problem* P is an ordered pair $P = \langle E \mid\mid D \rangle$ where E is an equational problem and D is a *disequational problem* as follows:

$$E = \left\langle \Delta \mid\mid s_1 \overset{?}{\approx}_\alpha t_1, \cdots, s_n \overset{?}{\approx}_\alpha t_n \right\rangle$$

$$D = \left\langle \nabla \mid\mid p_1 \overset{?}{\not\approx}_\alpha q_1, \cdots, p_m \overset{?}{\not\approx}_\alpha q_m \right\rangle$$

and Δ, ∇ are consistent contexts. We call them the **initial contexts** of the problem P.

Remark

We consider Δ, ∇ as the initial freshness constraints we impose on equations and disequations of P, respectively.

Example

$$P_1 = \left\langle \emptyset, X \stackrel{?}{\approx}_\alpha Y \mid\mid \emptyset, X \stackrel{?}{\not\approx}_\alpha a \right\rangle$$

$$P_2 = \left\langle b\#Z, (b\ \ a) \cdot X \stackrel{?}{\approx}_\alpha Y \mid\mid \emptyset, [a]\,X \stackrel{?}{\not\approx}_\alpha [b]\,Y \right\rangle$$

Solutions to Disunification

Definition

A solution to a disunification problem $P = \langle E \mid\mid D \rangle$ is a pair (Γ, σ) of a consistent context Γ and a substitution σ satisfying the following conditions:

1. (Γ, σ) is a solution of E.
2. (Γ, σ) satisfies the disequations in D. That is:
    2.1 $\Gamma \not\vdash \Delta\sigma$, or
    2.2 $\Gamma \not\vdash p \approx_\alpha q$, for all $p \not\approx_\alpha q$ in D.

Restricted Instantiation

Definition

A "pair with exception" is a pair $\langle \Gamma \mid\mid \sigma \rangle - \Psi$ of a **pair** $\langle \Gamma \mid\mid \sigma \rangle$ and an indexed family of **pairs** $\Psi = \{(\nabla_I, \psi_I) \mid I \in I\}$

.

Restricted Instantiation

Definition

We say a **pair** (Γ, σ) is an instance of a family Ψ iff each instance of (Γ, σ) is a instance of some $(\nabla_I, \psi_I) \in \Psi$.

We then write $\Psi \leq (\Gamma, \sigma)$.

Restricted Instantiation

Definition

A pair (Δ, λ) is an instance of a "pair with exceptions" $\langle \Gamma \mid\mid \sigma \rangle - \Psi$ iff (Δ, λ) is an instance of (Γ, σ) but not an instance of Ψ.

We denote this fact by $\langle \Gamma \mid\mid \sigma \rangle - \Psi \leq (\Delta, \lambda)$.

Restricted Instantiation

Definition

We call a "pair with exceptions" $\langle \Gamma \mid\mid \sigma \rangle - \Psi$ consistent iff it has at least one instace.

Inconsistence Characterization

Lemma (Inconsistence Lemma)
A "pair with exceptions" $\langle \Gamma \mid\mid \sigma \rangle - \Psi$ *is inconsistent iff* (Γ, σ) *is an instance of* Ψ.

Corollary

Let $\langle \Gamma \mid\mid \sigma \rangle - \Psi$ *be a "pair with exceptions". If there is some* $(\nabla_I, \psi_I) \in \Psi$ *such that: there exists* δ *a substitution, for all* X*,*

$$\Gamma \vdash X\sigma \approx_\alpha X\psi_I\delta$$

Then $\langle \Gamma \mid\mid \sigma \rangle - \Psi$ *is inconsistent iff* $\Gamma \cap \nabla_I \neq$ *emptyset or* $\Gamma \vdash X\delta$*.*

Representation of Solutions

As with the case for unification problems we are interested in generate a *complete set* of solutions to a disunification problem P.

Definition

We call a set S of "pair with exceptions" a *complete representation* of solutions to the disunification problem P iff S satisfies the following conditions:

1. If $(\Delta, \lambda) \geq \langle \Gamma \mid\mid \sigma \rangle - \Psi$ for some $\langle \Gamma \mid\mid \sigma \rangle - \Psi$ in S then (Δ, λ) solves P.
2. If $(Delta, \lambda)$ solves P then it is an instance of some $\langle \Gamma \mid\mid \sigma \rangle - \Psi$ in S.
3. $\langle \Gamma \mid\mid \sigma \rangle - \Psi$ is consistent for all $\langle \Gamma \mid\mid \sigma \rangle - \Psi \in S$.

Theorem

Let

$$P = \left\langle \Delta, s_1 \overset{?}{\approx}_\alpha t_1, \cdots, s_n \overset{?}{\approx}_\alpha t_n \mid\mid \nabla, p_1 \overset{?}{\not\approx}_\alpha q_1, \cdots, p_m \overset{?}{\not\approx}_\alpha q_m \right\rangle$$

be a disunification problem. Define the family

$$\Psi := \bigcup_{p \overset{?}{\not\approx}_\alpha q \in D} c\mathcal{U}\left(p \overset{?}{\approx}_\alpha q\right)$$

Then the set

$$S = \{\langle \Gamma \mid\mid \sigma \rangle - \Psi \mid (\Gamma, \sigma) \in c\mathcal{U}(E) \ \textit{but not}\ \Psi(\Gamma, \sigma)\}$$

is a complete representation of solutions to the problem P.

Inconsistence Algorithm

input : $\langle \Gamma \mid\mid \sigma \rangle - \psi$ a finite pair with exceptions
output : **BOOL:** True if the input is consistent False otherwise

LET CON : BOOL ;
foreach $(\nabla_I, \psi_I) \in \Psi$ **do**
    if $\texttt{matching}((\Gamma, X\sigma \approx_\alpha X\psi_I) = (\Gamma', \delta))$ **then**
        if $\Gamma \cap \nabla_I \neq \emptyset$ *OR* $\Gamma \vdash X\delta$ **then**
            RETURN FALSE AND STOP
        else
            CON := TRUE
        end
    else
        RETURN FALSE AND STOP
    end
end

Algorithm 1: Inconsistence Algorithm

Disunification Algorithm

input : A disunification problem $P = \langle E \mid\mid D \rangle$.

output : a set S of pair with exceptions (can be emptyset)

LET $(\Gamma, \sigma) := \texttt{Unify}(E)$;

LET $\Psi := \bigcup_{p_i \overset{?}{\not\approx}_\alpha q_i \in D} \{(\nabla_i, \psi_i) = \texttt{Unify}(\nabla, p_i \overset{?}{\approx}_\alpha q_i)\}$;

if $\texttt{Consistent}(\langle \Gamma \mid\mid \sigma \rangle - \Psi)$ **then**

  RETURN $\langle \Gamma \mid\mid \sigma \rangle - \Psi$

else

  RETURN $\emptyset$ and STOP

end

Algorithm 2: Disunification Algorithm

Future Work

We plan to continue working on Nominal Disunification. That's our task list:

1. Extend Disunification to Nominal Equational Theories E (in the the context of Nominal Universal Algebras, [3]), like AC and C for example.
2. Study more general equational problems, like *Disunification with parameters*.

References I

W. L. Buntine and H.-J. Bürckert.
On solving equations and disequations.
J. ACM, 41(4):591–629, July 1994.

M. Fernández and M. J. Gabbay.
Nominal rewriting (journal version).
205(6):917–965, June 2007.

M. J. Gabbay and A. Mathijssen.
Nominal universal algebra: equational logic with names and binding.
19(6):1455–1508, December 2009.

Thank you.