A Formalisation of Nominal Equivalence with AC f-Symbols

Washington Carvalho II Daniele S. Nantes

Mauricio Ayala-Rincón Maribel Fernández

Grupo de Teoria da Computação - GTC

Departmento de Ciência da Computação - CIC

Universidade de Brasília - UnB

January, 28^{th} 2016

OUTLINE

Motivation

Problem Definition

Work Done so far

Conclusion and future work

Motivation

- $\approx_\alpha$-equivalence:

$$f(m,n) := \begin{cases} \texttt{if } m = 0 \texttt{ or } n = 0 \texttt{ then } m + n \\ \texttt{else} \begin{cases} \texttt{if } m \geq n \texttt{ then } f(m-n,n) \\ \texttt{else } f(n,m) \end{cases} \end{cases} .$$

$$g(i,j) := \begin{cases} \texttt{if } i = 0 \texttt{ or } j = 0 \texttt{ then } i + j \\ \texttt{else} \begin{cases} \texttt{if } i \geq j \texttt{ then } g(i-j,j) \\ \texttt{else } g(i,j) \end{cases} \end{cases}$$

- Associative (A) operators: $\oplus$, $(\ (\ ))$, x^y, $[\ ]_{n\times n} \cdot [\ ]_{n\times n}$, ...
- Associative-commutative (AC) operators: $\wedge$, $\vee$, $\cup$, $\cap$, ., $+$,...

Motivation

- $\approx_\alpha$ in the abstract data syntax:
  1. *Nominal Logic* [12]
  2. *Nominal Unification* [5, 11, 13, 17, 18]
  3. *Nominal Rewriting* [8, 9, 10]
  4. *Deduction Systems* [6]
  5. *Programing Languages* [4, 14, 15]
  6. *Proof Assistants* [1, 16]
- Other equational theories: $\approx_{\alpha E}$,
  $E \in \{A, C, U, I, G, AC, ACI, ACUI, ...\}$ [2, 3]
- We are proposing a study of $\approx_{\alpha,E}$
- A formalisation in Coq [7] of $\approx_{\alpha,A}$ and $\approx_{\alpha,AC}$

Nominal signature

- $\mathbf{\Sigma} := \left\{ \begin{array}{ll} \mathcal{A}: & a, b, c, ... \\ \mathcal{X}: & X, Y, Z, ... \\ \mathcal{S}: & f, g, h, ... \end{array} \right.$
- $\mathbf{\Pi} := \{$the set of *finite permutations* over $\mathcal{A}\}$
- $\pi \in \mathbf{\Pi}$ are expressed as lists of *swappings*: $[(a_1\ b_1), ..., (a_n\ b_n)]$
- Roughly speaking freshness constraints $(a \# X)$ express the idea that the atom a doesn't occur unbounded in X
- *Freshness contexts* are represented by $\nabla \subset \mathcal{A} \times \mathcal{X}$

Nominal signature

Basic definitions in Coq

```
Inductive Atom : Set := atom : nat → Atom.
Inductive Var : Set := var : nat → Var.
Definition Perm := list (Atom × Atom).
Definition Context := set (Atom × Var).
```

Nominal signature

$s, t ::= \langle\rangle \mid \mathsf{a} \mid [a]t \mid \langle s, t\rangle \mid f_k\, t \mid \pi.X$

```
Inductive term : Set :=
  | Ut : term
  | At : Atom → term
  | Ab : Atom → term → term
  | Pr : term → term → term
  | Fc : nat → term → term
  | Su : Perm → Var → term
.
```

Nominal signature

$$s, t ::= \langle\rangle \mid \mathtt{a} \mid [a]t \mid \langle s, t\rangle \mid f_k^E\ t \mid \pi.X$$

```
Inductive term : Set :=
  | Ut : term
  | At : Atom → term
  | Ab : Atom → term → term
  | Pr : term → term → term
  | Fc : nat → nat → term → term
  | Su : Perm → Var → term
.
```

Lambda calculus example: $x \mid \lambda x.t \mid (s\ t)$

$$\begin{cases} \mathtt{V}(a) & := & f_0(\mathtt{a}) & : & \mathtt{Atom} \to \mathtt{term} \\ \mathtt{L}(a,t) & := & f_1([a]t) & : & \mathtt{Atom} \times \mathtt{term} \to \mathtt{term} \\ \mathtt{A}(s,t) & := & f_2(\langle s,t\rangle) & : & \mathtt{term} \times \mathtt{term} \to \mathtt{term} \end{cases}$$

Nominal signature

$$s, t ::= \langle\rangle \mid \mathtt{a} \mid [a]t \mid \langle s, t\rangle \mid f_k^E\ t \mid \pi.X$$

```
Inductive term : Set :=
  | Ut : term
  | At : Atom → term
  | Ab : Atom → term → term
  | Pr : term → term → term
  | Fc : nat → nat → term → term
  | Su : Perm → Var → term
.
```

Lambda calculus example: $x \mid \lambda\, x.t \mid (s\ t)$

$$\begin{cases} \mathtt{V}(a) & := & f_0(\mathtt{a}) & : & \mathtt{Atom} \to \mathtt{term} \\ \mathtt{L}(a,t) & := & f_1([a]t) & : & \mathtt{Atom} \times \mathtt{term} \to \mathtt{term} \\ \mathtt{A}(s,t) & := & f_2(\langle s,t\rangle) & : & \mathtt{term} \times \mathtt{term} \to \mathtt{term} \end{cases}$$

π-action

$$((c\ d) :: \pi') \cdot a := \begin{cases} \texttt{if } c = a \texttt{ then } \pi' \cdot d \\ \texttt{else} \begin{cases} \texttt{if } d = a \texttt{ then } \pi' \cdot c \\ \texttt{else } \pi' \cdot a \end{cases} \end{cases}$$

$$\pi \cdot t := \begin{cases} \pi \cdot \langle\rangle & \to & \langle\rangle \\ \pi \cdot \mathtt{a} & \to & (\pi \cdot a) \\ \pi \cdot f_k^E\ t & \to & f_k^E\ (\pi \cdot t) \\ \pi \cdot \langle u, v \rangle & \to & \langle \pi \cdot u, \pi \cdot v \rangle \\ \pi \cdot ([a]t) & \to & [\pi \cdot a](\pi \cdot t) \\ \pi \cdot (\pi' \,.\, X) & \to & (\pi' \oplus \pi) \,.\, X \end{cases}$$

#-relation

$$\frac{}{\nabla \vdash a \# \langle\rangle}\,[\#\text{-ut}]$$

$$\frac{a \neq b}{\nabla \vdash \mathtt{a} \# \mathtt{b}}\,[\#\text{-at}]$$

$$\frac{\nabla \vdash a \# t}{\nabla \vdash a \# (f\ t)}\,[\#\text{-fc}]$$

$$\frac{}{\nabla \vdash a \# [a]t}\,[\#\text{-ab}_1]$$

$$\frac{\nabla \vdash a \# t_1 \quad \nabla \vdash a \# t_2}{\nabla \vdash a \# \langle t_1, t_2 \rangle}\,[\#\text{-pr}]$$

$$\frac{a \neq b \quad \nabla \vdash a \# t}{\nabla \vdash a \# [b]t}\,[\#\text{-ab}_2]$$

$$\frac{(\pi^{-1} \cdot a, X) \in \nabla}{\nabla \vdash a \# \pi.X}\,[\#\text{-su}]$$

$\approx_\alpha$-relation

$$\frac{}{\nabla \vdash \langle\rangle \approx_\alpha \langle\rangle}\,[\approx_\alpha\text{-ut}]$$

$$\frac{}{\nabla \vdash \mathsf{a} \approx_\alpha \mathsf{a}}\,[\approx_\alpha\text{-at}]$$

$$\frac{\nabla \vdash t \approx_\alpha t'}{\nabla \vdash f\ t \approx_\alpha f\ t'}\,[\approx_\alpha\text{-fc}]$$

$$\frac{\nabla \vdash t \approx_\alpha t'}{\nabla \vdash [a]t \approx_\alpha [a]t'}\,[\approx_\alpha\text{-ab}_1]$$

$$\frac{\nabla \vdash t_1 \approx_\alpha t_1' \quad \nabla \vdash t_2 \approx_\alpha t_2'}{\nabla \vdash \langle t_1, t_2\rangle \approx_\alpha \langle t_1', t_2'\rangle}\,[\approx_\alpha\text{-pr}]$$

$$\frac{a \neq b \quad \nabla \vdash t \approx_\alpha (a\,b)\,t' \quad \nabla \vdash a \# t'}{\nabla \vdash [a]t \approx_\alpha [b]t'}\,[\approx_\alpha\text{-ab}_2]$$

$$\frac{\forall a \in ds(\pi, \pi'),\ (a, X) \in \nabla}{\nabla \vdash \pi.X \approx_\alpha \pi'.X}\,[\approx_\alpha\text{-su}]$$

$$ds(\pi, \pi') := \{a \mid \pi \cdot a \neq \pi' \cdot a\}$$

$\approx_\alpha$-relation

$$\frac{}{\nabla \vdash \langle\rangle \approx_\alpha \langle\rangle}\,[\approx_\alpha\text{-ut}]$$

$$\frac{}{\nabla \vdash \mathsf{a} \approx_\alpha \mathsf{a}}\,[\approx_\alpha\text{-at}]$$

$$\frac{\nabla \vdash t \approx_\alpha t'}{\nabla \vdash f\ t \approx_\alpha f\ t'}\,[\approx_\alpha\text{-fc}]$$

$$\frac{\nabla \vdash t \approx_\alpha t'}{\nabla \vdash [a]t \approx_\alpha [a]t'}\,[\approx_\alpha\text{-ab}_1]$$

$$\frac{\nabla \vdash t_1 \approx_\alpha t_1' \quad \nabla \vdash t_2 \approx_\alpha t_2'}{\nabla \vdash \langle t_1, t_2\rangle \approx_\alpha \langle t_1', t_2'\rangle}\,[\approx_\alpha\text{-pr}]$$

$$\frac{a \neq b \quad \nabla \vdash t \approx_\alpha (a\,b)\,t' \quad \nabla \vdash a \# t'}{\nabla \vdash [a]t \approx_\alpha [b]t'}\,[\approx_\alpha\text{-ab}_2]$$

$$\frac{\forall a \in ds(\pi, \pi'),\ (a, X) \in \nabla}{\nabla \vdash \pi.X \approx_\alpha \pi'.X}\,[\approx_\alpha\text{-su}]$$

$$ds(\pi, \pi') := \{a \mid \pi \cdot a \neq \pi' \cdot a\}$$

$\approx_\alpha$ inductive definition

```
Inductive alpha_equiv : Context → term → term → Prop :=
 | ≈α_Ut : ∀ C,    alpha_equiv C (<<>>) (<<>>)
 | ≈α_At : ∀ C a, alpha_equiv C (%a) (%a)
 | ≈α_Pr : ∀ C t1 t2 t1' t2', (alpha_equiv C t1 t1') →
                              (alpha_equiv C t2 t2') →
                    alpha_equiv C (<|t1,t2|>) (<|t1',t2'|>)
 | ≈α_Fc : ∀ m n t t' C,       (alpha_equiv C t t') →
                     alpha_equiv C (Fc m n t) (Fc m n t')
 | ≈α_Ab1 : ∀ C a t t',        (alpha_equiv C t t') →
                     alpha_equiv C ([a]^t) ([a]^t')
 | ≈α_Ab2 : ∀ C a a' t t',     a ≠ a' → C ⊢ a # t' →
                              (alpha_equiv C t (|[(a,a')] @ t')) →
                     alpha_equiv C ([a]^t) ([a']^t')
 | ≈α_Su : ∀ (C: Context) p p' (X: Var),
                 (∀ a, (ln_ds p p' a) → set_ln ((a,X)) C) →
                    alpha_equiv C (p\X) (p'\X) .
```

$\approx_\alpha$ inductive definition

```
Inductive alpha_equiv : Context → term → term → Prop :=
 | ≈α_Ut : ∀ C,    alpha_equiv C (<<>>) (<<>>)
 | ≈α_At : ∀ C a, alpha_equiv C (%a) (%a)
 | ≈α_Pr : ∀ C t1 t2 t1' t2', (alpha_equiv C t1 t1') →
                              (alpha_equiv C t2 t2') →
                    alpha_equiv C (<|t1,t2|>) (<|t1',t2'|>)
 | ≈α_Fc : ∀ m n t t' C,       (alpha_equiv C t t') →
                     alpha_equiv C (Fc m n t) (Fc m n t')
 | ≈α_Ab1 : ∀ C a t t',        (alpha_equiv C t t') →
                     alpha_equiv C ([a]^t) ([a]^t')
 | ≈α_Ab2 : ∀ C a a' t t',     a ≠ a' → C ⊢ a # t' →
                              (alpha_equiv C t (|[(a,a')] @ t')) →
                     alpha_equiv C ([a]^t) ([a']^t')
 | ≈α_Su : ∀ (C: Context) p p' (X: Var),
                 (∀ a, (ln_ds p p' a) → set_ln ((a,X)) C) →
                    alpha_equiv C (p\X) (p'\X) .
```

$\approx_\alpha$ inductive definition

```
Inductive alpha_equiv : Context → term → term → Prop :=
 | ≈α_Ut : ∀ C,    alpha_equiv C (<<>>) (<<>>)
 | ≈α_At : ∀ C a, alpha_equiv C (%a) (%a)
 | ≈α_Pr : ∀ C t1 t2 t1' t2', (alpha_equiv C t1 t1') →
                              (alpha_equiv C t2 t2') →
                    alpha_equiv C (<|t1,t2|>) (<|t1',t2'|>)
 | ≈α_Fc : ∀ m n t t' C,      (alpha_equiv C t t') →
                              alpha_equiv C (Fc m n t) (Fc m n t')
 | ≈α_Ab1 : ∀ C a t t',       (alpha_equiv C t t') →
                              alpha_equiv C ([a]^t) ([a]^t')
 | ≈α_Ab2 : ∀ C a a' t t',    a ≠ a' → C ⊢ a # t' →
                              (alpha_equiv C t (|[(a,a')] @ t')) →
                              alpha_equiv C ([a]^t) ([a']^t')
 | ≈α_Su : ∀ (C: Context) p p' (X: Var),
                    (∀ a, (ln_ds p p' a) → set_ln ((a,X)) C) →
                              alpha_equiv C (p\X) (p'\X) .
```

$\approx_\alpha$ inductive definition

```
Inductive alpha_equiv : Context → term → term → Prop :=
 | ≈α_Ut : ∀ C,    alpha_equiv C (<<>>) (<<>>)
 | ≈α_At : ∀ C a, alpha_equiv C (%a) (%a)
 | ≈α_Pr : ∀ C t1 t2 t1' t2', (alpha_equiv C t1 t1') →
                              (alpha_equiv C t2 t2') →
                    alpha_equiv C (<|t1,t2|>) (<|t1',t2'|>)
 | ≈α_Fc : ∀ m n t t' C,      (alpha_equiv C t t') →
                              alpha_equiv C (Fc m n t) (Fc m n t')
 | ≈α_Ab1 : ∀ C a t t',       (alpha_equiv C t t') →
                              alpha_equiv C ([a]^t) ([a]^t')
 | ≈α_Ab2 : ∀ C a a' t t',    a ≠ a' → C ⊢ a # t' →
                              (alpha_equiv C t (|[(a,a')] @ t')) →
                              alpha_equiv C ([a]^t) ([a']^t')
 | ≈α_Su : ∀ (C: Context) p p' (X: Var),
                    (∀ a, (ln_ds p p' a) → set_ln ((a,X)) C) →
                              alpha_equiv C (p\X) (p'\X) .
```

$\approx_\alpha$ inductive definition

```
Inductive alpha_equiv : Context → term → term → Prop :=
 | ≈α_Ut : ∀ C,    alpha_equiv C (<<>>) (<<>>)
 | ≈α_At : ∀ C a, alpha_equiv C (%a) (%a)
 | ≈α_Pr : ∀ C t1 t2 t1' t2', (alpha_equiv C t1 t1') →
                              (alpha_equiv C t2 t2') →
                    alpha_equiv C (<|t1,t2|>) (<|t1',t2'|>)
 | ≈α_Fc : ∀ m n t t' C,      (alpha_equiv C t t') →
                              alpha_equiv C (Fc m n t) (Fc m n t')
 | ≈α_Ab1 : ∀ C a t t',       (alpha_equiv C t t') →
                              alpha_equiv C ([a]^t) ([a]^t')
 | ≈α_Ab2 : ∀ C a a' t t',    a ≠ a' → C ⊢ a # t' →
                              (alpha_equiv C t (|[(a,a')] @ t')) →
                              alpha_equiv C ([a]^t) ([a']^t')
 | ≈α_Su : ∀ (C: Context) p p' (X: Var),
                    (∀ a, (ln_ds p p' a) → set_ln ((a,X)) C) →
                              alpha_equiv C (p\X) (p'\X) .
```

$\approx_\alpha$ inductive definition

```
Inductive alpha_equiv : Context → term → term → Prop :=
 | ≈α_Ut : ∀ C,    alpha_equiv C (<<>>) (<<>>)
 | ≈α_At : ∀ C a, alpha_equiv C (%a) (%a)
 | ≈α_Pr : ∀ C t1 t2 t1' t2', (alpha_equiv C t1 t1') →
                              (alpha_equiv C t2 t2') →
                    alpha_equiv C (<|t1,t2|>) (<|t1',t2'|>)
 | ≈α_Fc : ∀ m n t t' C,      (alpha_equiv C t t') →
                              alpha_equiv C (Fc m n t) (Fc m n t')
 | ≈α_Ab1 : ∀ C a t t',       (alpha_equiv C t t') →
                              alpha_equiv C ([a]^t) ([a]^t')
 | ≈α_Ab2 : ∀ C a a' t t',    a ≠ a' → C ⊢ a # t' →
                              (alpha_equiv C t ([(a,a')] @ t')) →
                              alpha_equiv C ([a]^t) ([a']^t')
 | ≈α_Su : ∀ (C: Context) p p' (X: Var),
                    (∀ a, (ln_ds p p' a) → set_ln ((a,X)) C) →
                              alpha_equiv C (p\X) (p'\X) .
```

$\approx_\alpha$ inductive definition

```
Inductive alpha_equiv : Context → term → term → Prop :=
 | ≈α_Ut : ∀ C,    alpha_equiv C (<<>>) (<<>>)
 | ≈α_At : ∀ C a, alpha_equiv C (%a) (%a)
 | ≈α_Pr : ∀ C t1 t2 t1' t2', (alpha_equiv C t1 t1') →
                              (alpha_equiv C t2 t2') →
                    alpha_equiv C (<|t1,t2|>) (<|t1',t2'|>)
 | ≈α_Fc : ∀ m n t t' C,      (alpha_equiv C t t') →
                              alpha_equiv C (Fc m n t) (Fc m n t')
 | ≈α_Ab1 : ∀ C a t t',       (alpha_equiv C t t') →
                              alpha_equiv C ([a]^t) ([a]^t')
 | ≈α_Ab2 : ∀ C a a' t t',    a ≠ a' → C ⊢ a # t' →
                             (alpha_equiv C t ([(a,a')] @ t')) →
                              alpha_equiv C ([a]^t) ([a']^t')
 | ≈α_Su : ∀ (C: Context) p p' (X: Var),
                    (∀ a, (In_ds p p' a) → set_In ((a,X)) C) →
                          alpha_equiv C (p\X) (p'\X) .
```

Weak equivalence ($\sim_\omega$)

$$\frac{}{\langle\rangle \sim_\omega \langle\rangle}\ [\sim_\omega\text{-ut}] \qquad \frac{}{a \sim_\omega a}\ [\sim_\omega\text{-at}] \qquad \frac{t \sim_\omega t'}{f_k^E\, t \sim_\omega f_k^E\, t'}\ [\sim_\omega\text{-fc}]$$

$$\frac{t_1 \sim_\omega t_1' \qquad t_2 \sim_\omega t_2'}{\langle t_1, t_2\rangle \sim_\omega \langle t_1', t_2'\rangle}\ [\sim_\omega\text{-pr}] \qquad \frac{t \sim_\omega t'}{[a]t \sim_\omega [a]t'}\ [\sim_\omega\text{-ab}]$$

$$\frac{ds(\pi, \pi') = \emptyset}{\pi \cdot X \sim_\omega \pi' \cdot X'}\ [\sim_\omega\text{-su}]$$

Lemma 1 (Restrict transitivity)

If $\nabla \vdash t_1 \approx_\alpha t_2$ and $t_2 \sim_\omega t_3$ than $\nabla \vdash t_1 \approx_\alpha t_3$.

Proof: *induction on $\approx_\alpha$ applying freshness preservation of $\sim_\omega$.*

Weak equivalence ($\sim_\omega$)

$$\frac{}{\langle\rangle \sim_\omega \langle\rangle}\ [\sim_\omega\text{-ut}] \qquad \frac{}{a \sim_\omega a}\ [\sim_\omega\text{-at}] \qquad \frac{t \sim_\omega t'}{f_k^E\, t \sim_\omega f_k^E\, t'}\ [\sim_\omega\text{-fc}]$$

$$\frac{t_1 \sim_\omega t_1' \qquad t_2 \sim_\omega t_2'}{\langle t_1, t_2\rangle \sim_\omega \langle t_1', t_2'\rangle}\ [\sim_\omega\text{-pr}] \qquad \frac{t \sim_\omega t'}{[a]t \sim_\omega [a]t'}\ [\sim_\omega\text{-ab}]$$

$$\frac{ds(\pi, \pi') = \emptyset}{\pi \cdot X \sim_\omega \pi' \cdot X'}\ [\sim_\omega\text{-su}]$$

Lemma 1 (Restrict transitivity)

If $\nabla \vdash t_1 \approx_\alpha t_2$ *and* $t_2 \sim_\omega t_3$ *than* $\nabla \vdash t_1 \approx_\alpha t_3$.

Proof: *induction on* $\approx_\alpha$ *applying freshness preservation of* $\sim_\omega$.

Preliminary lemmas towards $\approx_\alpha$-transitivity

Lemma 2 (Freshness preservation of $\approx_\alpha$)

If $\nabla \vdash a \# t$ *and* $\nabla \vdash t \approx_\alpha t'$ *then* $\nabla \vdash a \# t'$.

Proof: *induction on* $\approx_\alpha$ *applying* $\#$ *properties.*

Lemma 3 (Equivariance of $\approx_\alpha$)

If $\nabla \vdash t \approx_\alpha t'$ *then* $\nabla \vdash \pi \cdot t \approx_\alpha \pi \cdot t'$.

Proof: *induction on* $\approx_\alpha$ *applying the Lemma 1 .*

Lemma 4

$(\forall a \in ds(\pi, \pi'), \nabla \vdash a \# t)$ *iff* $\nabla \vdash \pi \cdot t \approx_\alpha \pi' \cdot t$.

Proof: *induction on* t *applying the definition of the permutation action.*

Lemma 5

If $\nabla \vdash t_1 \approx_\alpha t_2$ *and* $\nabla \vdash t_2 \approx_\alpha \pi \cdot t_2$ *then* $\nabla \vdash t_1 \approx_\alpha \pi \cdot t_2$.

Proof: *induction on* $\approx_\alpha$ *with applications of the lemmas 1, 2, 3 and 4.*

Preliminary lemmas towards $\approx_\alpha$-transitivity

Lemma 2 (Freshness preservation of $\approx_\alpha$)

If $\nabla \vdash a \# t$ *and* $\nabla \vdash t \approx_\alpha t'$ *then* $\nabla \vdash a \# t'$.

Proof: *induction on* $\approx_\alpha$ *applying* $\#$ *properties.*

Lemma 3 (Equivariance of $\approx_\alpha$)

If $\nabla \vdash t \approx_\alpha t'$ *then* $\nabla \vdash \pi \cdot t \approx_\alpha \pi \cdot t'$.

Proof: *induction on* $\approx_\alpha$ *applying the Lemma 1 .*

Lemma 4

$(\forall a \in ds(\pi, \pi'), \nabla \vdash a \# t)$ *iff* $\nabla \vdash \pi \cdot t \approx_\alpha \pi' \cdot t$.

Proof: *induction on* t *applying the definition of the permutation action.*

Lemma 5

If $\nabla \vdash t_1 \approx_\alpha t_2$ *and* $\nabla \vdash t_2 \approx_\alpha \pi \cdot t_2$ *then* $\nabla \vdash t_1 \approx_\alpha \pi \cdot t_2$.

Proof: *induction on* $\approx_\alpha$ *with applications of the lemmas 1, 2, 3 and 4.*

Preliminary lemmas towards $\approx_\alpha$-transitivity

Lemma 2 (Freshness preservation of $\approx_\alpha$)

If $\nabla \vdash a \# t$ *and* $\nabla \vdash t \approx_\alpha t'$ *then* $\nabla \vdash a \# t'$.

Proof: *induction on* $\approx_\alpha$ *applying* $\#$ *properties.*

Lemma 3 (Equivariance of $\approx_\alpha$)

If $\nabla \vdash t \approx_\alpha t'$ *then* $\nabla \vdash \pi \cdot t \approx_\alpha \pi \cdot t'$.

Proof: *induction on* $\approx_\alpha$ *applying the Lemma 1 .*

Lemma 4

$(\forall a \in ds(\pi, \pi'), \nabla \vdash a \# t)$ *iff* $\nabla \vdash \pi \cdot t \approx_\alpha \pi' \cdot t$.

Proof: *induction on* t *applying the definition of the permutation action.*

Lemma 5

If $\nabla \vdash t_1 \approx_\alpha t_2$ *and* $\nabla \vdash t_2 \approx_\alpha \pi \cdot t_2$ *then* $\nabla \vdash t_1 \approx_\alpha \pi \cdot t_2$.

Proof: *induction on* $\approx_\alpha$ *with applications of the lemmas 1, 2, 3 and 4.*

Preliminary lemmas towards $\approx_\alpha$-transitivity

Lemma 2 (Freshness preservation of $\approx_\alpha$)

If $\nabla \vdash a \# t$ *and* $\nabla \vdash t \approx_\alpha t'$ *then* $\nabla \vdash a \# t'$.

Proof: *induction on* $\approx_\alpha$ *applying* $\#$ *properties.*

Lemma 3 (Equivariance of $\approx_\alpha$)

If $\nabla \vdash t \approx_\alpha t'$ *then* $\nabla \vdash \pi \cdot t \approx_\alpha \pi \cdot t'$.

Proof: *induction on* $\approx_\alpha$ *applying the Lemma 1 .*

Lemma 4

$(\forall a \in ds(\pi, \pi'), \nabla \vdash a \# t)$ *iff* $\nabla \vdash \pi \cdot t \approx_\alpha \pi' \cdot t$.

Proof: *induction on* t *applying the definition of the permutation action.*

Lemma 5

If $\nabla \vdash t_1 \approx_\alpha t_2$ *and* $\nabla \vdash t_2 \approx_\alpha \pi \cdot t_2$ *then* $\nabla \vdash t_1 \approx_\alpha \pi \cdot t_2$.

Proof: *induction on* $\approx_\alpha$ *with applications of the lemmas 1, 2, 3 and 4.*

$\approx_\alpha$ is in fact an equivalence relation

Lemma 6 (Reflexivity of $\approx_\alpha$)

$\nabla \vdash t \approx_\alpha t$.

Proof: *induction on* t .

Lemma 7 (Transitivity of $\approx_\alpha$)

If $\nabla \vdash t_1 \approx_\alpha t_2$ *and* $\nabla \vdash t_2 \approx_\alpha t_3$ *then* $\nabla \vdash t_1 \approx_\alpha t_3$.

Proof: *induction on* $\nabla \vdash t_1 \approx_\alpha t_2$, *with applications of the lemmas 1, 2 and 5.*

Lemma 8 (Symmetry of $\approx_\alpha$)

If $\nabla \vdash t \approx_\alpha t'$ *then* $\nabla \vdash t' \approx_\alpha t$.

Proof: *induction on* $\nabla \vdash t \approx_\alpha t'$ *with applications of the lemmas 3 and 7.*

Extension of $\approx_\alpha$-rules

$$\frac{\|s\|, \|t\| \neq 1,\ \nabla \vdash s_{(1)} \approx_{\alpha,A} t_{(1)} \qquad \nabla \vdash f\ s_{[\star 1]} \approx_{\alpha,A} f\ t_{[\star 1]}}{\nabla \vdash f\ s \approx_{\alpha,A} f\ t}\ [\approx_A]$$

$$\frac{\|s\|, \|t\| \neq 1 \quad \exists_{(0 < i \leq \|t\|)},\ \nabla \vdash s_{(1)} \approx_{\alpha,AC} t_{(i)} \quad \nabla \vdash f\ s_{[\star 1]} \approx_{\alpha,AC} f\ t_{[\star i]}}{\nabla \vdash f\ s \approx_{\alpha,AC} f\ t}\ [\approx_{AC}]$$

$$\frac{\|s\|, \|t\| = 1 \quad \nabla \vdash s_{(1)} \approx_\alpha t_{(1)}}{\nabla \vdash f\ s \approx_\alpha f\ t}\ [\approx_{\|\,.\,\|=1}]$$

Extension of $\approx_\alpha$-rules

$$\frac{\|s\|, \|t\| \neq 1,\ \nabla \vdash s_{(1)} \approx_{\alpha,A} t_{(1)} \qquad \nabla \vdash f\ s_{[\star 1]} \approx_{\alpha,A} f\ t_{[\star 1]}}{\nabla \vdash f\ s \approx_{\alpha,A} f\ t}\ [\approx_A]$$

$$\frac{\|s\|, \|t\| \neq 1 \quad \exists_{(0 < i \leq \|t\|)},\ \nabla \vdash s_{(1)} \approx_{\alpha,AC} t_{(i)} \quad \nabla \vdash f\ s_{[\star 1]} \approx_{\alpha,AC} f\ t_{[\star i]}}{\nabla \vdash f\ s \approx_{\alpha,AC} f\ t}\ [\approx_{AC}]$$

$$\frac{\|s\|, \|t\| = 1 \quad \nabla \vdash s_{(1)} \approx_\alpha t_{(1)}}{\nabla \vdash f\ s \approx_\alpha f\ t}\ [\approx_{\|\,.\,\|=1}]$$

Extension of $\approx_\alpha$-rules

$$\frac{\|s\|, \|t\| \neq 1,\ \nabla \vdash s_{(1)} \approx_{\alpha,A} t_{(1)} \qquad \nabla \vdash f\ s_{[\star 1]} \approx_{\alpha,A} f\ t_{[\star 1]}}{\nabla \vdash f\ s \approx_{\alpha,A} f\ t}\ [\approx_A]$$

$$\frac{\|s\|, \|t\| \neq 1 \quad \exists_{(0 < i \leq \|t\|)},\ \nabla \vdash s_{(1)} \approx_{\alpha,AC} t_{(i)} \quad \nabla \vdash f\ s_{[\star 1]} \approx_{\alpha,AC} f\ t_{[\star i]}}{\nabla \vdash f\ s \approx_{\alpha,AC} f\ t}\ [\approx_{AC}]$$

$$\frac{\|s\|, \|t\| = 1 \quad \nabla \vdash s_{(1)} \approx_\alpha t_{(1)}}{\nabla \vdash f\ s \approx_\alpha f\ t}\ [\approx_{\|\,.\,\|=1}]$$

Tuple length

$$\|t\|_{f_k^E} := \begin{cases} \langle s, u \rangle & \to \|s\|_{f_k^E} + \|u\|_{f_k^E} \\ f_{k_0}^{E_0}\, s & \to \begin{cases} \texttt{if } f_{k_0}^{E_0} = f_k^E \texttt{ then } \|s\|_{f_k^E} \\ \texttt{else } 1 \end{cases} \\ \_ & \to 1 \end{cases}$$

Example

$\|f\,\langle [a](\pi \cdot X), f\,\langle b, \langle a, (g\,\langle \pi'.Y, \mathtt{b}\rangle)\rangle\rangle\rangle\|_f = 4$

i^{th} of a tuple

$$t_{(i)_{f_k^E}} := \begin{cases} \langle s, u \rangle \to \begin{cases} \texttt{if } i \leq \|s\|_{f_k^E} \texttt{ then } s_{(i)_{f_k^E}} \\ \texttt{else } u_{(i - \|s\|_{f_k^E})_{f_k^E}} \end{cases} \\ f_{k_0}^{E_0}\, s \to \begin{cases} \texttt{if } i = 0 \texttt{ then } \langle\rangle \\ \texttt{else} \begin{cases} \texttt{if } i \leq \|s\|_{f_k^E} \texttt{ then } \begin{cases} \texttt{if } f_{k_0}^{E_0} = f_k^E \texttt{ then } s_{(i)_{f_k^E}} \\ \texttt{else } f_{k_0}^{E_0}\, s \end{cases} \\ \texttt{else } \langle\rangle \end{cases} \end{cases} \\ \_ \to \begin{cases} \texttt{if } i = 0 \texttt{ then } \langle\rangle \\ \texttt{else } t \end{cases} \end{cases}$$

Example

$$(f\ \langle [a](\pi \cdot X), f\ \langle b, \langle \mathtt{a}, (g\ \langle \pi'.Y, \mathtt{b} \rangle) \rangle \rangle \rangle)_{(3)_f} = \mathtt{a}$$

The deletion of an i^{th} element of a tuple

$$
t_{[\star i]_{f_k^E}} := \begin{cases}
\langle s, u\rangle \to \begin{cases}
\texttt{if } i = 0 \texttt{ then } \langle s, u\rangle \\
\texttt{else } \begin{cases}
\texttt{if } i \leq \|s\|_{f_k^E} \texttt{ then } \begin{cases} \texttt{if } \|s\|_{f_k^E} = 1 \texttt{ then } u \\ \texttt{else } \langle s_{[\star i]_{f_k^E}}, u\rangle \end{cases} \\
\texttt{else } \begin{cases} \texttt{if } i \leq \|s\|_{f_k^E} + \|t\|_{f_k^E} \texttt{ then } \begin{cases} \texttt{if } \|t\|_{f_k^E} = 1 \texttt{ then } s \\ \texttt{else } \langle s, u_{[\star(i - \|s\|_{f_k^E})]_{f_k^E}}\rangle \end{cases} \\ \texttt{else } \langle s, u\rangle \end{cases}
\end{cases}
\end{cases} \\
f_{k_0}^{E_0}\, s \to \begin{cases}
\texttt{if } i = 0 \texttt{ then } f_{k_0}^{E_0}\, s \\
\texttt{else } \begin{cases}
\texttt{if } i \leq \|s\|_{f_k^E} \texttt{ then } \begin{cases} \texttt{if } f_{k_0}^{E_0} = f_k^E \texttt{ then } \begin{cases} \texttt{if } \|s\|_{f_k^E} = 1 \texttt{ then } \langle\rangle \\ \texttt{else } s_{[\star i]_{f_k^E}} \end{cases} \\ \texttt{else } \langle\rangle \end{cases} \\
\texttt{else } f_{k_0}^{E_0}\, s
\end{cases}
\end{cases} \\
\_ \to \begin{cases} \texttt{if } i = 1 \texttt{ then } \langle\rangle \\ \texttt{else } t \end{cases}
\end{cases}
$$

Example

$$(f\,\langle [a](\pi \cdot X), f\,\langle b, \langle \mathtt{a}, (g\,\langle \pi'.Y, \mathtt{b}\rangle)\rangle\rangle\rangle)_{[\star 4]_f} = f\,\langle [a](\pi \cdot X), f\,\langle b, \mathtt{a}\rangle\rangle$$

`size_term` measure

$$
\begin{aligned}
|[a]t| &\rightarrow |t| + 1 \\
|<s, t>| &\rightarrow |s| + |t| + 1 \\
|f\ t| &\rightarrow |t| + 1 \\
|\_| &\rightarrow 1
\end{aligned}
$$

Preliminary lemmas for $\approx_{\alpha,E}$, $E \in \{A, AC\}$

Lemma 9

If $\nabla \vdash t_1 \approx_{\alpha,E} t_2$ *and* $\nabla \vdash t_2 \approx_{\alpha} t_3$ *then* $\nabla \vdash t_1 \approx_{\alpha,E} t_3$.

Proof: *induction on* $\approx_{\alpha,E}$.

Lemma 10 (Freshness preservation of $\approx_{\alpha,E}$)

If $\nabla \vdash a \# t$ *and* $\nabla \vdash t \approx_{\alpha,E} t'$ *than* $\nabla \vdash a \# t'$.

Proof: *induction on* $\approx_{\alpha,E}$.

Lemma 11 (Equivariance of $\approx_{\alpha,E}$)

If $\nabla \vdash t \approx_{\alpha,E} t'$ *then* $\nabla \vdash \pi \cdot t \approx_{\alpha,E} \pi \cdot t'$.

Proof: *induction on* $\approx_{\alpha,E}$ *and application of the Lemma 9.*

Preliminary lemmas for $\approx_{\alpha,E}$, $E \in \{A, AC\}$

Lemma 9

If $\nabla \vdash t_1 \approx_{\alpha,E} t_2$ *and* $\nabla \vdash t_2 \approx_{\alpha} t_3$ *then* $\nabla \vdash t_1 \approx_{\alpha,E} t_3$.

Proof: *induction on* $\approx_{\alpha,E}$.

Lemma 10 (Freshness preservation of $\approx_{\alpha,E}$)

If $\nabla \vdash a \# t$ *and* $\nabla \vdash t \approx_{\alpha,E} t'$ *than* $\nabla \vdash a \# t'$.

Proof: *induction on* $\approx_{\alpha,E}$.

Lemma 11 (Equivariance of $\approx_{\alpha,E}$)

If $\nabla \vdash t \approx_{\alpha,E} t'$ *then* $\nabla \vdash \pi \cdot t \approx_{\alpha,E} \pi \cdot t'$.

Proof: *induction on* $\approx_{\alpha,E}$ *and application of the Lemma 9.*

Preliminary lemmas for $\approx_{\alpha,E}$, $E \in \{A, AC\}$

Lemma 9

If $\nabla \vdash t_1 \approx_{\alpha,E} t_2$ *and* $\nabla \vdash t_2 \approx_{\alpha} t_3$ *then* $\nabla \vdash t_1 \approx_{\alpha,E} t_3$.

Proof: *induction on* $\approx_{\alpha,E}$.

Lemma 10 (Freshness preservation of $\approx_{\alpha,E}$)

If $\nabla \vdash a \# t$ *and* $\nabla \vdash t \approx_{\alpha,E} t'$ *than* $\nabla \vdash a \# t'$.

Proof: *induction on* $\approx_{\alpha,E}$.

Lemma 11 (Equivariance of $\approx_{\alpha,E}$)

If $\nabla \vdash t \approx_{\alpha,E} t'$ *then* $\nabla \vdash \pi \cdot t \approx_{\alpha,E} \pi \cdot t'$.

Proof: *induction on* $\approx_{\alpha,E}$ *and application of the Lemma 9.*

$\approx_{\alpha,A}$ and $\approx_{\alpha,AC}$ are indeed equivalence relations

Lemma 12

If $\nabla \vdash t \approx_{\alpha,AC} t'$ *then* $\forall_{(0<i\leq \|t\|_{f_k^E})} \exists_{(0<j\leq \|t\|_{f_k^E})}$,
$\nabla \vdash t_{(i)_{f_k^E}} \approx_{\alpha,AC} t'_{(j)_{f_k^E}}$ *and* $\nabla \vdash t_{[\star i]_{f_k^E}} \approx_{\alpha,AC} t_{[\star j]_{f_k^E}}$.

Proof: *induction on* $\|t\|_{f_k^E}$.

Lemma 13 (Reflexivity of $\approx_{\alpha,E}$)

$\nabla \vdash t \approx_{\alpha,E} t$. **Proof**: *induction on* t .

Lemma 14 (Transitivity of $\approx_{\alpha,E}$)

If $\nabla \vdash t_1 \approx_{\alpha,E} t_2$ *and* $\nabla \vdash t_2 \approx_{\alpha,E} t_3$ *then* $\nabla \vdash t_1 \approx_{\alpha,E} t_3$.
Proof: *induction on* $|t_1|$ *using the lemmas 9, 10, 11 and 12,*

Lemma 15 (Symmetry of $\approx_{\alpha,E}$)

If $\nabla \vdash t \approx_{\alpha,E} t'$ *then* $\nabla \vdash t' \approx_{\alpha,E} t$. **Proof**: $\approx_{\alpha,E}$ *(L. 13 and 14).*

$\approx_{\alpha,A}$ and $\approx_{\alpha,AC}$ are indeed equivalence relations

Lemma 12

If $\nabla \vdash t \approx_{\alpha,AC} t'$ *then* $\forall_{(0<i\leq \|t\|_{f_k^E})} \exists_{(0<j\leq \|t\|_{f_k^E})}$,
$\nabla \vdash t_{(i)_{f_k^E}} \approx_{\alpha,AC} t'_{(j)_{f_k^E}}$ *and* $\nabla \vdash t_{[\star i]_{f_k^E}} \approx_{\alpha,AC} t_{[\star j]_{f_k^E}}$.

Proof: *induction on* $\|t\|_{f_k^E}$.

Lemma 13 (Reflexivity of $\approx_{\alpha,E}$)

$\nabla \vdash t \approx_{\alpha,E} t$. **Proof**: *induction on* t .

Lemma 14 (Transitivity of $\approx_{\alpha,E}$)

If $\nabla \vdash t_1 \approx_{\alpha,E} t_2$ *and* $\nabla \vdash t_2 \approx_{\alpha,E} t_3$ *then* $\nabla \vdash t_1 \approx_{\alpha,E} t_3$.
Proof: *induction on* $|t_1|$ *using the lemmas 9, 10, 11 and 12,*

Lemma 15 (Symmetry of $\approx_{\alpha,E}$)

If $\nabla \vdash t \approx_{\alpha,E} t'$ *then* $\nabla \vdash t' \approx_{\alpha,E} t$. **Proof**: $\approx_{\alpha,E}$ *(L. 13 and 14).*

Conclusion

- We've formalised the soundness of the theories $\approx_{\alpha,A}$ and $\approx_{\alpha,AC}$
- The proof begins with $\approx_{\alpha}$'s soundness and then it is extented, in a modular way, to other equational theories
- We should rebuild the proof of $\approx_{\alpha}$ without using $\sim_{\omega}$
- $E \in \{A, AC, AI, ACI, ..\}$
- Unification modulo $\approx_{\alpha,E}$

References I

[1] Brian Aydemir, Aaron Bohannon, and Stephanie Weirich.
Nominal Reasoning Techniques in Coq.
Electronic Notes in Theoretical Computer Science, 174(5):69–77, 2007.

[2] Franz Baader, Wayne Snyder, Paliath Narendran, M Schmidt-Schauß, and Klaus U Schulz.
Unification Theory.
Handbook of logic in artificial intelligence and logic programming, 2001.

[3] Alexandre Boudet, Evelyne Contejean, and Hervé Devie.
A New AC Unification Algorithm with an Algorithm for Solving Systems of Diophantine Equations.
Proceedings, Fifth Annual IEEE Symposium on Logic in Computer Science, pages 289–299, 1990.

References II

[4] William E. Byrd and Daniel P. Friedman.
αKanren: A Fresh Name in Nominal Logic Programming.
In Prooceedings of the 2007 Workshop on Scheme and Functional Programming, pages 79–90, 2007.

[5] Christophe François Olivier Calvès and Maribel Fernández.
Implementing Nominal Unification.
Electronic Notes in Theoretical Computer Science, 176(1):25–37, 2007.

[6] James Cheney.
α Prolog Users Guide & Language Reference Version 0.3 DRAFT.
pages 0–28, 2003.

[7] CoqTeam.
The Coq Proof Assistant Reference Manual.
2009.

References III

[8] Maribel Fernández and Murdoch James Gabbay.
Nominal rewriting.
Information and Computation, 205(6):917–965, 2007.

[9] Maribel Fernández and Murdoch James Gabbay.
Closed nominal rewriting and efficiently computable nominal algebra equality.
Electronic Proceedings in Theoretical Computer Science, 34:37–51, 2010.

[10] Maribel Fernández, Murdoch James Gabbay, and Ian Mackie.
Nominal rewriting systems.
Proceedings of the 6th ACM SIGPLAN international conference on Principles and practice of declarative programming - PPDP '04, pages 108–119, 2004.

References IV

[11] Ramana Kumar and Michael Norrish.
(Nominal) Unification by Recursive Descent with Triangular Substitutions.
Interactive Theorem Proving, 2010.

[12] Andrew M. Pitts.
Nominal Logic : A First Order Theory of Names and Binding.
Fourth International Symposium on Theoretical Aspects of Computer Software (TACS2001), 2215:219–242, 2001.

[13] Ana Cristina Rocha-oliveira, Mauricio Ayala-Rincón, and Maribel Fernández.
Completeness in PVS of a Nominal Unification Algorithm.
In *LSFA*, volume 2015, pages 19–34, 2015.

[14] Mark R. Shinwell.
The Fresh Approach: functional programming with names and binders.
Technical report, University of Cambridge, 2005.

References V

[15] Mark R. Shinwell, Andrew M. Pitts, and Murdoch James Gabbay.
FreshML: Programming with binders made simple.
Proceedings of the eighth ACM SIGPLAN international conference on Functional programming (ICFP '03), pages 263–274, 2003.

[16] Christian Urban.
Nominal Techniques in Isabelle/HOL.
Journal of Automated Reasoning, 40(4):327–356, 2008.

[17] Christian Urban.
Nominal Unification Revisited.
24th International Workshop on Unification, UNIF 2010, pages 513–527, 2010.

[18] Christian Urban, Andrew M. Pitts, and Murdoch James Gabbay.
Nominal unification.
Theoretical Computer Science, 323(1-3):473–497, 2004.

Thank You ...