NOMINAL C-UNIFICATION

Mauricio Ayala-Rincón[†], **Washington L. R. de Carvalho Segundo**[†‡],

Maribel Fernández[‡] and Daniele Nantes Sobrinho[†]

[†]UNIVERSIDADE DE BRASÍLIA

[‡]KING'S COLLEGE LONDON

27^{th} Int. Sym. on Logic-Based Program Synthesis and Transformation — LOPSTR

Namur, 12 October 2017

Outline

1. Motivation
   - Unification problems
2. Formalisation of a nominal C-unification algorithm
   - Termination
   - Soundness
   - Completeness
3. Nominal C-unification is
   - Infinitary
   - NP-complete
4. Conclusion and future work

Unification

- Equations between first-order terms $s \approx_? t$ where variables $\{X, Y, Z, \ldots\}$ can be substituted by terms
- Application contexts:
  - logic programming
  - type inference
  - term rewriting
  - theorem provers
  - security protocol analysis
  - information retrieval

Equational and binding operators

- Function symbols f with basic equational properties like:
  - A $\{f(X, f(Y, Z)) \approx f(f(X, Y), Z)\}$
  - C $\{\mathbf{f}(\mathbf{X}, \mathbf{Y}) \approx \mathbf{f}(\mathbf{Y}, \mathbf{X})\}$
  - D $\{f(g(X, Y), Z) \approx g(f(X, Z), f(Y, Z))\}$
  - U $\{f(X, 1) \approx X\}$
- Bound object-level variables $[a]s$

$$\forall a \forall b, P(a) \vee Q(b) \Rightarrow R(a, b)$$

represented as

$$f_{\forall}[a]f_{\forall}[b]f_{\Rightarrow}(f_{\vee}(P(a), P(b)),\ R(a, b))$$

Nominal basic objects

- **atoms** $\{a, b, c, \ldots\}$ and variables $\{X, Y, Z, \ldots\}$
- **Freshness contexts** $\nabla = \{a\#X, b\#Y, c\#Z, \ldots\}$
- permutations as **lists of name-swappings**
  $\pi = (a_0\, b_0) :: (a_1\, b_1) :: \ldots :: (a_n\, b_n) :: nil$

  The inverse of π is its reverse list
  $\pi^{-1} = (a_n\, b_n) :: (a_{n-1}\, b_{n-1}) :: \ldots :: (a_0\, b_0) :: nil$

Nominal syntax and $\{\alpha, C\}$-equivalence

- **Nominal syntax**

$$t, u, v \ ::= \ \langle\rangle \ \mid \ \overline{a} \ \mid \ [a]t \ \mid \ \langle u, v\rangle \ \mid \ \mathbf{f}_{\mathbf{k}}^{\mathbf{E}}\, t \ \mid \ \pi.X$$

- **Freshness relation** $\nabla \vdash a \# t$

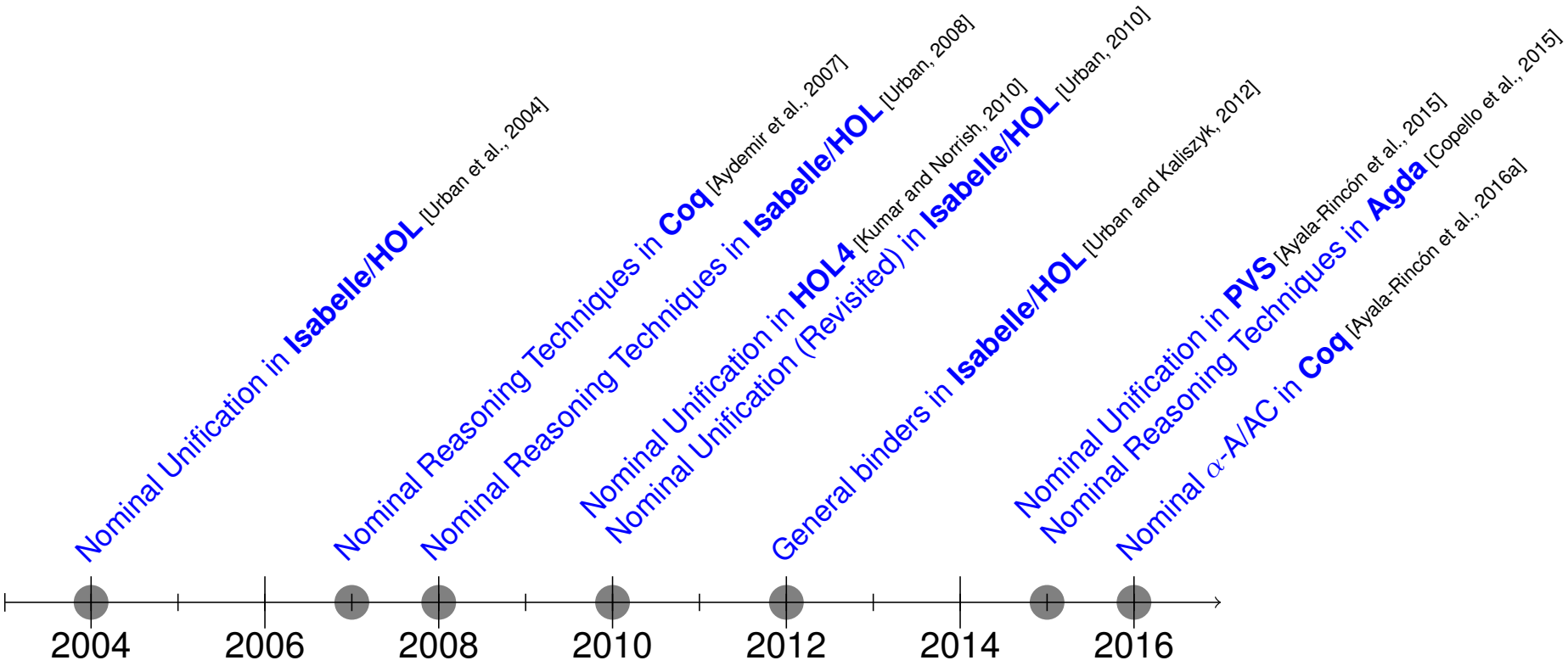
  a is fresh to t under the freshness context ∇

  $\nabla \vdash a \# \pi.X$ only if $(\pi^{-1} \cdot a) \# X \in \nabla$

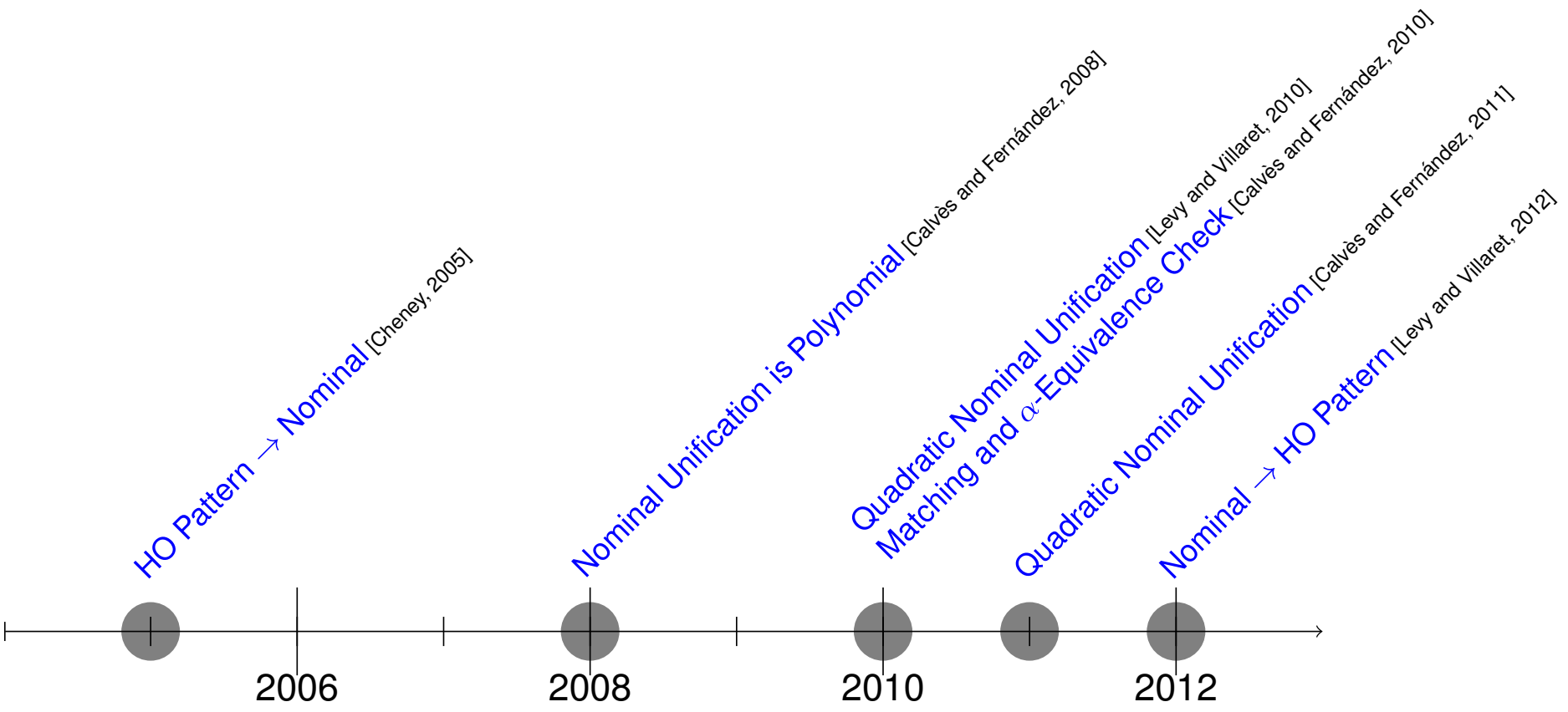
- $\{\alpha, \mathbf{C}\}$**-equivalence** $\nabla \vdash s \approx_{\{\alpha, C\}} t$

  $\{\alpha, E\}$-equivalence instantiated with $E = \{C\}$
  (see [Ayala-Rincón et al., 2016a])

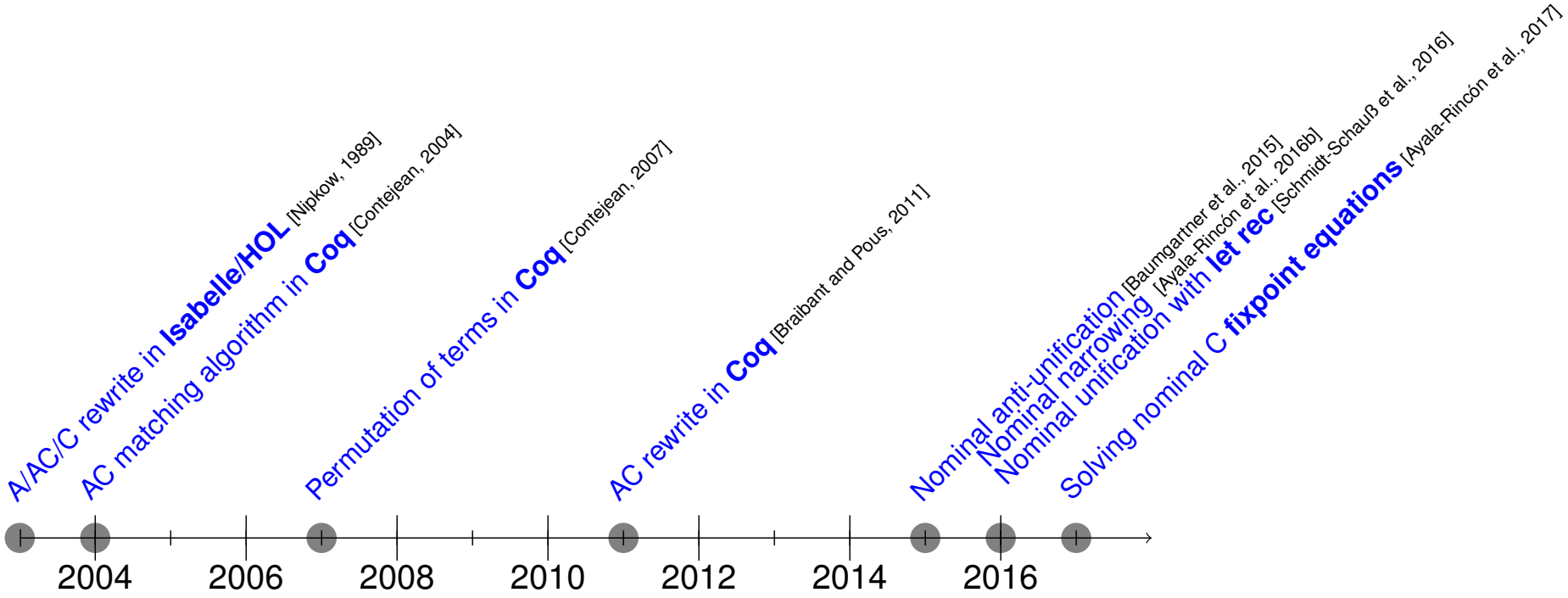
Related work: *formalisation on nominal*



Related work: *HO correspondence and efficiency of nominal algorithms*



Related work: *reasoning modulo equational theories and nominal unification extensions*



Unification problems

- **Unification problem** $\langle \nabla, P \rangle = \langle \nabla, id, P \rangle$
- P is a finite set of **equations** and **freshness constraints** of the form $\boxed{s \approx_? t}$ and $\boxed{a\#_? s}$

System $\Rightarrow_{\#}$

$$\frac{\langle \nabla, \sigma, P \uplus \{a\#_?\langle\rangle\}\rangle}{\langle \nabla, \sigma, P\rangle}\,(\#_?\langle\rangle) \qquad \frac{\langle \nabla, \sigma, P \uplus \{a\#_?\bar{b}\}\rangle}{\langle \nabla, \sigma, P\rangle}\,(\#_?\mathbf{a\bar{b}})$$

$$\frac{\langle \nabla, \sigma, P \uplus \{a\#_? f\ t\}\rangle}{\langle \nabla, \sigma, P \cup \{a\#_? t\}\rangle}\,(\#_?\mathbf{app}) \qquad \frac{\langle \nabla, \sigma, P \uplus \{a\#_?[a]t\}\rangle}{\langle \nabla, \sigma, P\rangle}\,(\#_?\mathbf{a[a]})$$

$$\frac{\langle \nabla, \sigma, P \uplus \{a\#_?[b]t\}\rangle}{\langle \nabla, \sigma, P \cup \{a\#_? t\}\rangle}\,(\#_?\mathbf{a[b]}) \qquad \frac{\langle \nabla, \sigma, P \uplus \{a\#_?\pi.X\}\rangle}{\langle \{(\pi^{-1}\cdot a)\#X\} \cup \nabla, \sigma, P\rangle}\,(\#_?\mathbf{var})$$

$$\frac{\langle \nabla, \sigma, P \uplus \{a\#_?\langle s, t\rangle\}\rangle}{\langle \nabla, \sigma, P \cup \{a\#_? s, a\#_? t\}\rangle}\,(\#_?\mathbf{pair})$$

System $\Rightarrow_{\approx}$

$$\frac{\langle \nabla, \sigma, P \uplus \{s \approx_? s\}\rangle}{\langle \nabla, \sigma, P\rangle}\ (\approx_? \textbf{refl}) \qquad \frac{\langle \nabla, \sigma, P \uplus \{\langle s_1, t_1\rangle \approx_? \langle s_2, t_2\rangle\}\rangle}{\langle \nabla, \sigma, P \cup \{s_1 \approx_? s_2, t_1 \approx_? t_2\}\rangle}\ (\approx_? \textbf{pair})$$

$$\frac{\langle \nabla, \sigma, P \uplus \{f_k^E\ s \approx_?\ f_k^E\ t\}\rangle}{\langle \nabla, \sigma, P \cup \{s \approx_?\ t\}\rangle}\text{, if } E \neq C\ (\approx_? \textbf{app})$$

$$\frac{\langle \nabla, \sigma, P \uplus \{f_k^C\ s \approx_?\ f_k^C\ t\}\rangle}{\langle \nabla, \sigma, P \cup \{s \approx_?\ v\}\rangle}\text{, } \left\{ \begin{array}{l} \text{where } s = \langle s_0, s_1\rangle \text{ and } t = \langle t_0, t_1\rangle \\ v = \langle t_i, t_{(i+1)\ mod\ 2}\rangle, i = 0, 1 \end{array} \right\} (\approx_? \textbf{C})$$

$$\frac{\langle \nabla, \sigma, P \uplus \{[a]s \approx_?\ [a]t\}\rangle}{\langle \nabla, \sigma, P \cup \{s \approx_?\ t\}\rangle}\ (\approx_? [\textbf{aa}]) \qquad \frac{\langle \nabla, \sigma, P \uplus \{[a]s \approx_?\ [b]t\}\rangle}{\langle \nabla, \sigma, P \cup \{s \approx_?\ (a\ b)\ t, a\#_? t\}\rangle}\ (\approx_? [\textbf{ab}])$$

$$\frac{\langle \nabla, \sigma, P \uplus \{\pi.X \approx_?\ t\}\rangle \text{ let } \sigma' := \sigma\{X/\pi^{-1} \cdot t\}}{\left\langle \nabla, \sigma', P\{X/\pi^{-1} \cdot t\} \cup \bigcup_{\substack{Y \in dom(\sigma'),\\ a\#Y \in \nabla}} \{a\#_? Y\sigma'\} \right\rangle}\text{, if } X \notin Var(t)\ (\approx_? \textbf{inst})$$

$$\frac{\langle \nabla, \sigma, P \uplus \{\pi.X \approx_?\ \pi'.X\}\rangle}{\langle \nabla, \sigma, P \cup \{\pi \oplus (\pi')^{-1}.X \approx_?\ X\}\rangle}\text{, if } \pi' \neq \mathtt{id}\ (\approx_? \textbf{inv})$$

Derivation tree for $\langle \nabla, P \rangle$

Nodes are labelled w.r.t. each $\Rightarrow_{\approx}$(resp. $\Rightarrow_{\#}$)-derivation step

1. The root node is labelled with $\mathcal{P} = \langle \nabla, id, P \rangle$
2. $\mathcal{P}$ is reduced by $\Rightarrow_{\approx}$ (for each branch), until reach $\mathcal{Q}_i$ (a normal form w.r.t. $\Rightarrow_{\approx}$)
3. For each $\mathcal{Q}_i = \langle \nabla_i, \delta_i, Q_i \rangle$. If Q_i contains only **fixpoint equations** and **freshness contraints** then it is reduced until reach $\bar{\mathcal{Q}}$ (a normal form w.r.t. $\Rightarrow_{\#}$)

Derivation tree for $\langle \nabla, P \rangle$

Nodes are labelled w.r.t. each $\Rightarrow_{\approx}$(resp. $\Rightarrow_{\#}$)-derivation step

1. The root node is labelled with $\mathcal{P} = \langle \nabla, id, P \rangle$
2. $\mathcal{P}$ is reduced by $\Rightarrow_{\approx}$ (for each branch), until reach $\mathcal{Q}_i$ (a normal form w.r.t. $\Rightarrow_{\approx}$)
3. For each $\mathcal{Q}_i = \langle \nabla_i, \delta_i, Q_i \rangle$. If Q_i contains only **fixpoint equations** and **freshness contraints** then it is reduced until reach $\bar{\mathcal{Q}}$ (a normal form w.r.t. $\Rightarrow_{\#}$)

Derivation tree for $\langle \nabla, P \rangle$

Nodes are labelled w.r.t. each $\Rightarrow_{\approx}$(resp. $\Rightarrow_{\#}$)-derivation step

1. The root node is labelled with $\mathcal{P} = \langle \nabla, id, P \rangle$
2. $\mathcal{P}$ is reduced by $\Rightarrow_{\approx}$ (for each branch),
   until reach $\mathcal{Q}_i$ (a normal form w.r.t. $\Rightarrow_{\approx}$)
3. For each $\mathcal{Q}_i = \langle \nabla_i, \delta_i, Q_i \rangle$.
   If Q_i contains only **fixpoint equations** and **freshness contraints**
   then it is reduced until reach $\bar{\mathcal{Q}}$ (a normal form w.r.t. $\Rightarrow_{\#}$)

Definition of a solution

Definition (Solution for a triple or a problem)

A **solution** for a triplet $\mathcal{P} = \langle \nabla, \delta, P \rangle$ is given by a pair $\langle \Delta, \sigma \rangle$ that satisfies

1. $\forall\, a\#X \in \nabla,\ \Delta \vdash a \,\#\, X\sigma$
2. if $a\#_? t \in P$ then $\Delta \vdash a \,\#\, t\sigma$
3. if $s \approx_? t \in P$ then $\Delta \vdash s\sigma \approx_{\{\alpha, C\}} t\sigma$
4. $\exists \lambda$ s.t. $\forall X \in dom(\delta\lambda) \cup dom(\sigma), \Delta \vdash X\delta\lambda \approx_{\{\alpha, C\}} X\sigma$

The **solution set** for a problem or triple $\mathcal{P}$ is denoted by $\mathcal{U}_C(\mathcal{P})$.

Example

Nominal unification
[Urban et al., 2004]

$$\mathcal{P} = \langle \emptyset, \{[a][b]X \approx_? [b][a]X\} \rangle$$
$$\vdots$$
$$\langle \emptyset, \{\mathbf{X} \approx_? (\mathbf{a}\,\mathbf{b}).\mathbf{X}\} \rangle$$
$$|$$
$$\langle \{a\#X, b\#X\}, id \rangle$$

In general $\langle\ \emptyset,\ \{\pi.\mathbf{X} \approx \mathbf{X}\}\ \rangle \Longrightarrow \langle\ dom(\pi)\#X,\ id\ \rangle$

Example

Nominal unification
[Urban et al., 2004]

$$\mathcal{P} = \langle \emptyset, \{[a][b]X \approx_? [b][a]X\} \rangle$$
$$\vdots$$
$$\langle \emptyset, \{\mathbf{X} \approx_? (\mathbf{a}\,\mathbf{b}).\mathbf{X}\} \rangle$$
$$|$$
$$\langle \{a\#X, b\#X\}, id \rangle$$

In general $\langle\ \emptyset,\ \{\pi.\mathbf{X} \approx \mathbf{X}\}\ \rangle \Longrightarrow \langle\ dom(\pi)\#X,\ id\ \rangle$

Example

$$\mathcal{P} = \langle \emptyset, \{ [e](a\,b).X * Y \approx_? [f](a\,c)(c\,d).X * Y \} \rangle$$

$\Rightarrow_{(\approx_? [\mathbf{ab}])}$

$$\langle \emptyset, id, \{ (a\,b).X * Y \approx_? (a\,c)(c\,d)(e\,f).X * (e\,f).Y \,,\; e\#_?(a\,c)(c\,d).X * Y \} \rangle$$

$\Rightarrow_{(\approx_? \mathbf{C})}$ branch 1

$\Rightarrow_{(\approx_? \mathbf{C})}$ branch 2

Example: branch 1

$$\langle \emptyset, id, \{ (a\,b).X \approx_? (a\,c)(c\,d)(e\,f).X,\ Y \approx_? (e\,f).Y,\ e\#_?(a\,c)(c\,d).X * Y \}\rangle$$

$$\Rightarrow_{(\approx_? \mathbf{inv})(2\times)}$$

$$\langle \emptyset, id, \{(a\,b)[(a\,c)(c\,d)(e\,f)]^{-1}.X \approx_? X,\ [(e\,f)]^{-1}.Y \approx_? Y,\ e\#_?(a\,c)(c\,d).X * Y \}\rangle$$

$$\Rightarrow \begin{array}{l}(\#_? \mathbf{app}),\\ (\#_? \mathbf{pair})\end{array}$$

$$\langle \emptyset, id, \{(a\,b)(e\,f)(c\,d)(a\,c).X \approx_? X, (e\,f).Y \approx_? Y,\ e\#_?(a\,c)(c\,d).X,\ e\#_? Y \}\rangle$$

$$\Rightarrow_{(\#_? \mathbf{var})(2\times)}$$

$$\langle \boxed{e\#X}, \boxed{e\#Y}, id, \{ (a\,b)(e\,f)(c\,d)(a\,c).X \approx_? X,\ (e\,f).Y \approx_? Y \}\rangle = \mathcal{Q}_1$$

Example: branch 2

$$\langle \emptyset, id, \{ (a\,b).X \approx_? (e\,f).Y,\ Y \approx_? \pi_1.X,\ e\#_?(a\,c)(c\,d).X * Y \} \rangle$$

$$\Rightarrow_{(\approx_? \mathbf{inst})}$$

$$\langle \emptyset, \{X/(e\,f)(a\,b).Y\}, \{ Y \approx_? (a\,c)(c\,d)(e\,f)(e\,f)[(a\,b)]^{-1}.Y,\ e\#_?(a\,c)(c\,d)(e\,f)[(a\,b)]^{-1}.Y * Y \} \rangle$$

$$\Rightarrow_{(\approx_? \mathbf{inv})}$$

$$\langle \emptyset, \{X/(e\,f)(a\,b).Y\}, \{[(a\,c)(c\,d)(e\,f)(e\,f)(a\,b)]^{-1}.Y \approx_? Y,\ e\#_?(a\,c)(c\,d)(e\,f)(a\,b).Y * Y \} \rangle$$

$$\Rightarrow \begin{array}{l}(\#_? \mathbf{app}),\\ (\#_? \mathbf{pair})\end{array}$$

$$\langle \emptyset, \{X/(e\,f)(a\,b).Y\}, \{(a\,b)(e\,f)(e\,f)(c\,d)(a\,c).Y \approx_? Y,\ e\#_?(a\,c)(c\,d)(e\,f)(a\,b).Y,\ e\#_? Y \} \rangle$$

$$\Rightarrow_{(\#_? \mathbf{var})(2\times)}$$

$$\langle \boxed{e\#Y}, \boxed{f\#Y}, \{X/(e\,f)(a\,b).Y\}, \{ (a\,b)(e\,f)(e\,f)(c\,d)(a\,c).Y \approx_? Y \} \rangle = \mathcal{Q}_2$$

Termination

Lemma (Termination of $\Rightarrow_{\approx}$ and $\Rightarrow_{\#}$)

There is no infinite chain of reductions $\Rightarrow_{\approx}$ (or $\Rightarrow_{\#}$) starting from an arbitrary triple $\mathcal{P} = \langle \nabla, \sigma, P \rangle$.

The proof is by well-founded induction on $\mathcal{P}$

$$\|P\| = \sum_{s \approx_? t \in P_{\approx}} |s| + |t| + \sum_{a \#_? u \in P_{\#}} |u|$$

1. $\|\mathcal{P}_{\approx}\| = \langle |Var(P_{\approx})|, \|P_{\approx}\|, |P_{\mathsf{nfp}_{\approx}}| \rangle$
2. $\|\mathcal{P}_{\#}\| = \|P_{\#}\|$

Soundness

Theorem (Soundness of $\mathcal{T}_{\langle \nabla, P \rangle}$)

If $\mathcal{P}' = \langle \nabla', \sigma, P' \rangle$ is the label of a leaf in $\mathcal{T}_{\langle \nabla, P \rangle}$, then

1. $\mathcal{U}_C(\mathcal{P}') \subseteq \mathcal{U}_C(\langle \nabla, id, P \rangle)$ *and*
2. *if* $P'_{fp_{\approx}} \neq P'$ *then* $\mathcal{U}_C(\mathcal{P}') = \emptyset$

- Induction on $\Rightarrow_{\approx}$ and $\Rightarrow_{\#}$
- Non trivial cases: $\Rightarrow_{(\approx_? [\mathbf{ab}])}$ and $\Rightarrow_{(\approx_? \mathbf{inst})}$

Completeness

Theorem (Completeness of $\mathcal{T}_{\langle \nabla, P \rangle}$)

Let $\langle \nabla, P \rangle$ and $\mathcal{T}_{\langle \nabla, P \rangle}$ be a unification problem and its derivation tree.
Then $\mathcal{U}_C(\langle \nabla, id, P \rangle) = \bigcup_{\mathcal{Q} \in ST(\mathcal{T}_{\langle \nabla, P \rangle})} \mathcal{U}_C(\mathcal{Q})$.

- Induction on $\Rightarrow_{\approx}$ and $\Rightarrow_{\#}$
- Non trivial cases: $\Rightarrow_{(\approx_? \mathbf{C})}$, $\Rightarrow_{(\approx_? [\mathbf{ab}])}$ and $\Rightarrow_{(\approx_? \mathbf{inst})}$

Atom permutations as k-cycles

$$\mathcal{Q}_1 = \langle e\#X, e\#Y, id, \{ (a\,b)(e\,f)(c\,d)(a\,c).X \approx_? X \,,\; (e\,f).Y \approx_? Y \} \rangle$$

$$\mathcal{Q}_2 = \langle e\#Y, f\#Y, \{X/(e\,f)(a\,b).Y\}, \{ (a\,b)(e\,f)(e\,f)(c\,d)(a\,c).Y \approx_? Y \} \rangle$$

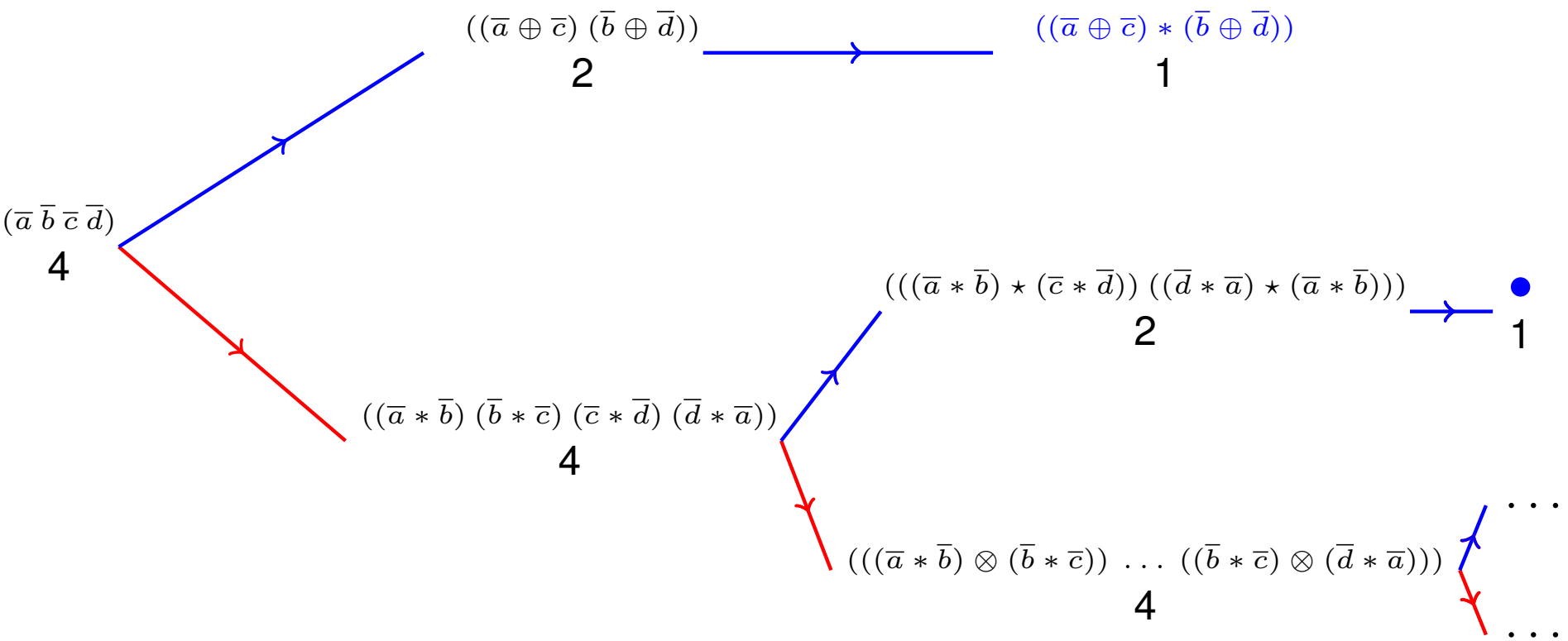
- $(a\,b)(e\,f)(c\,d)(a\,c) \longrightarrow (a\,b\,c\,d)(e\,f)$
- $(a\,b)(e\,f)(e\,f)(c\,d)(a\,c) \longrightarrow (a\,b\,c\,d)$

Pseudo-cycles

$\kappa = (a_0\ a_1\ \ldots\ a_{k-1}) \in \pi$

1. $\overline{\kappa} = (\overline{a_0} \cdots \overline{a_{k-1}})$ is a *(trivial) pseudo-cycle* w.r.t. κ
2. $\kappa' = (A_0\ \ldots A_{k'-1})$ is a *pseudo-cycle* w.r.t. κ, if
   1. each element of κ' is of the form $B_i * B_j$, B_i, B_j are different elements of κ'', a *pseudo-cycle* w.r.t. κ
   2. $A_i \not\approx_{\alpha,C} A_j$ for $i \neq j$, $0 \leq i, j \leq k' - 1$
   3. for each $0 \leq i < k' - 1$, $\kappa \cdot A_i \approx_{\{\alpha,C\}} A_{(i+1) mod\, k'}$

Example: $\kappa = (a\, b\, c\, d)$



Results

Theorem (Power of two cycles)

A pseudo-cycle κ generates unitary pseudo-cycles iff $|\kappa| = 2^p$.

Theorem (Combinatory solutions)

Let $\mathcal{P} = \langle \emptyset, \{\pi.X \approx_? X\} \rangle$ be a fixpoint problem. $\mathcal{P}$ has a combinatory solution iff there exists a unitary pseudo-cycle κ w.r.t. π.

Unitary pseudo-cycles as combinatory solutions

- If (s) is a unitary pseudo-cycle of κ then $\kappa \cdot s \approx_{\{\alpha,C\}} s$
- In our example:
  - $\langle e\#X, e\#Y, \{X/(\overline{a} \oplus \overline{c}) * (\overline{b} \oplus \overline{d})\}\rangle \in \mathcal{U}_C(\mathcal{Q}_1)$
  - $\langle e\#X, f\#Y, \{X/(e\,f)(a\,b).Y\}\{Y/(\overline{a} \oplus \overline{c}) * (\overline{b} \oplus \overline{d})\}\rangle \in \mathcal{U}_C(\mathcal{Q}_2)$

Extended pseudo-cycles

In [Ayala-Rincón et al., 2017] we defined **extended pseudo-cycles** to encompass all feasible combinatory solutions

Example

epc's w.r.t $(a\,b\,c\,d)$ include other syntactic elements

$s' = (h\,\overline{a} \oplus h\,\overline{c}) * (h\,\overline{b} \oplus h\,\overline{d})$ and $t' = ([i]\overline{a} \oplus [i]\overline{c}) * ([i]\overline{b} \oplus [i]\overline{d})$

So that $\{X/s'\}$ and $\{X/t'\}$ (resp. $\{Y/s'\}$ and $\{Y/t'\}$) are solutions of

$(a\ b\ c\ d)(e\ f).X \approx_? X$ (resp. $(a\ b\ c\ d).Y \approx_? Y$)

NP-completeness

1. Decide, for a given $\mathcal{P}$, if $\mathcal{U}_C(\mathcal{P}) \neq \emptyset$
   - Guessing *non-deterministically* a path to a successful leaf
2. NP-completeness: positive 1-in-3-SAT reduces to nominal C-unification
   - $\mathcal{C} = \{\mathcal{C}_i | 1 \leq i \leq n\}$ where $\mathcal{C}_i = p_i \vee q_i \vee r_i$
   - $\oplus$ commutative
   - $\begin{cases} a & \rightarrow & True \\ b & \rightarrow & False \end{cases}$
   - Each clause $\mathcal{C}_i = p_i \vee q_i \vee r_i$ in $\mathcal{C}$ is polynomially translated into

$$((X_{p_i} \oplus X_{q_i}) \oplus X_{r_i}) \oplus Y_i \approx_? ((\overline{b} \oplus \overline{b}) \oplus \overline{a}) \oplus ((\overline{b} \oplus \overline{a}) \oplus \overline{b})$$

Conclusion

- Formalisation of a nominal C-unification algorithm
  1. **termination**
  2. **soundness**
  3. **completeness**
- **Nominal C-unification** is
  1. **Infinitary**
  2. **NP-Complete**

Future work

1. Implementantion
2. Nominal
   - A / AC / C — matching
   - AC — unification
   - AC / C — narrowing

Future work

1. Implementantion (in progress)
2. Nominal
   - A / AC / C — matching (in progress)
   - AC — unification
   - AC / C — narrowing

THANK YOU

References I

Ayala-Rincón, M., Carvalho-Segundo, W., Fernández, M., and Nantes-Sobrinho, D. (2016a).

A Formalisation of Nominal Equivalence with Associative-Commutative Function Symbols.
In *Proc. of the 11th Workshop on Logical and Semantic Frameworks with Applications (LSFA)*, volume 332 of *ENTCS*, pages 21–38. Elseviser.

Ayala-Rincón, M., Carvalho-Segundo, W., Fernández, M., and Nantes-Sobrinho, D. (2017).

On Solving Nominal Fixpoint Equations.
In *Proc. of the 11th Int. Symp. on Frontiers of Combining Systems (FroCoS)*, volume 10483 of *LNCS*, pages 209–226. Springer.

Ayala-Rincón, M., Fernández, M., and Nantes-Sobrinho., D. (2016b).

Nominal Narrowing.
In *Proc. of the 1st Int. Conf. on Formal Structures for Computation and Deduction (FSCD)*, volume 52 of *LIPIcs*, pages 11:1–11:17.

Ayala-Rincón, M., Fernández, M., and Rocha-oliveira, A. C. (2015).

Completeness in PVS of a Nominal Unification Algorithm.
In *Proc. of the 10th Workshop on Logical and Semantic Frameworks with Applications (LSFA)*, volume 323 of *ENTCS*, pages 57–74. Elseviser.

Aydemir, B., Bohannon, A., and Weirich, S. (2007).

Nominal Reasoning Techniques in Coq.
ENTCS, 174(5):69–77.

Baumgartner, A., Kutsia, T., Levy, J., and Villaret, M. (2015).

Nominal Anti-Unification.
In *Proc. of the 26th Int. Conf. on Rewriting Techniques and Applications, (RTA)*, volume 36 of *LIPics*, pages 57–73.

References II

Braibant, T. and Pous, D. (2011).

Tactics for Reasoning Modulo AC in Coq.
In *In Proc. of the 1st. Int. Conf. on Certified Programs and Proofs (CPP)*, volume 7086 of *LNCS*, pages 167–182. Springer.

Calvès, C. F. and Fernández, M. (2008).

A Polynomial Nominal Unification Algorithm.
Theoretical Computer Science, 403(2-3):285–306.

Calvès, C. F. and Fernández, M. (2010).

Matching and Alpha-Equivalence Check for Nominal Terms.
J. of Computer and System Sciences, 76(5):283–301.

Calvès, C. F. and Fernández, M. (2011).

The First-order Nominal Link.
In *Proc. of the 20th Int. Symp. Logic-based Program Synthesis and Transformation (LOPSTR)*, volume 6564 of *LNCS*, pages 234–248. Springer.

Cheney, J. (2005).

Relating nominal and higher-order pattern unification.
In *Proc. of the 19th int. Workshop on Unification (UNIF)*, LORIA, pages 104–119.

Contejean, E. (2004).

A Certified AC Matching Algorithm.
In *Proc. of the 15th Int. Conf. on Rewriting Techniques and Applications (RTA)*, volume 3091 of *LNCS*, pages 70–84. Springer.

References III

Contejean, E. (2007).

Modeling permutations in coq for coccinelle.
In *Rewriting, Computation and Proof, Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of His 60th Birthday*, LNCS, pages 259–269. Springer.

Copello, E., Tasistro, A., Szasz, N., Bove, A., and Fernández, M. (2015).

Principles of Alpha-Induction and Recursion for the Lambda Calculus in Constructive Type Theory.
In *Logical and Semantic Frameworks with Applications*, pages 51–66.

Kumar, R. and Norrish, M. (2010).

(Nominal) Unification by Recursive Descent with Triangular Substitutions.
In *Proc. of Interactive Theorem Proving, 1st Int. Conf. (ITP)*, volume 6172 of *LNCS*, pages 51–66. Springer.

Levy, J. and Villaret, M. (2010).

An Efficient Nominal Unification Algorithm.
In *Proc. of the 21st Int. Conf. on Rewriting Techniques and Applications (RTA)*, volume 6 of *LIPIcs*, pages 209–226.

Levy, J. and Villaret, M. (2012).

Nominal unification from a higher-order perspective.
ACM Trans. Comput. Log., 13(2):10:1–10:31.

Nipkow, T. (1989).

Equational Reasoning in Isabelle.
Science of Computer Programming, 12(2):123–149.

References IV

Schmidt-Schauß, M., Kutsia, T., Levy, J., and Villaret, M. (2016).

Nominal Unification of Higher Order Expressions with Recursive Let.
In *Proc. of the 26th Int. Sym. on Logic-Based Program Synthesis and Transformation (LOPSTR)*, volume 10184 of *LNCS*, pages 328–344. Springer.

Urban, C. (2008).

Nominal Techniques in Isabelle/HOL.
J. of Autom. Reasoning, 40(4):327–356.

Urban, C. (2010).

Nominal Unification Revisited.
In *Proc. of the 24th Int. Work. on Unification (UNIF)*, volume 42 of *EPTCS*, pages 1–11.

Urban, C. and Kaliszyk, C. (2012).

General Bindings an Alpha-Equivalence in Nominal Isabelle.
Logical Methods in Computer Science, 8:1–35.

Urban, C., Pitts, A. M., and Gabbay, M. J. (2004).

Nominal Unification.
Theoretical Computer Science, 323(1-3):473–497.