An Intersection Type System for Nominal Terms

Mauricio Ayala-Rincón
Maribel Fernández
Ana Cristina Rocha-Oliveira

Grupo de Teoria da Computação - UnB - Brasília - Brasil
Department of Informatics - KCL - London - UK

Work supported by CAPES/Brazil

29th January, 2016
Seminário Informal (, mas Formal!), Brasília/DF, Brasil

What is nominal good for?

- Deal with binders in an elegant way.
- Built in α-equivalence.
- First-order substitutions.
- Decidable and efficient unification/matching.
- Frameworks based on nominal setting: α-Prolog, Fresh ML, Cαlm...

Specifying binding operations:

- Explicit substitutions:

$$M\{x \mapsto N\} \quad \rightarrow \quad M \qquad (x \notin fv(M))$$

Specifying binding operations:

- Explicit substitutions:

$$M\{x \mapsto N\} \quad \to \quad M \qquad (x \notin fv(M))$$

- π-calculus:

$$P \mid \nu a.Q \quad \to \quad \nu a.(P \mid Q) \qquad (a \notin fv(P))$$

Specifying binding operations:

- Explicit substitutions:

$$M\{x \mapsto N\} \quad \to \quad M \qquad (x \notin fv(M))$$

- π-calculus:

$$P \mid \nu a.Q \quad \to \quad \nu a.(P \mid Q) \qquad (a \notin fv(P))$$

- Logic:

$$P \text{ and } (\forall x.Q) \quad \to \quad \forall x.(P \text{ and } Q) \qquad (x \notin fv(P))$$

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

α-equivalence deduction rules:

$$\frac{}{\nabla \vdash a \approx_\alpha a}\ (\approx_\alpha \mathtt{a}) \qquad \frac{ds(\pi, \pi')\#X \subseteq \nabla}{\nabla \vdash \pi \cdot X \approx_\alpha \pi' \cdot X}\ (\approx_\alpha \mathtt{X})$$

$$\frac{\nabla \vdash s_1 \approx_\alpha t_1 \ \ldots \ \nabla \vdash s_n \approx_\alpha t_n}{\nabla \vdash f(s_1, \ldots, s_n) \approx_\alpha f(t_1, \ldots, t_n)}\ (\approx_\alpha \mathtt{f})$$

$$\frac{\nabla \vdash s \approx_\alpha t}{\nabla \vdash [a]s \approx_\alpha [a]t}\ (\approx_\alpha \mathtt{absa}) \qquad \frac{\nabla \vdash s \approx_\alpha (a\ b) \cdot t \quad \nabla \vdash a\#t}{\nabla \vdash [a]s \approx_\alpha [b]t}\ (\approx_\alpha \mathtt{absb})$$

where $\nabla = \{a_1\#X_1, a_2\#X_2, \ldots, a_n\#X_n\}$.

Freshness deduction rules:

$$\frac{}{\nabla \vdash a\#\overline{b}}\ (\#\mathtt{ab}) \qquad \frac{(\pi^{-1}(a), X) \in \nabla}{\nabla \vdash a\#\pi \cdot X}\ (\#\mathtt{X})$$

$$\frac{}{\nabla \vdash a\#()}\ (\#\mathtt{unit}) \qquad \frac{\nabla \vdash a\#s}{\nabla \vdash a\#f\ s}\ (\#\mathtt{f})$$

$$\frac{\nabla \vdash a\#s_1 \qquad \nabla \vdash a\#s_2}{\nabla \vdash a\#(s_1, s_2)}\ (\#\mathtt{pair})$$

$$\frac{}{\nabla \vdash a\#[a]s}\ (\#\mathtt{absa}) \qquad \frac{\nabla \vdash a\#s}{\nabla \vdash a\#[b]s}\ (\#\mathtt{absb})$$

Nominal Rewriting Systems

$$\nabla \vdash l \to r \qquad Vars(r, \nabla) \subseteq Vars(l)$$

$$\begin{array}{lrcl}
(\text{Beta}): & & \vdash & (\lambda[a]X)X' \to X\{a \mapsto X'\} \\
(\sigma_{\mathtt{app}}): & & \vdash & (X\ X')\{a \mapsto Y\} \to X\{a \mapsto Y\}\ X'\{a \mapsto Y\} \\
(\sigma_{\mathtt{var}}): & & \vdash & a\{a \mapsto X\} \to X \\
(\sigma_{\epsilon}) : & a\#Y & \vdash & Y\{a \mapsto X\} \to Y \\
(\sigma_{\mathtt{lam}}): & b\#Y & \vdash & (\lambda[b]X)\{a \mapsto Y\} \to \lambda[b](X\{a \mapsto Y\})
\end{array}$$

Notation: $X\{a \mapsto Y\} = \mathtt{subst}([a]X, Y)$.

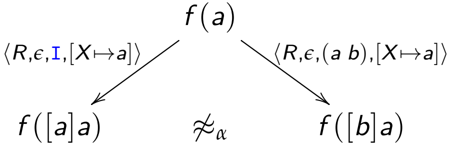
Unexpected problems:

Confluence of orthogonal NRSs does not hold in general.

Unexpected problems:

Confluence of orthogonal NRSs does not hold in general.

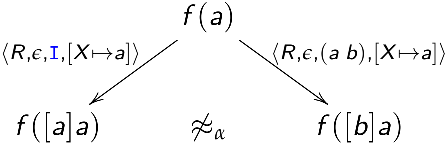
Example: $R \;=\; \vdash f(X) \to f([a]X)$



Unexpected problems:

Confluence of orthogonal NRSs does not hold in general.

Example: $R \;=\; \vdash f(X) \to f([a]X)$



Solution:

- Additional conditions (α-stability); or
- Transform the notion of rewriting (closed rewriting).

Unexpected problems:

Confluence of orthogonal NRSs does not hold in general.

Example: $R \;=\; \vdash f(X) \rightarrow f([a]X)$

$$f(a) \xrightarrow{\langle R,\epsilon,\mathtt{I},[X \mapsto a]\rangle} f([a]a) \qquad f(a) \xrightarrow{\langle R,\epsilon,(a\ b),[X \mapsto a]\rangle} f([b]a)$$

$$f([a]a) \quad \not\approx_{\alpha} \quad f([b]a)$$

Solution:

- Additional conditions (α-stability); or
- Transform the notion of rewriting (closed rewriting).

What about types?

Importance of Types

- Add formalism to programming languages.
- Prevent errors.
- Existing nominal type systems: simple, polymorphic and dependent type systems.

Grammar of types

- A set of type variables $\mathbb{T}$
- A set of type constructors as $\mathbb{T}_C$ (bool, nat, real, list etc)
- A signature Σ with function symbols and their corresponding type declarations $\sigma \hookrightarrow \tau$
- Types are given by

$$\tau ::= \beta \mid () \mid (\tau \times \tau) \mid \mathtt{C}\,\tau \mid [\tau]\tau \mid \tau \cap \tau.$$

Grammar of types

- A set of type variables $\mathbb{T}$
- A set of type constructors as $\mathbb{T}_C$ (bool, nat, real, list etc)
- A signature Σ with function symbols and their corresponding type declarations $\sigma \hookrightarrow \tau$
- Types are given by

$$\tau ::= \beta \mid () \mid (\tau \times \tau) \mid \mathtt{C}\,\tau \mid [\tau]\tau \mid \tau \cap \tau.$$

Partial order between types

$$(A1)\ \sigma \leq \sigma \qquad (A2)\ \sigma \cap \tau \leq \sigma \qquad (A3)\ \sigma \cap \tau \leq \tau$$

$$(\mathrm{R1})\ \frac{\sigma \leq \tau, \sigma \leq \rho}{\sigma \leq \tau \cap \rho} \qquad (\mathrm{R2})\ \frac{\sigma \leq \tau, \tau \leq \rho}{\sigma \leq \rho}$$

Type Inference Rules: Quasi-derivations

$$(\mathtt{a})\ \frac{}{\Gamma \Join a:\sigma, \Delta \vdash a:\sigma} \qquad (\mathtt{var})\ \frac{}{\Gamma \Join X:\sigma, \Delta \vdash \pi \cdot X:\sigma}$$

$$(\mathtt{abs})\ \frac{\Gamma \Join a:\sigma, \Delta \vdash t:\tau}{\Gamma, \Delta \vdash [a]t:[\sigma]\tau} \qquad (\cap_I)\ \frac{\Gamma, \Delta \vdash t:\sigma_1 \qquad \Gamma, \Delta \vdash t:\sigma_2}{\Gamma, \Delta \vdash t:\sigma_1 \cap \sigma_2}$$

$$(\cap_E)\ \frac{\Gamma, \Delta \vdash t:\sigma_1 \cap \sigma_2}{\Gamma, \Delta \vdash t:\sigma_i} \qquad (\times)\ \frac{\Gamma, \Delta \vdash t_1:\sigma_1 \quad \Gamma, \Delta \vdash t_2:\sigma_2}{\Gamma, \Delta \vdash \langle t_1, t_2\rangle : (\sigma_1 \times \sigma_2)}$$

$$(\times_0)\ \frac{}{\Gamma, \Delta \vdash \langle\ \rangle : (\ )} \qquad (\mathtt{f})\ \frac{\Sigma_f = \tau \hookrightarrow \gamma \quad \sigma \le \tau \quad \Gamma, \Delta \vdash t:\sigma}{\Gamma, \Delta \vdash f\ t:\gamma}$$

Type Inference Rules: Quasi-derivations

$$(\mathtt{a})\ \frac{}{\Gamma \bowtie a:\sigma, \Delta \vdash a:\sigma} \qquad (\mathtt{var})\ \frac{}{\Gamma \bowtie X:\sigma, \Delta \vdash \pi \cdot X:\sigma}$$

$$(\mathtt{abs})\ \frac{\Gamma \bowtie a:\sigma, \Delta \vdash t:\tau}{\Gamma, \Delta \vdash [a]t : [\sigma]\tau} \qquad (\cap_I)\ \frac{\Gamma, \Delta \vdash t:\sigma_1 \qquad \Gamma, \Delta \vdash t:\sigma_2}{\Gamma, \Delta \vdash t:\sigma_1 \cap \sigma_2}$$

$$(\cap_E)\ \frac{\Gamma, \Delta \vdash t:\sigma_1 \cap \sigma_2}{\Gamma, \Delta \vdash t:\sigma_i} \qquad (\times)\ \frac{\Gamma, \Delta \vdash t_1:\sigma_1 \qquad \Gamma, \Delta \vdash t_2:\sigma_2}{\Gamma, \Delta \vdash \langle t_1, t_2\rangle : (\sigma_1 \times \sigma_2)}$$

$$(\times_0)\ \frac{}{\Gamma, \Delta \vdash \langle\ \rangle : (\ )} \qquad (\mathtt{f})\ \frac{\Sigma_f = \tau \hookrightarrow \gamma \qquad \tau \leq \sigma \qquad \Gamma, \Delta \vdash t:\sigma}{\Gamma, \Delta \vdash f\ t:\gamma}$$

Definition (Essential environment)

For a quasi-derivation of $\Gamma', \Delta \vdash t : \tau$, let $\Gamma \Join X : \sigma, \Delta \vdash \pi \cdot X : \sigma$ be a leaf of it such that $X \in \mathit{Vars}(t)$. Thus, the set ${}^{\pi^{-1}}\Gamma \setminus \{a : \tau' | \Delta \vdash a\#X\}$ is an **essential environment** of the quasi-derivation with respect to X.

Definition (Essential environment)

For a quasi-derivation of $\Gamma', \Delta \vdash t : \tau$, let $\Gamma \rtimes X : \sigma, \Delta \vdash \pi \cdot X : \sigma$ be a leaf of it such that $X \in \mathit{Vars}(t)$. Thus, the set ${}^{\pi^{-1}}\Gamma \setminus \{a : \tau' | \Delta \vdash a \# X\}$ is an **essential environment** of the quasi-derivation with respect to X.

Definition (Diamond property)

A quasi-derivation of $\Gamma', \Delta \vdash t : \tau$ has the **diamond property** if, for each $X \in \mathit{Vars}(t)$, the essential environments with respect to X are equal.

Results

Lemma (Subtype property)

If $\Gamma, \Delta \vdash t : \tau$ *and* $\tau \leq \tau'$, *then* $\Gamma, \Delta \vdash t : \tau'$.

Results

Lemma (Subtype property)

If $\Gamma, \Delta \vdash t : \tau$ *and* $\tau \leq \tau'$*, then* $\Gamma, \Delta \vdash t : \tau'$*.*

Lemma (Object level equivariance)

If $\Gamma, \Delta \vdash t : \sigma$*, then* ${}^{\pi}\Gamma, \Delta \vdash \pi \cdot t : \sigma$*.*

Results

Lemma (Subtype property)

If $\Gamma, \Delta \vdash t : \tau$ *and* $\tau \leq \tau'$, *then* $\Gamma, \Delta \vdash t : \tau'$.

Lemma (Object level equivariance)

If $\Gamma, \Delta \vdash t : \sigma$, *then* ${}^{\pi}\Gamma, \Delta \vdash \pi \cdot t : \sigma$.

Lemma (α-equivalence preserves types)

If $\Gamma, \Delta \vdash t : \tau$ *and* $\Delta \vdash t \approx_{\alpha} s$, *then* $\Gamma, \Delta \vdash s : \tau$.

Typeability

Definition (Typeability problem)

A given term in context $\Delta \vdash t$ is a typeability problem that asks if there exist a solution $\langle \Gamma, \tau \rangle$ such that $\Gamma, \Delta \vdash t : \tau$.

Typeability

Definition (Typeability problem)

A given term in context $\Delta \vdash t$ is a typeability problem that asks if there exist a solution $\langle \Gamma, \tau \rangle$ such that $\Gamma, \Delta \vdash t : \tau$.

Definition (Principal typings)

A pair $\langle \Gamma, \tau \rangle$ is a principal typing of $\Delta \vdash t$ if it solves this typeability problem and, for any other solution $\langle \Gamma', \tau' \rangle$, $\Gamma' \leq \Gamma$ and $\tau \leq \tau'$ hold.

Definition (Typeable closed rules)

A typeable closed rule $\Phi, \nabla \vdash l \to r : \tau$ satisfies:

- $Vars(\Phi, \nabla, r) \subseteq Vars(l)$;
- $\langle \Phi, (\tau \times \tau) \rangle$ is a principal typing of $\nabla \vdash \langle l, r \rangle$.
- $\nabla \vdash l \to r$ is a closed rule (it matches a freshened version).
- If $\Sigma_f = \tau \hookrightarrow \gamma$ and $\Gamma, \Delta \vdash f\ t : \gamma$ occurs in the derivation of types, then $\Gamma, \Delta \vdash t : \sigma$ and $\tau \leq \sigma$.

Definition (Typeable closed rules)

A typeable closed rule $\Phi, \nabla \vdash l \rightarrow r : \tau$ satisfies:

- $Vars(\Phi, \nabla, r) \subseteq Vars(l)$;
- $\langle \Phi, (\tau \times \tau) \rangle$ is a principal typing of $\nabla \vdash \langle l, r \rangle$.
- $\nabla \vdash l \rightarrow r$ is a closed rule (it matches a freshened version).
- If $\Sigma_f = \tau \hookrightarrow \gamma$ and $\Gamma, \Delta \vdash f\ t : \gamma$ occurs in the derivation of types, then $\Gamma, \Delta \vdash t : \sigma$ and $\tau \leq \sigma$.

Lemma (Subject Reduction)

Given a typeable closed rule $\Phi, \nabla \vdash l \rightarrow r : \tau$, if $\Gamma, \Delta \vdash s : \sigma$ and $\Delta \vdash s \stackrel{R}{\rightarrow}_c t$, then $\Gamma, \Delta \vdash t : \sigma$.

Conclusion and Future Work

- We have a preliminary intersection type system for nominal terms that preserves types for α-equivalent terms.
- It is expected to develop an algorithm to return principal typings for terms in context.
- The conditions in which subject reduction (expansion) holds must be studied.

Elliot Fairweather, Maribel Fernández, and Murdoch James Gabbay.
Principal types for nominal theories.
In *Fundamentals of Computation Theory - 18th International Symposium, FCT 2011, Oslo, Norway, August 22-25, 2011. Proceedings*, pages 160–172, 2011.

Elliot Fairweather, Maribel Fernández, Nora Szasz, and Alvaro Tasistro.
Dependent types for nominal terms with atom substitutions.
In *13th International Conference on Typed Lambda Calculi and Applications, TLCA 2015, July 1-3, 2015, Warsaw, Poland*, pages 180–195, 2015.

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

Steffen van Bakel.
Rank 2 intersection type assignment in term rewriting systems.
Fundam. Inform., 26(2):141–166, 1996.

Steffen van Bakel and Mariangiola Dezani-Ciancaglini.
Characterising strong normalisation for explicit substitutions.
In *LATIN 2002: Theoretical Informatics, 5th Latin American Symposium, Cancun, Mexico, April 3-6, 2002, Proceedings*, pages 356–370, 2002.