A Dependent Type Theory with Names and Binding

Ulrich Schöpp Ian Stark

Laboratory for Foundations of Computer Science
University of Edinburgh

September 21, 2004

Names

- Everywhere: abstract syntax, π-calculus, references, ...
- What are names?
- This talk: Add names to dependent type theory.
- Why extend type theory with names?
- Why not use natural numbers? (e.g. $\lambda x_1 . x_5\ x_1$)

Names

Example: Formal reasoning about the λ-calculus

$$\mathsf{Lam} ::= \mathsf{var} \text{ of } \mathbf{N} \mid \mathsf{app} \text{ of } \mathsf{Lam} \times \mathsf{Lam} \mid \mathsf{lam} \text{ of } [\mathbf{N}]\mathsf{Lam}$$

- Common informal reasoning:

  $$(\exists n \text{ fresh name.}\, P(n)) \implies (\forall n \text{ fresh name.}\, P(n))$$

  Holds for any P because there is little we can do with names.

- Encoding names by numbers. [McKinna & Pollack] ...
  Above informal proof cannot be formalised directly because we can do much more with numbers than with names.
  [McKinna & Pollack] prove the implication for each P.

$\Rightarrow$ Extend type theory with names and make known that little can be done with names.

FM Set Theory

FM set theory contains a nice notion of names [Gabbay & Pitts].

Name-swapping, Syntax-independent freshness,
Freshness quantifier И, Abstraction $[\mathbf{N}]X$, Binding $n.x$,
Unique choice of fresh names $\mathrm{new}\ n.\, M$.

Our development is based on the properties of these constructions.

Names in Dependent Type Theory

Add FM-style names to dependent type theory

- Based on freshness

$$A * B = \{\langle x, y\rangle : A \times B \mid x \text{ and } y \text{ do not share names}\}$$

- Non-standard dependent sums Σ^* and products Π^*.
- Names and FM-style constructions (И, $[\mathbf{N}]B$, $n.x$, $\mathrm{new}\ n.\, M$) are expressed in terms of Σ^* and Π^*.
- FM-Sets models this structure.

Σ^* informally

Type	Elements
$\Sigma x : A.\, B$	$\langle M, N \rangle$ where $M : A$ and $N : B[M/x]$
$\Sigma^* x : A.\, B$	$M.N$ where $M : A$ and $N : B[M/x]$ and the names of M are hidden in $M.N$

When B does not depend on x, write $\begin{cases} A \times B \text{ for } \Sigma x : A.\, B. \\ [A]B \text{ for } \Sigma^* x : A.\, B. \end{cases}$

Examples

- $[\mathbf{N}]$Lam. α-equivalence classes for one name

$$n.(\mathsf{app}\ n\ m) = n'.(\mathsf{app}\ n'\ m) \qquad n,\, n' \text{ fresh for } m$$

- $\Sigma^* x : \mathbb{N}.\, B \;\cong\; \Sigma x : \mathbb{N}.\, B.$

Π^* informally

Type	Elements
$\Pi x : A.\, B$	function mapping $M : A$ to $B[M/x]$
$\Pi^* x : A.\, B$	partial function mapping $M : A$ containing only fresh names to $B[M/x]$

When B does not depend on x, write $\begin{cases} A \to B \text{ for } \Pi x : A.\, B. \\ A \twoheadrightarrow\!\!* \; B \text{ for } \Pi^* x : A.\, B. \end{cases}$

Examples

- $\mathbf{N} \multimap\!\!* \;$ Lam. α-equivalence classes for one name

$$n \mapsto (\mathsf{app}\ n\ m)$$

- $\mathbf{N} \to$ Lam contains more than α-equivalence classes.

$$n \mapsto \begin{cases} \mathsf{app}(n, n) & \text{if } n = m \\ \mathsf{lam}(n.n) & \text{otherwise} \end{cases}$$

- $\Pi^* x : \mathbb{N}.\, B \;\cong\; \Pi x : \mathbb{N}.\, B.$

Simple Sums Σ and Products Π

$\boxed{\Sigma x : A.\, B \text{ and } \Pi x : A.\, B \text{ for a simple type } A.}$

Simple weakening from $\mathbb{B}/\Gamma$ to $\mathbb{B}/(\Gamma \times A)$ is given by

$$\pi_A^* : \begin{array}{c} B \\ \downarrow f \\ \Gamma \end{array} \longmapsto \begin{array}{c} B \times A \\ \downarrow f\times A \\ \Gamma \times A \end{array} \qquad \begin{tikzcd} B \times A \arrow[r] \arrow[d, "f\times A"'] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "f"] \\ \Gamma \times A \arrow[r, "\pi_A"'] & \Gamma \end{tikzcd}$$

Simple products and sums are given by

$$\Sigma_A \dashv \pi_A^* \dashv \Pi_A$$

and are well-behaved under substitution

$$\Sigma_A(u \times A)^* \xrightarrow{\cong} u^*\Sigma_A\,, \qquad u^*\Pi_A \xrightarrow{\cong} \Pi_A(u \times A)^*\,.$$

Simple Monoidal Sums Σ^* and Products Π^*

$\boxed{\text{Replace } \times \text{ by } *.}$

Monoidal weakening from $\mathbb{B}/\Gamma$ to $\mathbb{B}/(\Gamma * A)$

$$W_A : \begin{tikzcd} B \arrow[d, "f"'] \\ \Gamma \end{tikzcd} \longmapsto \begin{tikzcd} B * A \arrow[d, "f*A"'] \\ \Gamma * A \end{tikzcd}$$

W_A interacts well with substitution if $- * A$ preserves pullbacks.
Simple monoidal products and sums are given by

$$\Sigma^*_A \dashv W_A \dashv \Pi^*_A,$$

$$\Sigma^*_A(u * A)^* \xrightarrow{\cong} u^*\Sigma^*_A\,, \qquad u^*\Pi^*_A \xrightarrow{\cong} \Pi^*_A(u * A)^*\,.$$

A Bunched Dependent Type Theory

Give a dependent type theory with

- additive types (Σ, Π, ...),
- multiplicative types ($\Sigma^*_A \dashv W_A \dashv \Pi^*_A$).

We use *bunches* [O'Hearn, Pym] to integrate the additive structure with the multiplicative structure.

$$\frac{}{\diamond\ \mathsf{Bunch}} \qquad \frac{\Gamma \vdash A\ \mathsf{Type}}{\Gamma,\, x : A\ \mathsf{Bunch}} \qquad \frac{\Gamma\ \mathsf{Bunch} \qquad \Delta\ \mathsf{Bunch}}{\Gamma * \Delta\ \mathsf{Bunch}}$$

Examples:

$$(x : \mathsf{Lam},\, y : \mathsf{Lam}) * z : \mathsf{Lam}$$

$$((x : A * y : B),\, z : C) * u : D$$

Simple Monoidal Products (Π^*_A)

$$\frac{\Gamma * x : A \vdash M : B}{\Gamma \vdash \lambda^* x : A.\, M : \Pi^* x : A.\, B} \qquad \frac{\Gamma \vdash M : \Pi^* x : A.\, B \qquad \Delta \vdash N : A}{\Gamma * \Delta \vdash M @ N : B\,[N/x]}$$

Freefrom Types (W_A)

The freefrom type $B^{*\,M:A}$ consists of all the elements of B that are *free from* the names in the term $M : A$.

$$W_A : \qquad \Gamma \vdash B \text{ Type} \qquad \longmapsto \qquad \Gamma * x : A \vdash B^{*\,x:A} \text{ Type}$$

Current restriction: Both A and B are closed in $B^{*\,M:A}$.

Simple Monoidal Sums (Σ^*_A)

$$\frac{x : A \vdash B \text{ Type} \qquad \Gamma \vdash M : A \qquad \Gamma \vdash N : B\,[M/x]}{\Gamma \vdash \mathrm{bind}(M, N) : (\Sigma^* x : A.\, B)^{*\,M:A}}$$

$$\frac{\Gamma \vdash M : \Sigma^* x : A.\, B \qquad (\Gamma * x : A),\, y : B \vdash N : C^{*\,x:A}}{\Gamma \vdash \mathrm{let}\ M\ \mathrm{be}\ x.y\ \mathrm{in}\ N : C}$$

Categories with Bindable Names

A *category with bindable names* $(\mathbb{B}, *, \mathbf{N})$ is given by

$$\begin{array}{rcl} \mathbb{B} & - & \text{Category (with pullbacks, coproducts,...)} \\ * & - & \text{Strict affine symmetric monoidal structure} \\ \mathbf{N} & - & \text{Object of } \mathbb{B} \end{array}$$

such that

- $\mathbf{N}$ has decidable equality and two names are fresh precisely when they are different.
- The functor $W_{\mathbf{N}}$ is an equivalence of fibrations.

$$\begin{tikzcd} \mathbb{B}^{\rightarrow} \arrow[rr, "W_{\mathbf{N}}"] \arrow[dr, "cod"'] & & \mathbb{B}/(- * \mathbf{N}) \arrow[dl, "Gl(-*\mathbf{N})"] \\ & \mathbb{B} & \end{tikzcd}$$

The Fibred Equivalence $W_{\mathbf{N}}$

- Equivalence of $\mathbb{B}/\Gamma$ and $\mathbb{B}/(\Gamma * \mathbf{N})$ for all Γ.

  "It doesn't matter whether we work with an α-equivalence class or with a concretely named instance."

- Captures name binding.
- Informal practice (Barendregt Variable Convention).

Bindable Names in terms of Σ^* and Π^*

$W_{\mathbf{N}}$ is a fibred equivalence if and only if we have

- fibred adjunctions $\Sigma^*_{\mathbf{N}} \dashv W_{\mathbf{N}} \dashv \Pi^*_{\mathbf{N}}$, and
- a vertical natural isomorphism $i\colon \Sigma^*_{\mathbf{N}} \to \Pi^*_{\mathbf{N}}$ with

$$
\begin{tikzcd}
W_{\mathbf{N}}\Sigma^*_{\mathbf{N}} \arrow[rr, "W_{\mathbf{N}} i"] & & W_{\mathbf{N}}\Pi^*_{\mathbf{N}} \arrow[dl, "\varepsilon'"] \\
& Id \arrow[ul, "\eta"] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\Sigma^*_{\mathbf{N}} W_{\mathbf{N}} \arrow[rr, "i_{W_{\mathbf{N}}}"] \arrow[dr, "\varepsilon"'] & & \Pi^*_{\mathbf{N}} W_{\mathbf{N}} \\
& Id \arrow[ur, "\eta'"'] &
\end{tikzcd}
$$

Use *hidden-name types* to add the isomorphism to the type theory.

$$\Sigma^* n : \mathbf{N}.\, B \quad \cong \quad \mathrm{H} n.\, B \quad \cong \quad \Pi^* n : \mathbf{N}.\, B$$

Rules and equations for H are those from both Σ^* and Π^* plus equations from the above triangles.

Hidden-Name Types: Unique Choice of Fresh Names

Unique choice of a fresh name

$$\frac{\Gamma * n : \mathbf{N} \vdash M : B^{* n:\mathbf{N}}}{\Gamma \vdash \text{new } n.\, M : B}$$

is derivable by

$$\frac{\dfrac{\Gamma * n : \mathbf{N} \vdash M : B^{* n:\mathbf{N}}}{\Gamma \vdash \lambda^* n.\, M : \mathrm{H}n.\, B^{* n:\mathbf{N}}} \qquad \dfrac{\vdots}{(\Gamma * n : \mathbf{N}),\, x : B^{* n:\mathbf{N}} \vdash x : B^{* n:\mathbf{N}}}}{\Gamma \vdash \text{let } (\lambda^* n.\, M) \text{ be } n.x \text{ in } x : B}$$

Example equation:

$$\frac{\Gamma \vdash M : B}{\Gamma \vdash (\text{new } n.\, M^{*n}) = M : B}$$

Hidden-Name Types: Substitution

$$\mathsf{Lam} ::= \mathsf{var}\ \mathrm{of}\ \mathbf{N} \mid \mathsf{app}\ \mathrm{of}\ \mathsf{Lam} \times \mathsf{Lam} \mid \mathsf{lam}\ \mathrm{of}\ \mathrm{H}n.\,\mathsf{Lam}$$

Substitution in weak HOAS-style (Π^*).

Define substitution of $R : \mathsf{Lam}$ for $m : \mathbf{N}$.

$$\begin{array}{rcl}
\mathsf{subst} & : & \mathsf{Lam} \to \mathsf{Lam} \\
\mathsf{subst}(\mathsf{var}(n)) & = & \mathrm{if}\ (n = m)\ \mathrm{then}\ R\ \mathrm{else}\ \mathsf{var}(n) \\
\mathsf{subst}(\mathsf{app}(M, N)) & = & \mathsf{app}(\mathsf{subst}(M), \mathsf{subst}(N)) \\
\mathsf{subst}(\mathsf{lam}(M)) & = & \mathsf{lam}(\lambda^* n.\,\mathsf{subst}(M@n))
\end{array}$$

Substitution in FM-style (Σ^*).

$\mathsf{subst}(\mathsf{lam}(M)) =$
$\quad \mathrm{let}\ M\ \mathrm{be}\ n.x\ \mathrm{in}\ (\mathrm{let}\ \mathrm{bind}(n, \mathsf{subst}(x))\ \mathrm{be}\ y^{*n}\ \mathrm{in}\ \mathsf{lam}(y)^{*n})$

Hidden-Name Types: Free Variables

$$\begin{array}{rcl} \mathsf{singleton} & : & \mathbf{N} \to \mathsf{L} \\ \mathsf{concat} & : & \mathsf{L} \to \mathsf{L} \to \mathsf{L} \\ \mathsf{remove} & : & \Pi n : \mathbf{N}.\ \big(\mathsf{L} \to \mathsf{L}^{*\, n:\mathbf{N}}\big) \end{array}$$

Free variables of a term in FM-style (Σ^*).

$$\begin{array}{rcl} \mathsf{fv} & : & \mathsf{Lam} \to \mathsf{L} \\ \mathsf{fv}(\mathsf{var}(n)) & = & \mathsf{singleton}(n) \\ \mathsf{fv}(\mathsf{app}(M, N)) & = & \mathsf{concat}(\mathsf{fv}(M), \mathsf{fv}(N)) \\ \mathsf{fv}(\mathsf{lam}(M)) & = & \mathrm{let}\ M\ \mathrm{be}\ n.x\ \mathrm{in}\ \mathsf{remove}(n, \mathsf{fv}(x)) \end{array}$$

Closest to wHOAS style (uses Σ^* in new):

$$\mathsf{fv}(\mathsf{lam}(M)) = \mathrm{new}\ n.\, \mathsf{remove}(n, \mathsf{fv}(M@n))$$

Conclusion and Further Work

Conclusion

Σ^* and Π^* are good for names!

With Σ^* and Π^* we can

- define names;
- work with names;
- add names to dependent type theory.

Further Work

- Better freefrom types.
- Decidability questions.
- Practicable?
- Categorical structure
- Other choices primitives (e.g. $\Pi^* +$ Swapping $+ \mathrm{new}$)
- ...