Explicit Substitutions for Contextual Type Theory

Andreas Abel[1] Brigitte Pientka[2]

[1]Department of Computer Science
Ludwig-Maximilians-University Munich, Germany

[2]School of Computer Science
McGill University, Montreal, Canada

Logical Frameworks and Metalanguages: Theory and Practice 2010
Satellite Workshop at FLoC 2010, Edinburgh, UK
15 July 2010

Outline

Theory of Metavariables

- Metavariables are part of every implementation of type theory.
- Have been a “neglected child” by theoreticians.
- One remedy: Nanevski, Pfenning, and Pientka’s *Contextual Modal Type Theory*
- This work: explicit substitutions. Why?
  1. Efficient implementation.
  2. Initial model.
  3. Basis for formalization.

Metavariables as Contextual Objects

- Agda, Beluga, Coq, Delphin, Twelf use meta-variables in context.

```
map : {A B : Set}{n : N}->(A -> B) -> Vec A n -> Vec B n

sq : {n : Nat} -> Vec Nat n -> Vec (Vec Nat n) n
sq v = map (\ k -> map (\ l -> k * l) v) v

sq {n} v = map ?A ?B ?n (\ k -> map ?A' ?B' ?n' ...
```

- Type reconstruction instantiates meta-variables.

$$\begin{array}{lcl} n : \mathsf{Nat}, v : \mathsf{Vec\,Nat}\,n & \vdash & ?n : \mathsf{Nat} \\ n : \mathsf{Nat}, v : \mathsf{Vec\,Nat}\,n, k : \mathsf{Nat} & \vdash & ?n' : \mathsf{Nat} \end{array}$$

- Instantiation $?n := k$ would violate scope, $?n := \mathsf{suc}(v)$ typing.
- Store metavariables with their typing in meta-context:

$$?n : (n{:}\mathsf{Nat}, v{:}\ldots \;\triangleright\; \mathsf{Nat}),\; ?n' : (n{:}\mathsf{Nat}, v{:}\ldots, k{:}\mathsf{Nat} \;\triangleright\; \mathsf{Nat})$$

Beluga: HOAS and Recursion

- Higher-order abstract syntax (HOAS) in the logical framework (LF)

$$\begin{array}{lcl} \mathsf{abs} & : & (\mathsf{tm} \to \mathsf{tm}) \to \mathsf{tm} \\ \mathsf{app} & : & \mathsf{tm} \to (\mathsf{tm} \to \mathsf{tm}) \\ \mathsf{two} & = & \mathsf{abs}\,(\lambda f.\,\mathsf{abs}\,(\lambda x.\,\mathsf{app}\,f\,(\mathsf{app}\,f\,x))) \end{array}$$

- Embedded into a programming language as contextual type $\Gamma \triangleright \mathsf{tm}$.

$$\begin{array}{lcl} \multicolumn{3}{l}{\mathsf{reduce} : (\Gamma \triangleright \mathsf{tm}) \to (\Gamma \triangleright \mathsf{tm})} \\ \mathsf{reduce}\ (\Gamma \triangleright \mathsf{app}\,(\mathsf{abs}\,T)\,S) & = & \Gamma \triangleright T\ S \\ \mathsf{reduce}\ (\Gamma \triangleright \mathsf{abs}\ T) & = & \Gamma \triangleright \mathsf{abs}\ T' \\ \multicolumn{3}{l}{\quad \mathsf{where}\ \Gamma, x{:}\mathsf{tm} \triangleright T'\,x = \mathsf{reduce}\ (\Gamma, x{:}\mathsf{tm} \triangleright T\ x)} \\ \mathsf{reduce}\ (\Gamma \triangleright \mathsf{app}\ R\ S) & = & \Gamma \triangleright \mathsf{app}\ R'\ S \\ \multicolumn{3}{l}{\quad \mathsf{where}\ \Gamma \triangleright R' = \mathsf{reduce}\ (\Gamma \triangleright R)} \\ \mathsf{reduce}\ (\Gamma \triangleright R) & = & \Gamma \triangleright R \end{array}$$

- Contextual/Meta-variables $T : \Gamma \triangleright \mathsf{tm} \to \mathsf{tm}$, $R, R', S : \Gamma \triangleright \mathsf{tm}$.

Beluga Typing Judgements

- Contexts:

Γ	LF-variables	$x : A$
Δ	Meta-variables	$X : \Gamma \triangleright A$
Φ	Program variables	$x : \tau$

- Typing judgements:

$\Delta; \Gamma \vdash_{\mathrm{LF}} t : A$	LF objects
$\Delta \mid \Phi \vdash_{\mathrm{BEL}} e : \tau$	Beluga programs

Beluga Typing Rules

- Introducing contextual objects:

$$\frac{}{\Delta, X{:}(\Gamma \triangleright A); \Gamma \vdash_{\mathrm{LF}} X : A} \qquad \frac{\Delta; \Gamma \vdash_{\mathrm{LF}} t : A}{\Delta \mid \Phi \vdash_{\mathrm{BEL}} \Gamma \triangleright t : \Gamma \triangleright A}$$

- Metavariable abstraction in programs:

$$\frac{\Delta, X{:}(\Gamma \triangleright A) \mid \Phi \vdash_{\mathrm{BEL}} e : \tau}{\Delta \mid \Phi \vdash_{\mathrm{BEL}} \lambda X.e : \Pi^{\Box} X{:}(\Gamma \triangleright A).\,\tau}$$

- Application introduces a meta-substitution:

$$\frac{\Delta \mid \Phi \vdash_{\mathrm{BEL}} f : \Pi^{\Box} X{:}(\Gamma \triangleright A).\,\tau \qquad \Delta; \Gamma \vdash_{\mathrm{LF}} t : A}{\Delta \mid \Phi \vdash_{\mathrm{BEL}} f\ t : [\![t/X]\!]\tau}$$

Contribution of This Work

- Only consider LF-level $\Delta; \Gamma \vdash_{\mathrm{LF}} M : A$.
- Give typing and equality rules in term of explicit substitutions $[\sigma]M$ and explicit meta-substitutions $[\![\theta]\!]M$.
- Clarify interaction of $[\sigma]M$ and $[\![\theta]\!]M$.
- Formulate lazy weak head evaluation strategy which propagates both substitutions on demand.
- Formulate algorithms for type and equality checking.

Typing with Explicit Substitutions

- Variables are represented as de Bruijn indices $x_1, x_2, \ldots$.

$$\frac{|\Gamma'| = n}{\Delta;\Gamma, A, \Gamma' \vdash x_{n+1} : [\uparrow^{n+1}]A} \qquad \frac{\Delta;\Gamma, A \vdash M : B}{\Delta;\Gamma \vdash \lambda M : \Pi A.B}$$

$$\frac{\Delta;\Gamma \vdash M : \Pi A.B \qquad \Delta;\Gamma \vdash N : A}{\Delta;\Gamma \vdash M\,N : [\uparrow^0, N]B} \qquad \frac{\Delta;\Gamma \vdash \sigma : \Gamma' \qquad \Delta;\Gamma' \vdash M : A}{\Delta;\Gamma \vdash [\sigma]M : [\sigma]A}$$

- Explicit substitution $\Delta;\Gamma \vdash \sigma : \Gamma'$ maps a valuation of Γ to one of Γ'.

$$\frac{|\Gamma'| = n}{\Delta;\Gamma, \Gamma' \vdash \uparrow^n : \Gamma} \qquad \frac{\Delta;\Gamma_1 \vdash \sigma : \Gamma_2 \qquad \Delta;\Gamma_2 \vdash \sigma' : \Gamma_3}{\Delta;\Gamma_1 \vdash [\sigma]\sigma' : \Gamma_3}$$

$$\frac{\Delta;\Gamma \vdash \sigma : \Gamma' \qquad \Delta;\Gamma' \vdash A \qquad \Delta;\Gamma \vdash M : [\sigma]A}{\Delta;\Gamma \vdash (\sigma, M) : (\Gamma', A)}$$

Rules for the Meta Level

- A shift of perspective...

$$\Delta; \Gamma \vdash M : A \quad \longrightarrow \quad \Delta \vdash M : (\Gamma \triangleright A)$$

- Deriving the rules for the meta level.

$$\frac{|\Delta'| = n}{\Delta, \Gamma \triangleright A, \Delta' \vdash X_{n+1} : [\![\Uparrow^{n+1}]\!](\Gamma \triangleright A)} \qquad \frac{\Delta \vdash \theta : \Delta' \qquad \Delta' \vdash M : \Gamma \triangleright A}{\Delta \vdash [\![\theta]\!]M : [\![\theta]\!](\Gamma \triangleright A)}$$

- Meta-substitution $\Delta \vdash \theta : \Delta'$.

$$\frac{|\Delta'| = n}{\Delta, \Delta' \vdash \Uparrow^{n} : \Delta} \qquad \frac{\Delta_1 \vdash \theta : \Delta_2 \qquad \Delta_2 \vdash \theta' : \Delta_3}{\Delta_1 \vdash [\![\theta]\!]\theta' : \Delta_3}$$

$$\frac{\Delta \vdash \theta : \Delta' \qquad \Delta' \vdash \Gamma \triangleright A \qquad \Delta \vdash M : [\![\theta]\!](\Gamma \triangleright A)}{\Delta \vdash (\theta, M) : \Delta', \Gamma \triangleright A}$$

Interaction of Substitutions

- Viewing $\Delta;\Gamma \vdash \sigma : \Gamma'$ as $\Delta \vdash \sigma : (\Gamma \triangleright \Gamma')$:

$$\frac{\Delta \vdash \theta : \Delta' \qquad \Delta' \vdash \sigma : (\Gamma \triangleright \Gamma')}{\Delta \vdash [\![\theta]\!]\sigma : [\![\theta]\!](\Gamma \triangleright \Gamma')}$$

- However, $[\sigma]\theta$ is not definable.
- Reductions:

$$\begin{array}{lll} [\![\theta]\!][\sigma]M & \longrightarrow & [[\![\theta]\!]\sigma][\![\theta]\!]M \\ [\sigma][\![\theta]\!]M & \not\longrightarrow & \ldots \qquad (\text{unless } [\![\theta]\!]M \longrightarrow\ ) \end{array}$$

- Hence, our closures are of form $[\sigma][\![\theta]\!]M$.

Computing Substitutions

- Variable lookup.

$$
\begin{array}{lcl}
[\sigma, M]x_{m+1} & \longrightarrow & [\sigma]x_m \\
[\sigma, M]x_1 & \longrightarrow & M \\
[\uparrow^n]x_m & \longrightarrow & x_{n+m}
\end{array}
$$

- Propagation into term.

$$
\begin{array}{lcl}
[\sigma]X_m & \not\longrightarrow & \\
[\sigma](M\,N) & \longrightarrow & ([\sigma]M)\,([\sigma]N) \\
[\sigma]\lambda M & \longrightarrow & \lambda\,[[\uparrow^1]\sigma, x_1]M
\end{array}
$$

- Propagation into substitution.

$$
\begin{array}{lcl}
[\sigma](\sigma', M) & \longrightarrow & ([\sigma]\sigma', [\sigma]M) \\
[\sigma][\sigma_1]\sigma_2 & \longrightarrow & [[\sigma]\sigma_1]\sigma_2
\end{array}
$$

etc.

Computing Meta-Substitutions

- Meta-variable lookup.

$$\begin{array}{lcl} [\![\theta, M]\!] X_{m+1} & \longrightarrow & [\![\theta]\!] X_m \\ [\![\theta, M]\!] X_1 & \longrightarrow & M \\ [\![\Uparrow^n]\!] X_m & \longrightarrow & X_{n+m} \end{array}$$

- Propagation into term.

$$\begin{array}{lcl} [\![\theta]\!] x_m & \longrightarrow & x_m \\ [\![\theta]\!] (M\, N) & \longrightarrow & ([\![\theta]\!] M)\, ([\![\theta]\!] N) \\ [\![\theta]\!] \lambda M & \longrightarrow & \lambda\, [\![\theta]\!] M \end{array}$$

- Propagation into substitution.

$$\begin{array}{lcl} [\![\theta]\!] (\sigma, M) & \longrightarrow & ([\![\theta]\!] \sigma, [\![\theta]\!] M) \\ [\![\theta]\!] [\sigma_1] \sigma_2 & \longrightarrow & [[\![\theta]\!] \sigma_1] [\![\theta]\!] \sigma_2 \\ [\![\theta]\!] \uparrow^n & \longrightarrow & \uparrow^n \qquad \ldots \end{array}$$

Weak Head Reduction

- Do not propagate substitutions under λ, wait for application.

$$([\sigma][\![\theta]\!]\lambda M)\, N \longrightarrow [\sigma, N][\![\theta]\!]M$$

- Choice: combine substitutions eagerly $\Longrightarrow$ environments.

CBN-Whnfs	W	$::=$	$[\rho][\![\eta]\!]\lambda M \mid x_n\, \vec{L} \mid ([\rho]X_n)\, \vec{L}$
Closures	L	$::=$	$[\rho][\![\eta]\!]M \mid x_n$
Environments	ρ	$::=$	$\uparrow^n \mid (\rho, L) \mid [\uparrow^n]\rho$
Meta-environments	η	$::=$	$\Uparrow^n \mid (\eta, M)$

Algorithmic Equality

- Syntax-directed relation $W \overset{\mathsf{w}}{\sim} W'$ to check $\beta\eta$-equality.
- Follows idea of Coquand (1991) to η-expand lazily.

$$\frac{(\mathsf{whnf}\ [\rho'][\![\eta]\!]M) \overset{\mathsf{w}}{\sim} x_{n+1}\,\vec{L'}\,x_1}{[\rho][\![\eta]\!]\lambda M \overset{\mathsf{w}}{\sim} x_n\,\vec{L}}\ \rho' = ([\uparrow^1]\rho, x_1),\ \ L'_i = [\uparrow^1]L_i$$

- Soundness *(if algorithm says yes, then objects are really $\beta\eta$-equal)* is easy induction.
- Completeness *(if two $\beta\eta$-equal objects are given to algorithm, it says yes)* by PER model (future work, following Abel Coquand 2007).

Type Checking Normal Forms, Bidirectional

- Inference $\Delta; \Gamma \vdash M \rightrightarrows L$.

$$\frac{}{\Delta; \Gamma, L \vdash x_1 \rightrightarrows [\uparrow^1]L} \qquad \frac{\Delta, \Gamma' \triangleright A; \Gamma \vdash \sigma \leftleftarrows [\![\Uparrow^1]\!]\Gamma'}{\Delta, \Gamma' \triangleright A; \Gamma \vdash [\sigma]X_1 \rightrightarrows [\sigma][\![\Uparrow^1]\!]A}$$

$$\frac{\Delta; \Gamma \vdash M \rightrightarrows L \qquad \text{whnf } L = [\rho][\![\theta]\!]\Pi A.B \qquad \Delta; \Gamma \vdash N \leftleftarrows [\rho][\![\theta]\!]A}{\Delta; \Gamma \vdash M\, N \rightrightarrows [\rho, N][\![\theta]\!]B}$$

- Checking $\Delta; \Gamma \vdash M \leftleftarrows L$.

$$\frac{\text{whnf } L = [\rho][\![\eta]\!](\Pi\, A.\, B) \quad \Delta; \Gamma, [\rho][\![\eta]\!]A \vdash M \leftleftarrows [\uparrow^1\rho, x_1][\![\eta]\!]B}{\Delta; \Gamma \vdash \lambda M \leftleftarrows L}$$

$$\frac{\Delta; \Gamma \vdash M \rightrightarrows L \quad \text{whnf } L \overset{\text{w}}{\sim} \text{whnf } L'}{\Delta; \Gamma \vdash M \leftleftarrows L'}$$

Conclusions

- Basis for this work:
  1. Nanevski, Pfenning, Pientka (2008): *Contextual Modal Type Theory* Merging declarative and algorithmic presentation by hereditary substitutions.
  2. Abel, Coquand (2007): *Untyped Algorithmic Equality for the Martin-Löf's Logical Framework with Surjective Pairing* Declarative typing and equality separate from algorithms, connected by a PER model.
  3. Explicit substitutions, e.g., *Abadi, Cardelli, Curien, Levy (1991)* or *Dowek, Hardin, Kirchner (2000)*.
- Future work:
  1. Finish normalization proof.
  2. Extend to full Beluga.
  3. Investigate most efficient normalization strategies.