Multi-level Contextual Type Theory

Mathieu Boespflug Brigitte Pientka

McGill University

26 August 2011

$$\forall x.\,(A[x]) \wedge B$$

$$\forall x.\, (A[x]) \land B$$

$$\Downarrow$$

$$\forall x.\, (A[x] \land B)$$

$$\forall A. \forall B.\ \forall x.\ (A[x]) \land B$$

$$\Downarrow$$

$$\forall A. \forall B.\ \forall x.\ (A[x] \land B)$$

$$\forall A. \forall B.\ \forall x.\ (A[x]) \wedge B$$

$$\Downarrow \qquad \longleftarrow \text{meta-level implication}$$

$$\forall A. \forall B.\ \forall x.\ (A[x] \wedge B)$$

Object language: first order predicate logic.
Meta- language: LF

Embedding FOL in LF

```
o : type .
i : type .

all : (i → o) → o.
and : o → o → o.

nd : o → type .
allI : ({x:i} nd (A x)) → nd (all (λx. A x)).
andI : nd A → nd B → nd (and A B).
```

Object language: first order predicate logic.
Meta- language: LF

Embedding FOL in LF

```
o : type .
i : type .

all : (i → o) → o.
and : o → o → o.

nd : o → type .
allI : {A:i→o} ({x:i} nd (A x)) → nd (all (λx. A x)).
andI : {A:o} {B:o} nd A → nd B → nd (and A B).
```

Object language: first order predicate logic.
Meta- language: LF

$$\frac{\texttt{andI (allI (}\lambda\texttt{x. D1 x)) D2}}{\texttt{allI (}\lambda\texttt{x. andI (D1 x) D2)}}$$

Object language: first order predicate logic.
Meta- language: ??

$$\frac{\texttt{andI \_ \_ (allI \_ (}\lambda\texttt{x. D1 x)) D2}}{\texttt{allI \_ (}\lambda\texttt{x. andI \_ \_ (D1 x) D2)}}$$

Object language: first order predicate logic.
Meta- language: ??

$$\frac{\texttt{andI (all (}\lambda\texttt{x. A x)) B (allI (}\lambda\texttt{x. A x) (}\lambda\texttt{x. D1 x)) D2}}{\texttt{allI (}\lambda\texttt{x. and (A x) B) (}\lambda\texttt{x. andI (A x) B (D1 x) D2)}}$$

Object language: first order predicate logic.
Meta- language: ??

```
rec  proof :
        (nd (and (all (λx. A x)) B))[]
     → (nd (all (λx. and (A x) B)))[] =
```

Object language: first order predicate logic.
Meta- language: Beluga

```
rec  proof : {A::(i→o)[]} {B::o[]}
        (nd (and (all (λx. A x)) B))[]
     → (nd (all (λx. and (A x) B)))[] =
```

Object language: first order predicate logic.
Meta- language: Beluga

```
rec proof : {A::(i→o)[]} {B::o[]}
      (nd (and (all (λx. A x)) B))[]
   → (nd (all (λx. and (A x) B)))[] =
 λ A ⇒ λ B ⇒  fn d ⇒ case d of
  [] andI (allI D1) D2 ⇒
     [] allI (λx. andI (D1 x) D2);
```

Object language: first order predicate logic.
Meta- language: Beluga

```
rec proof : {A::(i→o)[]} {B::o[]}
      (nd (and (all (λx. A x)) B))[]
   → (nd (all (λx. and (A x) B)))[] =
 λ A ⇒ λ B ⇒ fn d ⇒ case d of
  [] andI _ _ (allI _ D1) D2 ⇒
     [] allI _ (λx. andI _ _ (D1 x) D2);
```

Object language: first order predicate logic.
Meta- language: Beluga

Inventory

	stand for	**depend on**
variables	terms	
meta-variables	meta-terms	variables
meta^2-variables	meta^2-terms	meta-variables, variables

Inventory

	stand for	**depend on**
variables	terms	
meta-variables	meta-terms	variables
meta^2-variables	meta^2-terms	meta-variables, variables
$\vdots$	$\vdots$	$\vdots$
meta^n-variables	meta-^{n}-terms	meta-$^{n-1}$-variables, …

$$\lambda x.\, \lambda y.\, \mathrm{andI}\ (D1\ x)\ D2$$

$$\lambda x.\, \lambda y.\, \mathrm{andI}\ (D1\ x)\ D2 : \Pi x{:}\iota.\, \Pi y{:}\iota.\, o$$

What can $D1$ be instantiated with?

- an open term where x can appear free...
- ... of type $\Pi y{:}\iota.\, o$.

Contextual Modal Type Theory (Nanevski et al, 2008)

$$\frac{\Delta(X) = A[\Gamma]}{\Delta; \Psi \vdash X : A[\Gamma]}$$

- Well-typed meta-terms do not go wrong on instantiation.
- **Invariant 1:** Meta-variables always associated with "stuck" substitutions, with $\mathsf{dom}(\rho) = \mathsf{dom}(\Gamma)$.

Contextual Modal Type Theory (Nanevski et al, 2008)

$$\frac{\Delta(X) = A[\Gamma] \quad \Delta; \Psi \vdash \rho : \Gamma}{\Delta; \Psi \vdash X[\rho] : [\rho]A}$$

- Well-typed meta-terms do not go wrong on instantiation.
- **Invariant 1:** Meta-variables always associated with “stuck” substitutions, with $\mathsf{dom}(\rho) = \mathsf{dom}(\Gamma)$.

Contextual Modal Type Theory (Nanevski et al, 2008)

$$\frac{\Delta(X) = A[\Gamma] \quad \Delta; \Psi \vdash \rho : \Gamma}{\Delta; \Psi \vdash X[\rho] : [\rho]A}$$

$$\frac{\Psi(x) = A}{\Delta; \Psi \vdash x : A}$$

- Well-typed meta-terms do not go wrong on instantiation.
- **Invariant 1:** Meta-variables always associated with “stuck” substitutions, with $\mathsf{dom}(\rho) = \mathsf{dom}(\Gamma)$.

CMTT... to ω...

$$\frac{\Delta; \Psi, x{:}A \vdash M : B}{\Delta; \Psi \vdash \lambda x : A. M : \Pi x{:}A. B}$$

$$\frac{\Delta, X{:}A[\Phi]; \Psi \vdash M : B}{\Delta; \Psi \vdash \lambda X : A[\Phi]. M : \Pi X{:}A[\Phi]. B}$$

CMTT... to ω...

$$\left.\begin{array}{c}
\dfrac{\Delta_k; \Delta_{k-1}, x^0{:}A; \cdots \vdash M : B}{\Delta_k; \Delta_{k-1}; \cdots \vdash \lambda x^0 : A.M : \Pi x^0{:}A.B} \\[2em]
\dfrac{\Delta_k; \Delta_{k-1}; \cdots ; \Delta_1, x^1{:}A[\Phi]; \Delta_0 \vdash M : B}{\Delta_k; \Delta_{k-1}; \cdots ; \Delta_1; \Delta_0 \vdash \lambda x^1 : A[\Phi].M : \Pi x^1{:}A[\Phi].B} \\[2em]
\vdots \\[2em]
\dfrac{\Delta_k; \Delta_{k-1}; \cdots ; x^k{:}A[\Phi]; \cdots \vdash M : B}{\Delta_k; \Delta_{k-1}; \cdots \vdash \lambda x^k : A[\Phi].M : \Pi x^k{:}A[\Phi].B}
\end{array}\right\}$$

... to MLCMTT

$$\frac{\Psi \mathbin{+\!\!+} x^n{:}A[\Phi^n] \vdash M : B}{\Psi \vdash \lambda x^n : A[\Phi^n].M : \Pi x^n{:}A[\Phi^n].B}$$

... to MLCMTT

$$\frac{\Psi \vdash R : \Pi x^n{:}A[\Phi^n].B \quad \Psi|_n +\!\!+\; \Phi^n \vdash N : A}{\Psi \vdash R\,(\hat{\Phi}^n.N) : [\hat{\Phi}^n.N/x^n]B}$$

- From a schema of contexts lists to a context schema.
- From a schema of rules to a single rule.

MLCMTT (typing rules for terms)

$$\frac{\Psi(x^n) = A[\Phi^n] \quad \Psi \vdash \sigma \Leftarrow \Phi^n}{\Psi \vdash x^n[\sigma] \Rightarrow [\sigma]A}$$

$$\frac{\Psi \vdash R : \Pi x^n{:}A[\Phi^n].B \quad \Psi|_n +\!\!+\, \Phi^n \vdash N : A}{\Psi \vdash R\,(\hat{\Phi}^n.N) : [\hat{\Phi}^n.N/x^n]B}$$

$$\frac{\Psi +\!\!+\, x^n{:}A[\Phi^n] \vdash M : B}{\Psi \vdash \lambda x^n : A[\Phi^n].M : \Pi x^n{:}A[\Phi^n].B}$$

- Single syntactic category of level-indexed variables.
- Can abstract at any level at any time.

Context insertion

$$\begin{aligned} & x^3{:}\iota, w^2{:}\iota, x^2{:}\iota \mathbin{+\!\!+} y^5{:}\iota, v^0{:}\iota, w^0{:}\iota \\ = \; & y^5{:}\iota, x^3{:}\iota, w^2{:}\iota, x^2{:}\iota, v^0{:}\iota, w^0{:}\iota \end{aligned}$$

- **Invariant 2:** context assumptions always in decreasing order of level.

Context insertion

$$
\begin{aligned}
& x^3{:}\iota, w^2{:}\iota, x^2{:}\iota \mathbin{+\!\!+} y^5{:}\iota, v^0{:}\iota, w^0{:}\iota \\
= \; & y^5{:}\iota, x^3{:}\iota, w^2{:}\iota, x^2{:}\iota, v^0{:}\iota, w^0{:}\iota
\end{aligned}
$$

- **Invariant 2:** context assumptions always in decreasing order of level.
- Chopping context at level n stays easy.
- Argument that meta^n-variables don't depend on meta^{n-m}-variables trivial.

Example revisited

$$\frac{\texttt{andI \_ \_ (allI \_ (}\lambda\texttt{x}^0\texttt{. (x}^0\texttt{.D1}^1\texttt{)x}^0\texttt{)) .D2}^1}{\texttt{allI \_ (}\lambda\texttt{x}^0\texttt{. andI \_ \_ ((x}^0\texttt{.D1}^1\texttt{) x}^0\texttt{) .D2}^1\texttt{)}}$$

where

$$\begin{array}{lcl} \texttt{M}^2\texttt{,N}^2\texttt{,L}^2 & : & o[A^1{:}o[],B^1{:}o[],D1^1{:}\mathrm{nd}\,A^1[x^0{:}\iota],D2^1{:}\mathrm{nd}\,B^1[]] \\ \texttt{P}^2\texttt{,Q}^2\texttt{,R}^2 & : & o[A^1{:}o[],B^1{:}o[],D1^1{:}\mathrm{nd}\,A^1[x^0{:}\iota],D2^1{:}\mathrm{nd}\,B^1[],x^0{:}\iota] \end{array}$$

Example revisited

$$\frac{\texttt{andI}\ .\texttt{M}^2\ .\texttt{N}^2\ (\texttt{allI}\ .\texttt{L}^2\ (\lambda \texttt{x}^0 .\ (\texttt{x}^0\ .\texttt{D1}^1)\texttt{x}^0))\ \texttt{D2}^1}{\texttt{allI}\ .\texttt{P}^2\ (\lambda \texttt{x}^0 .\ \texttt{andI}\ .\texttt{Q}^2\ .\texttt{R}^2\ ((\texttt{x}^0\ .\texttt{D1}^1)\ \texttt{x}^0)\ \texttt{D2}^1)}$$

where

$$\begin{array}{lcl} \texttt{M}^2, \texttt{N}^2, \texttt{L}^2 & : & o[A^1{:}o[], B^1{:}o[], D1^1{:}\mathrm{nd}\, A^1[x^0{:}\iota], D2^1{:}\mathrm{nd}\, B^1[]] \\ \texttt{P}^2, \texttt{Q}^2, \texttt{R}^2 & : & o[A^1{:}o[], B^1{:}o[], D1^1{:}\mathrm{nd}\, A^1[x^0{:}\iota], D2^1{:}\mathrm{nd}\, B^1[], x^0{:}\iota] \end{array}$$

Results

- Usual structural properties.
- Substitution property.
  $\longrightarrow$ *“substitution does not go wrong.”*
- Termination of heredetary substitutions.
- Decidability of type checking.

Related Work

- “A logical basis for explicit substitutions” (Pfenning, 2007)
  $\longrightarrow$ *Results only worked out for k=2, only simple types.*
- Multi-level logics of contexts (Giunchiglia, Serafini, 1993-1994)
- Multi-level meta-variables (Sato et al, 2003)
  $\longrightarrow$ *Textual substitution, no confluence*
- Lambda Context Calculus (Gabbay and Lengrand, 2007)
  $\longrightarrow$ *Nominal approach to meta-variabes*
- “Open proofs and open types” (Geuvers and Gogjov, 2002)
  $\longrightarrow$ *Reduction and instantiation do not commute.*
- Staged computation, code generation (Davies and Pfenning, 2001) (Glück and Jorgensen, 1995-1996) (Yuse and Yigarashi, 2006).

Conclusion

What we have:

- Uniform language for representing normal meta^n-terms.

Conclusion

What we have:

- Uniform language for representing normal meta^n-terms.

Where we are going:

- Implement it.
- Reasoning about programs conveniently and efficiently.
- Reasoning about Beluga in Beluga.