A Well-typed Interpreter for a Dependently Typed Language

Nils Anders Danielsson

Chalmers

February 21, 2007

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

Introduction

- Well-typed representation of dependently typed lambda calculus (no raw terms).
- Interpreter (= normalisation proof).
- Implemented in AgdaLight.

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

Well-typed language

$$\begin{aligned}
&\textbf{data } \mathit{Ty} : * \textbf{ where}\\
&\quad \mathit{Nat}' \;: \mathit{Ty}\\
&\quad \mathit{Bool}' : \mathit{Ty}
\end{aligned}$$

$$\begin{aligned}
&\textbf{data } \mathit{Op} : \mathit{Ty} \to * \textbf{ where}\\
&\quad \mathit{plus} \;: \mathit{Op}\ \mathit{Nat}'\\
&\quad \mathit{and} \;: \mathit{Op}\ \mathit{Bool}'
\end{aligned}$$

$$\begin{aligned}
&\textbf{data } \mathit{Term} : \mathit{Ty} \to * \textbf{ where}\\
&\quad \mathit{nat} \;\,: \mathit{Nat} \;\to \mathit{Term}\ \mathit{Nat}'\\
&\quad \mathit{bool} : \mathit{Bool} \to \mathit{Term}\ \mathit{Bool}'\\
&\quad \mathit{op} \;\;\,: \;\{\sigma : \mathit{Ty}\} \to \mathit{Op}\ \sigma\\
&\qquad\quad \to \mathit{Term}\ \sigma \to \mathit{Term}\ \sigma \to \mathit{Term}\ \sigma
\end{aligned}$$

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

Interpreter which cannot get stuck

$$\begin{aligned}
&\mathit{Sem} : \mathit{Ty} \to *\\
&\mathit{Sem}\ \mathit{Nat}' \;= \mathit{Nat}\\
&\mathit{Sem}\ \mathit{Bool}' = \mathit{Bool}
\end{aligned}$$

$$\begin{aligned}
&[\![\cdot]\!] : \mathit{Term}\ \sigma \to \mathit{Sem}\ \sigma\\
&[\![\mathit{nat}\ n]\!] \qquad = n\\
&[\![\mathit{bool}\ b]\!] \qquad = b\\
&[\![\mathit{op}\ o\ t_1\ t_2]\!] = \mathit{fun}\ o\ [\![t_1]\!]\ [\![t_2]\!]\\
&\quad \textbf{where}\\
&\quad \mathit{fun} : \mathit{Op}\ \sigma \to (\mathit{Sem}\ \sigma \to \mathit{Sem}\ \sigma \to \mathit{Sem}\ \sigma)\\
&\quad \mathit{fun}\ \mathit{plus} = (+)\\
&\quad \mathit{fun}\ \mathit{and} = (\wedge)
\end{aligned}$$

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

Object language

- Variant of Martin-Löf's logical framework.
- Variables (de Bruijn indices).
- Explicit substitutions.
- All definitions are mutually recursive.

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

Contexts

$$\begin{aligned}
&\textbf{data } \mathit{Ctxt} : * \textbf{ where}\\
&\quad \varepsilon \;\;\; : \mathit{Ctxt}\\
&\quad (\triangleright) : (\Gamma : \mathit{Ctxt}) \to \mathit{Ty}\ \Gamma \to \mathit{Ctxt}
\end{aligned}$$

$$\frac{}{\varepsilon\ \mathit{context}} \qquad \frac{\Gamma\ \mathit{context} \qquad \Gamma \vdash \tau\ \mathit{type}}{\Gamma \triangleright \tau\ \mathit{context}}$$

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

Types

data $Ty : Ctxt \to * $ **where**
$\star : Ty\ \Gamma$
$El : \Gamma \vdash \star \to Ty\ \Gamma$
$\Pi : (\tau : Ty\ \Gamma) \to Ty\ (\Gamma \rhd \tau) \to Ty\ \Gamma$

$$\frac{}{\Gamma \vdash \star\ type} \qquad \frac{\Gamma \vdash t : \star}{\Gamma \vdash El\ t\ type}$$

$$\frac{\Gamma \vdash \tau_1\ type \qquad \Gamma \rhd \tau_1 \vdash \tau_2\ type}{\Gamma \vdash \Pi\ \tau_1\ \tau_2\ type}$$

Types

data $Ty : Ctxt \to * $ **where**
$\star : Ty\ \Gamma$
$El : \Gamma \vdash \star \to Ty\ \Gamma$
$\Pi : (\tau : Ty\ \Gamma) \to Ty\ (\Gamma \rhd \tau) \to Ty\ \Gamma$

$(/) : Ty\ \Gamma \to \Gamma \Rightarrow \Delta \to Ty\ \Delta$
$\star \quad / \rho = \star$
$El\ t \quad / \rho = El\ (t \mathbin{/_{\vdash}} \rho)$
$\Pi\ \tau_1\ \tau_2 / \rho = \Pi\ (\tau_1 / \rho)\ (\tau_2\ /\ \rho \uparrow \tau_1)$

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

Terms

$$\frac{\Gamma \vdash v : \tau}{\Gamma \vdash var\ v : \tau} \qquad \frac{\Gamma \rhd \tau_1 \vdash t : \tau_2}{\Gamma \vdash \lambda\ t : \Pi\ \tau_1\ \tau_2}$$

$$\frac{\Gamma \vdash t_1 : \Pi\ \tau_1\ \tau_2 \qquad \Gamma \vdash t_2 : \tau_1}{\Gamma \vdash t_1 @ t_2 : \tau_2\ /\ sub\ t_2}$$

$$\frac{\Gamma \vdash t : \tau_1 \qquad eq : \tau_1 =_{\star} \tau_2}{\Gamma \vdash t ::_{\vdash} eq : \tau_2}$$

$$\frac{\Gamma \vdash t : \tau \qquad \rho : \Gamma \Rightarrow \Delta}{\Delta \vdash t \mathbin{/_{\vdash}} \rho : \tau\ /\ \rho}$$

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

Variables

$(\ni)$: $(\Gamma : Ctxt) \to Ty\ \Gamma \to *$
vz : $\Gamma \rhd \sigma \ni \sigma / wk\ \sigma$
Variable zero.
σ is in Γ, not $\Gamma \rhd \sigma$, so it needs to be weakened.
$vs\ v$ The variable after v.
$(::_{\ni})$ Conversion rule.

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

Substitutions

$(\Rightarrow)$: $Ctxt \to Ctxt \to *$
sub Single term substitutions ($[vz := t]$).
wk Weakenings.
$(\uparrow)$ Lifting.
id Identity.
$(\odot)$ Composition.

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

Equality

- Congruence.
- β- and η-rules.
- Evaluation rules for $(/_{\vdash})$.
- Casts $(::)$ can be removed.

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

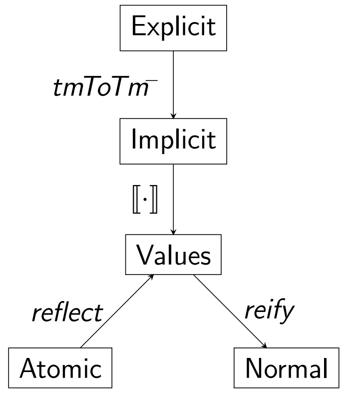
Interpreter

A choice

- Use general recursion.
  - Relatively easy.
  - Partially correct.
  - No immediate guarantees about termination.
- Use structural recursion.
  - Harder.
  - Totally correct.

Here: structural recursion using normalisation by evaluation.

Overview



Types ensure soundness

- $Tm^{-}, Val, Atom, NF : \Gamma \vdash \tau \;\rightarrow\; *$
- $nf : NF\ t$ means that nf is a normal form $\beta\eta$-equivalent to t.

$$
\begin{array}{lll}
tmToTm^{-} & : (t : \Gamma \vdash \tau) & \rightarrow\; Tm^{-}\ t \\
[\![\cdot]\!] & : Tm^{-}\ t \rightarrow Env\ \rho & \rightarrow\; Val\ (t \mathbin{/\!\!\vdash} \rho) \\
reflect & : Atom\ t & \rightarrow\; Val\ t \\
reify & : Val\ t & \rightarrow\; NF\ t
\end{array}
$$

Interesting cases

- Π types are represented using functions that perform application.

$$
\begin{array}{l}
\textbf{data}\ Val : \Gamma \vdash \tau \rightarrow * \ \textbf{where} \\
\quad \Pi_{Val} \ :\ \{t_1 : \Gamma \vdash \Pi\ \tau_1\ \tau_2\} \\
\qquad \rightarrow (f \ :\ (\Gamma' : Ctxt^{+}\ \Gamma) \\
\qquad\qquad \rightarrow \{t_2 : \Gamma \mathbin{+\!\!+} \Gamma' \vdash \tau_1 \ /\ wk^{*}\ \Gamma'\} \\
\qquad\qquad \rightarrow (v_2 : Val\ t_2) \\
\qquad\qquad \rightarrow Val\ ((t_1 \mathbin{/\!\!\vdash} wk^{*}\ \Gamma') @ t_2)) \\
\qquad \rightarrow Val\ t_1
\end{array}
$$

$$
\begin{array}{l}
[\![\cdot]\!] : Tm^{-}\ t \rightarrow Env\ \rho \rightarrow Val\ (t \mathbin{/\!\!\vdash} \rho) \\
[\![\lambda^{-}\ t_1^{-}]\!]\,\gamma = \Pi_{Val}\ (\backslash\Delta'\ v_2 \rightarrow \\
\quad [\![t_1^{-}]\!]\,(wk^{\star}_{Env}\ \gamma\ \Delta' \blacktriangleright_{Env} (v_2 \ ::_{Val}\ \ldots))\ ::_{Val} \ldots\beta\ldots)
\end{array}
$$

Interesting cases

$$
\begin{array}{l}
reify : (\tau : Ty\ \Gamma) \rightarrow \{t : \Gamma \vdash \tau\} \rightarrow Val\ t \rightarrow NF\ t \\
reify\ (\Pi\ \tau_1\ \tau_2)\ (\Pi_{Val}\ f) = \\
\quad \lambda_{NF}\ (reify\ (\tau_2\ /\ \_\ /\ \_) \\
\qquad\qquad (f\ (\varepsilon^{+} \triangleright^{+} \tau_1) \\
\qquad\qquad\quad (reflect\ (\tau_1\ /\ \_)\ (var_{At}\ vz)\ ::_{Val}\ \ldots))) \\
\quad ::_{NF} \ldots\eta\ldots
\end{array}
$$

$$
\begin{array}{l}
reflect : (\tau : Ty\ \Gamma) \rightarrow \{t : \Gamma \vdash \tau\} \rightarrow Atom\ t \rightarrow Val\ t \\
reflect\ (\Pi\ \tau_1\ \tau_2)\ at_1 = \Pi_{Val}\ (\backslash\Gamma'\ v_2 \rightarrow \\
\quad reflect\ (\tau_2\ /\ \_\ /\ \_)\ (wk^{*}_{At}\ at_1\ \Gamma'\ @_{At}\ reify\ (\tau_1\ /\ \_)\ v_2))
\end{array}
$$

Type checker

- What about parsing?
- Using the normaliser it is easy to define a type checker.

Discussion

- Looks horribly complicated?
  Remember that the types used are very precise:
  - Hence guiding the programmer.
  - And limiting the amount of possible errors.
- But it's not something you hack up in an afternoon.

Finally. . .

Simple example
Object language
Contexts
Types
Terms
Variables
Substitutions
Equality
Interpreter
Soundness
Interesting cases
Type checker
Discussion

- For more details, see paper:
  A Formalisation of a Dependently Typed Language as an Inductive-Recursive Family.
- Source code available to play with.

Appendix

Terms

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

$$
\begin{array}{lllll}
\multicolumn{5}{l}{\textbf{data}\ (\vdash) : (\Gamma : \mathit{Ctxt}) \to \mathit{Ty}\ \Gamma \to * \ \textbf{where}} \\
\quad \mathit{var} & : \Gamma \ni \tau & & \to & \Gamma \vdash \tau \\
\quad \lambda & : \Gamma \triangleright \tau_1 \vdash \tau_2 & & \to & \Gamma \vdash \Pi\ \tau_1\ \tau_2 \\
\quad (@) & : \Gamma \vdash \Pi\ \tau_1\ \tau_2 & \to (t_2 : \Gamma \vdash \tau_1) & \to & \Gamma \vdash \tau_2 \,/\, \mathit{sub}\ t_2 \\
\quad (::_{\vdash}) & : \Gamma \vdash \tau_1 & \to \tau_1 =_{\star} \tau_2 & \to & \Gamma \vdash \tau_2 \\
\quad (/_{\vdash}) & : \Gamma \vdash \tau & \to (\rho : \Gamma \Rightarrow \Delta) & \to & \Delta \vdash \tau \,/\, \rho
\end{array}
$$

Variables

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

$$
\begin{array}{ll}
\multicolumn{2}{l}{\textbf{data}\ (\ni) : (\Gamma : \mathit{Ctxt}) \to \mathit{Ty}\ \Gamma \to * \ \textbf{where}} \\
\quad \mathit{vz} & : \qquad\qquad \Gamma \triangleright \sigma \ni \sigma \,/\, \mathit{wk}\ \sigma \\
\quad \mathit{vs} & : \Gamma \ni \tau \to \Gamma \triangleright \sigma \ni \tau \,/\, \mathit{wk}\ \sigma \\
\quad (::_{\ni}) & : \Gamma \ni \tau_1 \to \tau_1 =_{\star} \tau_2 \to \Gamma \ni \tau_2
\end{array}
$$

Substitutions

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

$$
\begin{array}{ll}
\multicolumn{2}{l}{\textbf{data}\ (\Rightarrow) : \mathit{Ctxt} \to \mathit{Ctxt} \to * \ \textbf{where}} \\
\quad \mathit{sub} & : \Gamma \vdash \tau \to \Gamma \triangleright \tau \Rightarrow \Gamma \\
\quad \mathit{wk} & : (\sigma : \mathit{Ty}\ \Gamma) \to \Gamma \Rightarrow \Gamma \triangleright \sigma \\
\quad \mathit{id} & : \Gamma \Rightarrow \Gamma \\
\quad (\odot) & : \Gamma \Rightarrow \Delta \to \Delta \Rightarrow X \to \Gamma \Rightarrow X \\
\quad (\uparrow) & : \quad (\rho : \Gamma \Rightarrow \Delta) \to (\sigma : \mathit{Ty}\ \Gamma) \\
 & \quad \to \Gamma \triangleright \sigma \Rightarrow \Delta \triangleright (\sigma \,/\, \rho)
\end{array}
$$

Term equality

$$\begin{array}{l}
\textbf{data}\ (=_{\vdash}) : \Gamma_1 \vdash \tau_1 \to \Gamma_2 \vdash \tau_2 \to * \ \textbf{where} \\
\quad \text{-- Equivalence.} \\
\quad \mathit{refl}_{\vdash} \ : (t : \Gamma \vdash \tau) \to t =_{\vdash} t \\
\quad \mathit{sym}_{\vdash} \ : t_1 =_{\vdash} t_2 \to t_2 =_{\vdash} t_1 \\
\quad \mathit{trans}_{\vdash} : t_1 =_{\vdash} t_2 \to t_2 =_{\vdash} t_3 \to t_1 =_{\vdash} t_3 \\
\quad \text{-- Congruence.} \\
\quad \mathit{var}_{Cong} \ : v_1 =_{\ni} v_2 \to \mathit{var}\ v_1 =_{\vdash} \mathit{var}\ v_2 \\
\quad \lambda_{Cong} \ : t_1 =_{\vdash} t_2 \to \lambda\ t_1 =_{\vdash} \lambda\ t_2 \\
\quad (@_{Cong}) : t_1^1 =_{\vdash} t_1^2 \to t_2^1 =_{\vdash} t_2^2 \to t_1^1 @ t_2^1 =_{\vdash} t_1^2 @ t_2^2 \\
\quad (/_{\vdash Cong}) : t_1 =_{\vdash} t_2 \to \rho_1 =_{\Rightarrow} \rho_2 \to t_1 /_{\vdash} \rho_1 =_{\vdash} t_2 /_{\vdash} \rho_2 \\
\quad \ldots
\end{array}$$

Term equality

$$\begin{array}{l}
\textbf{data}\ (=_{\vdash}) : \Gamma_1 \vdash \tau_1 \to \Gamma_2 \vdash \tau_2 \to * \ \textbf{where} \\
\quad \text{-- Cast, beta and eta equality.} \\
\quad \mathit{castEq}_{\vdash} : (t ::_{\vdash} \mathit{eq}) =_{\vdash} t \\
\quad \beta \quad : (\lambda\ t_1) @ t_2 =_{\vdash} t_1 /_{\vdash} \mathit{sub}\ t_2 \\
\quad \eta \quad : \quad (t : \Gamma \vdash \Pi\ \tau_1\ \tau_2) \\
\qquad\quad \to \lambda\ ((t /_{\vdash} \mathit{wk}\ \tau_1) @ \mathit{var}\ \mathit{vz}) =_{\vdash} t \\
\quad \ldots
\end{array}$$

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

Term equality

$$\begin{array}{l}
\textbf{data}\ (=_{\vdash}) : \Gamma_1 \vdash \tau_1 \to \Gamma_2 \vdash \tau_2 \to * \ \textbf{where} \\
\quad \text{-- Substitution application axioms.} \\
\quad \lambda\ t \quad /_{\vdash}\ \rho \quad =_{\vdash} \lambda\ (t\ /_{\vdash}\ \rho \uparrow \tau_1) \\
\quad (t_1 @ t_2) \quad /_{\vdash}\ \rho \quad =_{\vdash} (t_1 /_{\vdash} \rho) @ (t_2 /_{\vdash} \rho) \\
\quad t \quad /_{\vdash}\ \mathit{id} \quad =_{\vdash} t \\
\quad t \quad /_{\vdash}\ (\rho_1 \odot \rho_2) =_{\vdash} t /_{\vdash} \rho_1 /_{\vdash} \rho_2 \\
\quad \mathit{var}\ v \quad /_{\vdash}\ \mathit{wk}\ \sigma \quad =_{\vdash} \mathit{var}\ (\mathit{vs}\ v) \\
\quad \mathit{var}\ \mathit{vz} \quad /_{\vdash}\ \mathit{sub}\ t \quad =_{\vdash} t \\
\quad \mathit{var}\ (\mathit{vs}\ v) /_{\vdash}\ \mathit{sub}\ t \quad =_{\vdash} \mathit{var}\ v \\
\quad \mathit{var}\ \mathit{vz} \quad /_{\vdash}\ (\rho \uparrow \sigma) \quad =_{\vdash} \mathit{var}\ \mathit{vz} \\
\quad \mathit{var}\ (\mathit{vs}\ v) /_{\vdash}\ (\rho \uparrow \sigma) \quad =_{\vdash} \mathit{var}\ v /_{\vdash} \rho /_{\vdash} \mathit{wk}\ (\sigma \mathbin{/} \rho)
\end{array}$$

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

Implicit substitutions

$$\begin{array}{l}
\textbf{data}\ \mathit{Tm}^{-} : \Gamma \vdash \tau \to * \ \textbf{where} \\
\quad \mathit{var}^{-} \ : (v : \Gamma \ni \tau) \to \mathit{Tm}^{-}\ (\mathit{var}\ v) \\
\quad \lambda^{-} \quad : \quad \{t : \Gamma \rhd \tau_1 \vdash \tau_2\} \\
\qquad\quad \to \mathit{Ty}^{-}\ \tau_1 \to \mathit{Tm}^{-}\ t \\
\qquad\quad \to \mathit{Tm}^{-}\ (\lambda\ t) \\
\quad (@^{-}) \ : \mathit{Tm}^{-}\ t_1 \to \mathit{Tm}^{-}\ t_2 \quad \to \mathit{Tm}^{-}\ (t_1 @ t_2) \\
\quad (::_{\vdash^{-}}) : \mathit{Tm}^{-}\ t_1 \to t_1 =_{\vdash} t_2 \to \mathit{Tm}^{-}\ t_2 \\
\\
\mathit{tm}^{-}\mathit{ToTm} \quad : \{t : \Gamma \vdash \tau\} \to \mathit{Tm}^{-}\ t \to \Gamma \vdash \tau \\
\mathit{tm}^{-}\mathit{ToTmEq} : (t^{-} : \mathit{Tm}^{-}\ t) \to \mathit{tm}^{-}\mathit{ToTm}\ t^{-} =_{\vdash} t \\
\mathit{tmToTm}^{-} \quad : (t : \Gamma \vdash \tau) \to \mathit{Tm}^{-}\ t
\end{array}$$

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

Normal forms

$$\begin{array}{l}
\textbf{data}\ \mathit{Atom} : \Gamma \vdash \tau \to * \ \textbf{where} \\
\quad \mathit{var}_{\mathit{At}} \ : (v : \Gamma \ni \tau) \to \mathit{Atom}\ (\mathit{var}\ v) \\
\quad (@_{\mathit{At}}) \ : \mathit{Atom}\ t_1 \to \mathit{NF}\ t_2 \to \mathit{Atom}\ (t_1 @ t_2) \\
\quad (::_{\mathit{At}}) \ : \mathit{Atom}\ t_1 \to t_1 =_{\vdash} t_2 \to \mathit{Atom}\ t_2 \\
\\
\textbf{data}\ \mathit{NF} : \Gamma \vdash \tau \to * \ \textbf{where} \\
\quad \mathit{atom}^{\star}_{\mathit{NF}} \ : \{t : \Gamma \vdash \star\} \quad \to \mathit{Atom}\ t \to \mathit{NF}\ t \\
\quad \mathit{atom}^{\mathit{El}}_{\mathit{NF}} \ : \{t : \Gamma \vdash \mathit{El}\ t'\} \to \mathit{Atom}\ t \to \mathit{NF}\ t \\
\quad \lambda_{\mathit{NF}} \quad : \mathit{NF}\ t \to \mathit{NF}\ (\lambda\ t) \\
\quad (::_{\mathit{NF}}) \quad : \mathit{NF}\ t_1 \to t_1 =_{\vdash} t_2 \to \mathit{NF}\ t_2
\end{array}$$

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

Context extensions

$$\begin{array}{l}
\textbf{data}\ \mathit{Ctxt}^{+}\ (\Gamma : \mathit{Ctxt}) : * \ \textbf{where} \\
\quad \varepsilon^{+} \quad : \mathit{Ctxt}^{+}\ \Gamma \\
\quad (\rhd^{+}) : (\Gamma' : \mathit{Ctxt}^{+}\ \Gamma) \to \mathit{Ty}\ (\Gamma +\!\!\!+ \Gamma') \to \mathit{Ctxt}^{+}\ \Gamma \\
\\
(+\!\!\!+) : (\Gamma : \mathit{Ctxt}) \to \mathit{Ctxt}^{+}\ \Gamma \to \mathit{Ctxt} \\
\Gamma +\!\!\!+ \varepsilon^{+} \qquad = \Gamma \\
\Gamma +\!\!\!+ (\Gamma' \rhd^{+} \tau) = (\Gamma +\!\!\!+ \Gamma') \rhd \tau
\end{array}$$

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

Values

$$
\begin{aligned}
&\textbf{data}\ \mathit{Val} : \Gamma \vdash \tau \to * \ \textbf{where} \\
&\quad (::_{\mathit{Val}}) : \mathit{Val}\ t_1 \to t_1 =_{\vdash} t_2 \to \mathit{Val}\ t_2 \\
&\quad \star_{\mathit{Val}} \ : \{t : \Gamma \vdash \star\} \quad \to \mathit{Atom}\ t \to \mathit{Val}\ t \\
&\quad \mathit{El}_{\mathit{Val}} : \{t : \Gamma \vdash \mathit{El}\ t'\} \to \mathit{Atom}\ t \to \mathit{Val}\ t \\
&\quad \Pi_{\mathit{Val}} \ : \quad \{t_1 : \Gamma \vdash \Pi\ \tau_1\ \tau_2\} \\
&\qquad\quad \to (f \ : \ (\Gamma' : \mathit{Ctxt}^{+}\ \Gamma) \\
&\qquad\qquad\qquad \to \{t_2 : \Gamma +\!\!+ \Gamma' \vdash \tau_1 \ / \ \mathit{wk}^{*}\ \Gamma'\} \\
&\qquad\qquad\qquad \to (v_2 : \mathit{Val}\ t_2) \\
&\qquad\qquad\qquad \to \mathit{Val}\ ((t_1 /_{\vdash} \mathit{wk}^{*}\ \Gamma') @ t_2)) \\
&\qquad\quad \to \mathit{Val}\ t_1
\end{aligned}
$$

$$
\begin{aligned}
&(@_{\mathit{Val}}) : \mathit{Val}\ t_1 \to \mathit{Val}\ t_2 \to \mathit{Val}\ (t_1 @ t_2) \\
&\mathit{wk}^{\star}_{\mathit{Val}} \ : \mathit{Val}\ t \to (\Gamma' : \mathit{Ctxt}^{+}\ \Gamma) \to \mathit{Val}\ (t /_{\vdash} \mathit{wk}^{*}\ \Gamma')
\end{aligned}
$$

Environments

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

$$
\begin{aligned}
&\emptyset \qquad\ : \varepsilon \Rightarrow \Delta \\
&(\blacktriangleright) \quad : (\rho : \Gamma \Rightarrow \Delta) \to \Delta \vdash \tau \ / \ \rho \to \Gamma \rhd \tau \Rightarrow \Delta
\end{aligned}
$$

$$
\begin{aligned}
&\textbf{data}\ \mathit{Env} : \Gamma \Rightarrow \Delta \to * \ \textbf{where} \\
&\quad \emptyset_{\mathit{Env}} \quad\ : \mathit{Env}\ \emptyset \\
&\quad (\blacktriangleright_{\mathit{Env}}) : \quad \{\rho : \Gamma \Rightarrow \Delta\} \to \{t : \Delta \vdash \sigma \ / \ \rho\} \\
&\qquad\qquad \to \mathit{Env}\ \rho \to \mathit{Val}\ t \to \mathit{Env}\ (\rho \blacktriangleright t) \\
&\quad (::_{\mathit{Env}}) \ : \mathit{Env}\ \rho_1 \to \rho_1 =_{\Rightarrow} \rho_2 \to \mathit{Env}\ \rho_2
\end{aligned}
$$

$$
\mathit{lookup} : (v : \Gamma \ni \tau) \to \mathit{Env}\ \rho \to \mathit{Val}\ (\mathit{var}\ v /_{\vdash} \rho)
$$

Evaluation

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

$$
\begin{aligned}
&[\![\cdot]\!] : \mathit{Tm}^{-}\ t \to \mathit{Env}\ \rho \to \mathit{Val}\ (t /_{\vdash} \rho) \\
&[\![\mathit{var}^{-}\ v]\!]\gamma \qquad = \mathit{lookup}\ v\ \gamma \\
&[\![t_1^{-}\ @^{-}\ t_2^{-}]\!]\gamma \ = ([\![t_1^{-}]\!]\gamma \ @_{\mathit{Val}} \ [\![t_2^{-}]\!]\gamma) \ ::_{\mathit{Val}} \ \ldots \\
&[\![t^{-} \ ::_{\vdash^{-}} \ \mathit{eq}]\!]\gamma = [\![t^{-}]\!]\gamma \ ::_{\mathit{Val}} \ \ldots \\
&[\![\lambda^{-}\ t_1^{-}]\!]\gamma \qquad = \Pi_{\mathit{Val}}\ (\backslash \Delta'\ v_2 \to \\
&\quad [\![t_1^{-}]\!]\ (\mathit{wk}^{\star}_{\mathit{Env}}\ \gamma\ \Delta' \blacktriangleright_{\mathit{Env}} (v_2 \ ::_{\mathit{Val}} \ldots)) \ ::_{\mathit{Val}} \ \ldots \beta \ldots)
\end{aligned}
$$

Normalisation

Object language
Terms
Variables
Substitutions
Term equality
Interpreter
Implicit substitutions
Normal forms
Values
Environments
Evaluation
Normalisation

$$
\mathit{id}_{\mathit{Env}} : (\Gamma : \mathit{Ctxt}) \to \mathit{Env}\ (\mathit{id}\ \Gamma)
$$

$$
\begin{aligned}
&\mathit{normalise} : (t : \Gamma \vdash \tau) \to \mathit{NF}\ t \\
&\mathit{normalise}\ t = \mathit{reify}\ \_\ ([\![\mathit{tmToTm}^{-}\ t]\!]\ \mathit{id}_{\mathit{Env}} \ ::_{\mathit{Val}} \ldots)
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{normaliseEq} : (t : \Gamma \vdash \tau) \to \mathit{nfToTm}\ (\mathit{normalise}\ t) =_{\vdash} t \\
&\mathit{normaliseEq}\ t = \mathit{nfToTmEq}\ (\mathit{normalise}\ t)
\end{aligned}
$$