A Formalisation of a Dependently Typed Language as an Inductive-Recursive Family

Nils Anders Danielsson

Chalmers

January 31, 2007

Introduction

- Abstract syntax data type for dependently typed language.
- No raw terms.
- Full normalisation (NBE).
- Equality checker.
- Type checker.
- Structurally recursive.

Meta language

- AgdaLight (Ulf Norell).
- Inductive-recursive families, implicit arguments.
- "Epigram with Haskell-like syntax."

Object language

- Variant of Martin-Löf's logical framework.
- Explicit substitutions.
- de Bruijn indices.
- Almost all definitions are mutually recursive.

Contexts

data *Ctxt* : *Set* **where**
  ε : *Ctxt*
  $(\triangleright) : (\Gamma : \mathit{Ctxt}) \to \mathit{Ty}\ \Gamma \to \mathit{Ctxt}$

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

Types

data $\mathit{Ty} : \mathit{Ctxt} \rightarrow \mathit{Set}$ **where**

$\star : \mathit{Ty}\ \Gamma$

$\mathit{El} : \Gamma \vdash \star \rightarrow \mathit{Ty}\ \Gamma$

$\Pi : (\tau : \mathit{Ty}\ \Gamma) \rightarrow \mathit{Ty}\ (\Gamma \triangleright \tau) \rightarrow \mathit{Ty}\ \Gamma$

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

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

Types

$$
\begin{aligned}
&\textbf{data}\ \mathit{Ty} : \mathit{Ctxt} \to \mathit{Set}\ \textbf{where} \\
&\quad \star : \mathit{Ty}\ \Gamma \\
&\quad \mathit{El} : \Gamma \vdash \star \to \mathit{Ty}\ \Gamma \\
&\quad \Pi : (\tau : \mathit{Ty}\ \Gamma) \to \mathit{Ty}\ (\Gamma \rhd \tau) \to \mathit{Ty}\ \Gamma
\end{aligned}
$$

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

Terms

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

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

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

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

Terms

$$
\begin{array}{llll}
\multicolumn{4}{l}{\textbf{data}\ (\vdash) : (\Gamma : \mathit{Ctxt}) \to \mathit{Ty}\ \Gamma \to \mathit{Set}\ \textbf{where}} \\
\quad \mathit{var} & : \Gamma \ni \tau & & \to \Gamma \vdash \tau \\
\quad \lambda & : \Gamma \rhd \tau_1 \vdash \tau_2 & & \to \Gamma \vdash \Pi\ \tau_1\ \tau_2 \\
\quad (@) & : \Gamma \vdash \Pi\ \tau_1\ \tau_2 & \to (t_2 : \Gamma \vdash \tau_1) & \to \Gamma \vdash \tau_2 \mathbin{/} \mathit{sub}\ t_2 \\
\quad (::^{\equiv}_{\vdash}) & : \Gamma \vdash \tau_1 & \to \tau_1 =_{\star} \tau_2 & \to \Gamma \vdash \tau_2 \\
\quad (/_{\vdash}) & : \Gamma \vdash \tau & \to (\rho : \Gamma \Rightarrow \Delta) & \to \Delta \vdash \tau \mathbin{/} \rho
\end{array}
$$

$$(::_{\vdash}) : \Gamma_1 \vdash \tau_1 \to \tau_1 =_{\star} \tau_2 \to \Gamma_2 \vdash \tau_2$$

- Note that $(\vdash)$ is indexed by Ty.

Terms

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

$$(::_{\vdash}) : \Gamma_1 \vdash \tau_1 \ \rightarrow\ \tau_1 =_\star \tau_2 \ \rightarrow\ \Gamma_2 \vdash \tau_2$$

- Note that $(\vdash)$ is indexed by Ty.

Variables

$$
\begin{array}{lll}
\textbf{data}\ (\ni) : (\Gamma : \mathit{Ctxt}) \rightarrow \mathit{Ty}\ \Gamma \rightarrow \mathit{Set}\ \textbf{where} \\
\quad \mathit{vz} : & & \Gamma \triangleright \sigma \ni \sigma \,/\, \mathit{wk}\ \sigma \\
\quad \mathit{vs} : & \Gamma \ni \tau \rightarrow & \Gamma \triangleright \sigma \ni \tau \,/\, \mathit{wk}\ \sigma \\
\quad (::^{\equiv}_{\ni}) : & \Gamma \ni \tau_1 \rightarrow \tau_1 =_\star \tau_2 \rightarrow \Gamma \ni \tau_2 &
\end{array}
$$

Substitutions

$$
\begin{aligned}
&\textbf{data}\ (\Rightarrow) : \mathit{Ctxt} \ \rightarrow\ \mathit{Ctxt} \ \rightarrow\ \mathit{Set}\ \textbf{where}\\
&\quad \mathit{sub} : \Gamma \vdash \tau \qquad\ \ \rightarrow \Gamma \triangleright \tau \Rightarrow \Gamma\\
&\quad \mathit{wk}\ \ : (\sigma : \mathit{Ty}\ \Gamma) \rightarrow \Gamma \Rightarrow \Gamma \triangleright \sigma\\
&\quad \mathit{id}\ \ \ : \Gamma \Rightarrow \Gamma\\
&\quad (\odot) : \Gamma \Rightarrow \Delta \ \rightarrow\ \Delta \Rightarrow X \ \rightarrow\ \Gamma \Rightarrow X\\
&\quad (\uparrow)\ \ : \quad (\rho : \Gamma \Rightarrow \Delta) \ \rightarrow\ (\sigma : \mathit{Ty}\ \Gamma)\\
&\qquad\quad \rightarrow\ \Gamma \triangleright \sigma \ \Rightarrow\ \Delta \triangleright (\sigma \,/\, \rho)
\end{aligned}
$$

$$
\begin{aligned}
&\emptyset \quad\ : \varepsilon \Rightarrow \Delta\\
&(\blacktriangleright) : (\rho : \Gamma \Rightarrow \Delta) \ \rightarrow\ \Delta \ \vdash\ \tau \,/\, \rho \ \rightarrow\ \Gamma \triangleright \tau \Rightarrow \Delta
\end{aligned}
$$

Equality

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

Term equality

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

Term equality

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

Term equality

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

Why heterogeneous equality?

- $\mathit{var}\ \mathit{vz} \;/_{\vdash}\; \mathit{sub}\ t =_{\vdash} t$.
- $\sigma \;/\; \mathit{wk}\ \sigma \;/\; \mathit{sub}\ t \stackrel{?}{=} \sigma$.
- With homogeneous equality:
  $\sigma \;/\; \mathit{wk}\ \sigma \;/\; \mathit{sub}\ t =_{\star} \sigma$ proved or postulated.
- Not proved because:
  Very large mutually recursive definition.
- Not postulated because:
  $\tau \;/\; \rho$ would not evaluate.

Why explicit substitutions?

- If $(/_{\vdash})$ were a function: similar problems.

Implicit substitutions

$$
\begin{array}{l}
\textbf{data}\ \mathit{Tm}^{-} : \Gamma \vdash \tau \ \rightarrow\ \mathit{Set}\ \textbf{where} \\
\quad \begin{array}{ll}
\mathit{var}^{-} & : (v : \Gamma \ni \tau) \ \rightarrow\ \mathit{Tm}^{-}\ (\mathit{var}\ v) \\
\lambda^{-} & : \quad \{t : \Gamma \triangleright \tau_1 \vdash \tau_2\} \\
 & \quad \rightarrow\ \mathit{Ty}^{-}\ \tau_1 \ \rightarrow\ \mathit{Tm}^{-}\ t \\
 & \quad \rightarrow\ \mathit{Tm}^{-}\ (\lambda\ t) \\
(@^{-}) & : \mathit{Tm}^{-}\ t_1 \ \rightarrow\ \mathit{Tm}^{-}\ t_2 \ \rightarrow\ \mathit{Tm}^{-}\ (t_1 @ t_2) \\
(::^{\equiv}_{\vdash^{-}}) & : \mathit{Tm}^{-}\ t_1 \ \rightarrow\ t_1 =_{\vdash} t_2 \ \rightarrow\ \mathit{Tm}^{-}\ t_2
\end{array}
\end{array}
$$

$$
\begin{array}{ll}
\mathit{tm}^{-}\mathit{ToTm} & : \{t : \Gamma \vdash \tau\} \ \rightarrow\ \mathit{Tm}^{-}\ t \ \rightarrow\ \Gamma \vdash \tau \\
\mathit{tm}^{-}\mathit{ToTmEq} & : (t^{-} : \mathit{Tm}^{-}\ t) \ \rightarrow\ \mathit{tm}^{-}\mathit{ToTm}\ t^{-} =_{\vdash} t \\
\mathit{tmToTm}^{-} & : (t : \Gamma \vdash \tau) \ \rightarrow\ \mathit{Tm}^{-}\ t
\end{array}
$$

Normal forms

$$
\begin{array}{l}
\textbf{data } \mathit{Atom} : \Gamma \vdash \tau \;\rightarrow\; \mathit{Set} \textbf{ where} \\
\quad \mathit{var}_{\mathit{At}} \;:\; (v : \Gamma \ni \tau) \;\rightarrow\; \mathit{Atom}\,(\mathit{var}\; v) \\
\quad (@_{\mathit{At}}) \;:\; \mathit{Atom}\; t_1 \;\rightarrow\; \mathit{NF}\; t_2 \;\rightarrow\; \mathit{Atom}\,(t_1 @ t_2) \\
\quad (::^{\equiv}_{\mathit{At}}) \;:\; \mathit{Atom}\; t_1 \;\rightarrow\; t_1 =_{\vdash} t_2 \;\rightarrow\; \mathit{Atom}\; t_2
\end{array}
$$

$$
\begin{array}{l}
\textbf{data } \mathit{NF} : \Gamma \vdash \tau \;\rightarrow\; \mathit{Set} \textbf{ where} \\
\quad \mathit{atom}^{\star}_{\mathit{NF}} \;:\; \{t : \Gamma \vdash \star\} \;\rightarrow\; \mathit{Atom}\; t \;\rightarrow\; \mathit{NF}\; t \\
\quad \mathit{atom}^{\mathit{El}}_{\mathit{NF}} \;:\; \{t : \Gamma \vdash \mathit{El}\; t'\} \;\rightarrow\; \mathit{Atom}\; t \;\rightarrow\; \mathit{NF}\; t \\
\quad \lambda_{\mathit{NF}} \;:\; \mathit{NF}\; t \;\rightarrow\; \mathit{NF}\,(\lambda\; t) \\
\quad (::^{\equiv}_{\mathit{NF}}) \;:\; \mathit{NF}\; t_1 \;\rightarrow\; t_1 =_{\vdash} t_2 \;\rightarrow\; \mathit{NF}\; t_2
\end{array}
$$

Context extensions

$$
\begin{array}{l}
\textbf{data}\ \mathit{Ctxt}^{+}\ (\Gamma : \mathit{Ctxt}) : \mathit{Set}\ \textbf{where} \\
\quad \varepsilon^{+} \quad : \mathit{Ctxt}^{+}\ \Gamma \\
\quad (\triangleright^{+}) : (\Gamma' : \mathit{Ctxt}^{+}\ \Gamma) \to \mathit{Ty}\ (\Gamma +\!\!+ \Gamma') \to \mathit{Ctxt}^{+}\ \Gamma
\end{array}
$$

$$
\begin{array}{l}
(+\!\!+) : (\Gamma : \mathit{Ctxt}) \to \mathit{Ctxt}^{+}\ \Gamma \to \mathit{Ctxt} \\
\Gamma +\!\!+ \varepsilon^{+} \qquad\qquad = \Gamma \\
\Gamma +\!\!+ (\Gamma' \triangleright^{+} \tau) = (\Gamma +\!\!+ \Gamma') \triangleright \tau
\end{array}
$$

Values

$$
\begin{array}{l}
\textbf{data } \mathit{Val} : \Gamma \vdash \tau \;\rightarrow\; \mathit{Set} \textbf{ where} \\
\quad (::_{\mathit{Val}}) : \mathit{Val}\ t_1 \;\rightarrow\; t_1 =_{\vdash} t_2 \;\rightarrow\; \mathit{Val}\ t_2 \\
\quad \star_{\mathit{Val}} \;\; : \{\, t : \Gamma \vdash \star \,\} \qquad \rightarrow\; \mathit{Atom}\ t \;\rightarrow\; \mathit{Val}\ t \\
\quad \mathit{El}_{\mathit{Val}} \; : \{\, t : \Gamma \vdash \mathit{El}\ t' \,\} \;\rightarrow\; \mathit{Atom}\ t \;\rightarrow\; \mathit{Val}\ t \\
\quad \Pi_{\mathit{Val}} \;\; : \quad \{\, t_1 : \Gamma \vdash \Pi\ \tau_1\ \tau_2 \,\} \\
\qquad\qquad \rightarrow\; (f \;\; : \;\; (\Gamma' : \mathit{Ctxt}^{+}\ \Gamma) \\
\qquad\qquad\qquad\quad \rightarrow\; \{\, t_2 : \Gamma +\!\!+ \Gamma' \vdash \tau_1 \;/\; \mathit{wk}^{*}\ \Gamma' \,\} \\
\qquad\qquad\qquad\quad \rightarrow\; (v : \mathit{Val}\ t_2) \\
\qquad\qquad\qquad\quad \rightarrow\; \mathit{Val}\ ((t_1 \;/\!\!\vdash\; \mathit{wk}^{*}\ \Gamma') @ t_2)) \\
\qquad\qquad \rightarrow\; \mathit{Val}\ t_1
\end{array}
$$

$$
\begin{array}{l}
(@_{\mathit{Val}}) : \mathit{Val}\ t_1 \;\rightarrow\; \mathit{Val}\ t_2 \;\rightarrow\; \mathit{Val}\ (t_1 @ t_2) \\
\mathit{wk}^{\star}_{\mathit{Val}} \;\; : \mathit{Val}\ t \;\rightarrow\; (\Gamma' : \mathit{Ctxt}^{+}\ \Gamma) \;\rightarrow\; \mathit{Val}\ (t \;/\!\!\vdash\; \mathit{wk}^{*}\ \Gamma')
\end{array}
$$

Environments

$$
\begin{array}{l}
\textbf{data } \mathit{Env} : \Gamma \Rightarrow \Delta \to \mathit{Set} \textbf{ where} \\
\quad \emptyset_{\mathit{Env}} \quad : \mathit{Env}\ \emptyset \\
\quad (\blacktriangleright_{\mathit{Env}}) : \quad \{\rho : \Gamma \Rightarrow \Delta\} \to \{t : \Delta \vdash \sigma \,/\, \rho\} \\
\qquad\qquad \to \mathit{Env}\ \rho \to \mathit{Val}\ t \to \mathit{Env}\ (\rho \blacktriangleright t) \\
\quad (::^{\equiv}_{\mathit{Env}})\ : \mathit{Env}\ \rho_1 \to \rho_1 =_{\Rightarrow} \rho_2 \to \mathit{Env}\ \rho_2
\end{array}
$$

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

Evaluation

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

reify + reflect

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

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

Normalisation

$$id_{Env} : (\Gamma : Ctxt) \;\to\; Env\;(id\;\Gamma)$$

$$normalise : (t : \Gamma \vdash \tau) \;\to\; NF\;t$$
$$normalise\;t = reify\;\_\;(\llbracket tmToTm^{-}\;t \rrbracket\, id_{Env}\;\;::_{Val}\;\ldots)$$

$$normaliseEq : (t : \Gamma \vdash \tau) \;\to\; nfToTm\;(normalise\;t) =_{\vdash}\; t$$
$$normaliseEq\;t = nfToTmEq\;(normalise\;t)$$

- The completeness proof is under way.

Equality

NFs	Strip casts, check syntactic equality.
Terms	Normalise, then check.
Types	Check structurally.

$$
\begin{aligned}
&\textbf{data } \mathit{TyEq?}\ (\tau_1 : \mathit{Ty}\ \Gamma_1)\ (\tau_2 : \mathit{Ty}\ \Gamma_2) : \mathit{Set}\ \textbf{where}\\
&\quad \mathit{equalTy} \quad\ \ : \tau_1 =_\star \tau_2 \ \longrightarrow\ \mathit{TyEq?}\ \tau_1\ \tau_2\\
&\quad \mathit{notEqualTy} : \mathit{TyEq?}\ \tau_1\ \tau_2
\end{aligned}
$$

$$
(\stackrel{?}{=}) : (\tau_1, \tau_2 : \mathit{Ty}\ \Gamma)\ \longrightarrow\ \mathit{TyEq?}\ \tau_1\ \tau_2
$$

Type checker

- Raw terms (*RawTm*).
- Lambdas annotated with raw types.

$$
\begin{aligned}
&\textbf{data}\ \mathit{IsTm}^{-}?\ (\Gamma : \mathit{Ctxt}) : \mathit{RawTm} \longrightarrow \mathit{Set}\ \textbf{where} \\
&\quad \mathit{isTm} \ :\ (\tau : \mathit{Ty}\ \Gamma) \rightarrow (t : \Gamma \vdash \tau) \rightarrow (t^{-} : \mathit{Tm}^{-}\ t) \\
&\qquad\qquad \rightarrow \mathit{IsTm}^{-}?\ \Gamma\ (\mathit{eraseTm}^{-}\ t^{-}) \\
&\quad \mathit{noTm}\ (e : \mathit{RawTm}) : \mathit{IsTm}^{-}?\ \Gamma\ e
\end{aligned}
$$

$$
\mathit{inferTm}^{-} : (\Gamma : \mathit{Ctxt}) \rightarrow (e : \mathit{RawTm}) \rightarrow \mathit{IsTm}^{-}?\ \Gamma\ e
$$

Discussion

- Internal method advantage:
  Types give a lot of info.
  Example: No de Bruijn index arithmetic.
- Disadvantage:
  Sometimes things become very dependent on each other.