

A type system with native homotopy universes

Robin Adams
Andrew Polonsky

The HoTT Workshop
Porto, 25.06.2016

Syntax

$t, A, e ::= \mathbf{1} \mid \text{Prop} \mid \text{Set} \mid \text{Grpd} \mid *_3$
| $P \leftrightarrow Q \mid A \Leftrightarrow B \mid G \Leftrightarrow H \mid \text{Grpd} \simeq_3 \text{Grpd} \mid a \sim_e b$
| $\mathbb{t} \mid x \mid (\prod x:A)B \mid (\sum x:A)B \mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$
| $*^* \mid \prod^*[x, x', x^*]:A^*.B^* \mid \sum^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$
| $r(t) \mid e^+(s) \mid e^-(t) \mid e^=(s, t) \mid \vec{e}(s) \mid \overleftarrow{e}(t)$

Syntax

$t, A, e ::= \mathbf{1} \mid \text{Prop} \mid \text{Set} \mid \text{Grpd} \mid *_3$
| $P \leftrightarrow Q \mid A \Leftrightarrow B \mid G \Leftrightarrow H \mid \text{Grpd} \simeq_3 \text{Grpd} \mid a \sim_e b$
| $\mathbb{t} \mid x \mid (\prod x:A)B \mid (\sum x:A)B \mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$
| $*^* \mid \prod^*[x, x', x^*]:A^*.B^* \mid \sum^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$
| $r(t) \mid e^+(s) \mid e^-(t) \mid e^-(s, t) \mid \vec{e}(s) \mid \overleftarrow{e}(t)$

$$r(A) \quad : \quad A \simeq A$$

$$a \simeq_A a' \quad := \quad a \sim_{r(A)} a'$$

Equality = Path substitution + transport

$$\frac{x : A \vdash B(x) : * \quad a^* : a \simeq_A a'}{B(a^*) : B(a) \simeq B(a')}$$

$$\frac{x : A \vdash b : B \quad a^* : a \simeq_A a'}{b(a^*) : b(a) \sim_{B(a^*)} b(a')}$$

Equality = Path substitution + transport

$$\frac{x : A \vdash B(x) : * \quad a^* : a \simeq_A a'}{B(a^*) : B(a) \simeq B(a')}$$

$$\frac{x : A \vdash b : B \quad a^* : a \simeq_A a'}{b(a^*) : b(a) \sim_{B(a^*)} b(a')}$$

- ▶ Let $B(x) := a \simeq_A x$. Given $\alpha : a \simeq_A a'$, $\alpha' : a' \simeq_A a''$, we have

$$\begin{aligned} B(\alpha') & : (a \simeq_A a') \simeq (a \simeq_A a'') \\ \alpha \circ \alpha' := B(\alpha')^+(\alpha) & : a \simeq_A a'' \end{aligned}$$

Equality = Path substitution + transport

$$\frac{x : A \vdash B(x) : * \quad a^* : a \simeq_A a'}{B(a^*) : B(a) \simeq B(a')}$$

$$\frac{x : A \vdash b : B \quad a^* : a \simeq_A a'}{b(a^*) : b(a) \sim_{B(a^*)} b(a')}$$

- ▶ Let $B(x) := a \simeq_A x$. Given $\alpha : a \simeq_A a'$, $\alpha' : a' \simeq_A a''$, we have

$$B(\alpha') : (a \simeq_A a') \simeq (a \simeq_A a'')$$

$$\alpha \circ \alpha' := B(\alpha')^+(\alpha) : a \simeq_A a''$$

- ▶ Let $B(x) := x \simeq_A a$. Given $\alpha : a \simeq_A a'$, we have

$$B(\alpha) : (a \simeq_A a) \simeq (a' \simeq_A a)$$

$$\alpha^{-1} := B(\alpha)^+(r(a)) : (a' \simeq_A a)$$

Equality = Path substitution + transport

$$\frac{x : A \vdash B(x) : * \quad a^* : a \simeq_A a'}{B((a^*)) : B(a) \simeq B(a')}$$

$$\frac{x : A \vdash b : B \quad a^* : a \simeq_A a'}{b((a^*)) : b(a) \sim_{B((a^*))} b(a')}$$

- ▶ Let $B(x) := a \simeq_A x$. Given $\alpha : a \simeq_A a'$, $\alpha' : a' \simeq_A a''$, we have

$$B((\alpha')) \quad : \quad (a \simeq_A a') \simeq (a \simeq_A a'')$$

$$\alpha \circ \alpha' := B((\alpha'))^+(\alpha) \quad : \quad a \simeq_A a''$$

- ▶ Let $B(x) := x \simeq_A a$. Given $\alpha : a \simeq_A a'$, we have

$$B((\alpha)) \quad : \quad (a \simeq_A a) \simeq (a' \simeq_A a)$$

$$\alpha^{-1} := B((\alpha))^+(r(a)) \quad : \quad (a' \simeq_A a)$$

- ▶ Let $b : B$, with $x \notin \text{FV}(b, B)$. Given $\alpha : a \simeq_A a'$, we have

$$r(B) := B((\alpha)) = B((\)) \quad : \quad B \simeq B$$

$$r(b) := b((\alpha)) = b((\)) \quad : \quad b \sim_{B((\))} b$$

Plan

- ▶ Background
- ▶ The system
- ▶ Formalization

Plan

- ▶ Background
- ▶ The system
- ▶ Formalization

$\lambda \approx$
$$A, t, e ::= * \mid x \mid \Pi x:A.B \mid \Sigma x:A.B$$
$$\mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$$

$\lambda \simeq$
$$\begin{aligned} A, t, e ::= & * \mid x \mid \Pi x:A.B \mid \Sigma x:A.B \mid A \simeq B \mid a \sim_e b \\ & \mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t \\ & \mid *^* \mid \Pi^*[x, x', x^*]:A^*.B^* \mid \Sigma^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^* \end{aligned}$$

$\lambda \simeq$
$$A, t, e ::= * \mid x \mid \Pi x:A.B \mid \Sigma x:A.B \mid A \simeq B \mid a \sim_e b$$
$$\mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$$
$$\mid *^* \mid \Pi^*[x, x', x^*]:A^*.B^* \mid \Sigma^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$$

$$\frac{A : * \quad B : *}{A \simeq B : *} \text{ (\simeq-Form)}$$

$$\frac{e : A \simeq B}{\sim_e : A \rightarrow B \rightarrow *} \text{ (\simeq-Elim)}$$

$\lambda \simeq$

$$A, t, e ::= * \mid x \mid \Pi x:A.B \mid \Sigma x:A.B \mid A \simeq B \mid a \sim_e b$$

$$\mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$$

$$\mid *^* \mid \Pi^*[x, x', x^*]:A^*.B^* \mid \Sigma^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$$

$$\frac{A : * \quad B : *}{A \simeq B : *} \text{ (\simeq-Form)}$$

$$\frac{e : A \simeq B}{\sim_e : A \rightarrow B \rightarrow *} \text{ (\simeq-Elim)}$$

Given $A^* : A \simeq A'$, $B^* : B \simeq B'$, one can construct:

$\lambda \simeq$

$$A, t, e ::= * \mid x \mid \Pi x:A.B \mid \Sigma x:A.B \mid A \simeq B \mid a \sim_e b$$

$$\mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$$

$$\mid *^* \mid \Pi^*[x, x', x^*]:A^*.B^* \mid \Sigma^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$$

$$\frac{A : * \quad B : *}{A \simeq B : *} \text{ (\simeq-Form)}$$

$$\frac{e : A \simeq B}{\sim_e : A \rightarrow B \rightarrow *} \text{ (\simeq-Elim)}$$

Given $A^* : A \simeq A'$, $B^* : B \simeq B'$, one can construct:

- ▶ $x^* A^* B^* : (A \times B) \simeq (A' \times B')$

$\lambda \simeq$

$$A, t, e ::= * \mid x \mid \Pi x:A.B \mid \Sigma x:A.B \mid A \simeq B \mid a \sim_e b$$

$$\mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$$

$$\mid *^* \mid \Pi^*[x, x', x^*]:A^*.B^* \mid \Sigma^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$$

$$\frac{A : * \quad B : *}{A \simeq B : *} \text{ (\simeq-Form)}$$

$$\frac{e : A \simeq B}{\sim_e : A \rightarrow B \rightarrow *} \text{ (\simeq-Elim)}$$

Given $A^* : A \simeq A'$, $B^* : B \simeq B'$, one can construct:

- ▶ $\times^* A^* B^* : (A \times B) \simeq (A' \times B')$
- ▶ $\rightarrow^* A^* B^* : (A \rightarrow B) \simeq (A' \rightarrow B')$

$\lambda \simeq$

$$A, t, e ::= * \mid x \mid \Pi x:A.B \mid \Sigma x:A.B \mid A \simeq B \mid a \sim_e b$$

$$\mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$$

$$\mid *^* \mid \Pi^*[x, x', x^*]:A^*.B^* \mid \Sigma^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$$

$$\frac{A : * \quad B : *}{A \simeq B : *} \text{ (\simeq-Form)}$$

$$\frac{e : A \simeq B}{\sim_e : A \rightarrow B \rightarrow *} \text{ (\simeq-Elim)}$$

Given $A^* : A \simeq A'$, $B^* : B \simeq B'$, one can construct:

- ▶ $\times^* A^* B^* : (A \times B) \simeq (A' \times B')$
- ▶ $\rightarrow^* A^* B^* : (A \rightarrow B) \simeq (A' \rightarrow B')$
- ▶ $\simeq^* A^* B^* : (A \simeq B) \simeq (A' \simeq B')$

$\lambda \simeq$

$A, t, e ::= * \mid x \mid \Pi x:A.B \mid \Sigma x:A.B \mid A \simeq B \mid a \sim_e b$
 $\mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$
 $\mid *^* \mid \Pi^*[x, x', x^*]:A^*.B^* \mid \Sigma^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$

$$\frac{A : * \quad B : *}{A \simeq B : *} \text{ (\simeq-Form)}$$

$$\frac{e : A \simeq B}{\sim_e : A \rightarrow B \rightarrow *} \text{ (\simeq-Elim)}$$

Given $A^* : A \simeq A'$, $B^* : B \simeq B'$, one can construct:

- ▶ $\times^* A^* B^* : (A \times B) \simeq (A' \times B')$
- ▶ $\rightarrow^* A^* B^* : (A \rightarrow B) \simeq (A' \rightarrow B')$
- ▶ $\simeq^* A^* B^* : (A \simeq B) \simeq (A' \simeq B')$
- ▶ $*^* : * \simeq *$

Reduction rules

$$(\lambda x : A. t) a \longrightarrow t[a/x]$$

$$\pi_i(s_1, s_2) \longrightarrow s_i$$

$$A \sim_{*} B \longrightarrow A \simeq B$$

$$f \sim_{\Pi^*[x, x', x^*]: A^*. B^*(x, x', x^*)} f' \longrightarrow \Pi a: A \Pi a': A' \Pi a^* : a \sim_{A^*} a'.$$

$$fx \sim_{B^*(a, a', a^*)} f'x'$$

$$p \sim_{\Sigma^*[x, x', x^*]: A^*. B^*(x, x', x^*)} p' \longrightarrow \Sigma a^* : \pi_1 p \sim_{A^*} \pi_1 p'.$$

$$\pi_2 p \sim_{B^*(\pi_1 p, \pi_1 p', a^*)} \pi_2 p'$$

$$e \sim_{\simeq^* A^* B^*} e' \longrightarrow \Pi a: A \Pi a': A' \Pi a^* : a \sim_{A^*} a'$$

$$\Pi b: B \Pi b': B' \Pi b^* : b \sim_{B^*} b'.$$

$$(a \sim_e b) \simeq (a' \sim_{e'} b')$$

Extensional equality of closed types

THEOREM. There is an operation $(\cdot)^* : \mathcal{T}erms(\lambda \simeq) \rightarrow \mathcal{T}erms(\lambda \simeq)$ such that

$$\Gamma \vdash M : A \implies \Gamma^* \vdash M^* : M \sim_{A^*} M'$$

Extensional equality of closed types

THEOREM. There is an operation $(\cdot)^* : \mathcal{T}erms(\lambda\simeq) \rightarrow \mathcal{T}erms(\lambda\simeq)$ such that

$$\Gamma \vdash M : A \implies \Gamma^* \vdash M^* : M \sim_{A^*} M'$$

In particular,

- ▶ For any closed type $\vdash A : *$, there exist

$$\begin{aligned} \vdash A^* & : A \simeq A \\ \simeq_A & := \sim(A^*) : A \rightarrow A \rightarrow * \end{aligned}$$

Extensional equality of closed types

THEOREM. There is an operation $(\cdot)^* : \mathcal{T}erms(\lambda_{\simeq}) \rightarrow \mathcal{T}erms(\lambda_{\simeq})$ such that

$$\Gamma \vdash M : A \implies \Gamma^* \vdash M^* : M \sim_{A^*} M'$$

In particular,

- ▶ For any closed type $\vdash A : *$, there exist

$$\begin{aligned} \vdash A^* & : A \simeq A \\ \simeq_A & := \sim(A^*) : A \rightarrow A \rightarrow * \end{aligned}$$

- ▶ For any closed term $\vdash a : A$, there exists

$$a^* : a \simeq_A a$$

Our goal

Restricting to a low dimension, work out a system for computing with equalities which is

- ▶ Simple and intuitive
- ▶ Amenable to formalization
- ▶ Feasible to scale to the next dimension, in principle

$\lambda \simeq_2$

$t, A, e ::= \mathbf{1} \mid \text{Prop} \mid \text{Set} \mid \text{Grpd} \mid *_3$
 $\mid P \leftrightarrow Q \mid A \Leftrightarrow B \mid G \Leftrightarrow H \mid \text{Grpd} \simeq_3 \text{Grpd} \mid a \sim_e b$
 $\mid \mathbb{t} \mid x \mid (\prod x:A)B \mid (\sum x:A)B \mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$
 $\mid *^* \mid \prod^*[x, x', x^*]:A^*.B^* \mid \sum^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$
 $\mid r(t) \mid e^+(s) \mid e^-(t) \mid e^-(s, t) \mid \overrightarrow{e}(s) \mid \overleftarrow{e}(t)$

$\lambda \simeq_2$

$t, A, e ::= \mathbf{1} \mid \text{Prop} \mid \text{Set} \mid \text{Grpd} \mid *_3$
 $\mid P \leftrightarrow Q \mid A \Leftrightarrow B \mid G \Leftrightarrow H \mid \text{Grpd} \simeq_3 \text{Grpd} \mid a \sim_e b$
 $\mid \mathbb{t} \mid x \mid (\prod x:A)B \mid (\sum x:A)B \mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$
 $\mid *^* \mid \prod^*[x, x', x^*]:A^*.B^* \mid \sum^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$
 $\mid r(t) \mid e^+(s) \mid e^-(t) \mid e^-(s, t) \mid \vec{e}(s) \mid \overleftarrow{e}(t)$

$$\boxed{\frac{A : *_k \quad B : *_k}{A \simeq_k B : *_k}}$$

$$\boxed{\frac{a : A \quad b : B \quad e : A \simeq_k B}{a \sim_e b : *_{k-1}}}$$

$\lambda \simeq_2$

$t, A, e ::= \mathbf{1} \mid \text{Prop} \mid \text{Set} \mid \text{Grpd} \mid *_3$
 $\mid P \leftrightarrow Q \mid A \Leftrightarrow B \mid G \Leftrightarrow H \mid \text{Grpd} \simeq_3 \text{Grpd} \mid a \sim_e b$
 $\mid \mathbb{t} \mid x \mid (\prod x:A)B \mid (\sum x:A)B \mid \lambda x:A.t \mid st \mid (s, t) \mid \pi_1 t \mid \pi_2 t$
 $\mid *^* \mid \prod^*[x, x', x^*]:A^*.B^* \mid \sum^*[x, x', x^*]:A^*.B^* \mid \simeq^* A^* B^*$
 $\mid r(t) \mid e^+(s) \mid e^-(t) \mid e^-(s, t) \mid \overrightarrow{e}(s) \mid \overleftarrow{e}(t)$

$$\boxed{\frac{A : *_k \quad B : *_k}{A \simeq_k B : *_k}}$$

$$\boxed{\frac{a : A \quad b : B \quad e : A \simeq_k B}{a \sim_e b : *_k}}$$

 $*_2 := \text{Grpd}$ $A \simeq_2 B := A \Leftrightarrow B$ $*_1 := \text{Set}$ $A \simeq_1 B := A \Leftrightarrow B$ $*_0 := \text{Prop}$ $A \simeq_0 B := A \leftrightarrow B$ $*_{-1} := \mathbf{1}$ $a \simeq_A a' := a \sim_{r(A)} a'$

Typing rules

$$\frac{k \in \{0, 1, 2, 3\}}{\vdash *_{k-1} : *_{k}}$$

$$\frac{\Gamma \vdash A : *_{j} \quad \Gamma, x : A \vdash B : *_{k} \quad j, k \in \{0, 1, 2\}}{\Gamma \vdash \Sigma x : A. B : *_{\max(j, k)}}$$

$$\frac{\Gamma \vdash A : *_{j} \quad \Gamma, x : A \vdash B : *_{k} \quad j, k \in \{0, 1, 2\}}{\Gamma \vdash \Pi x : A. B : \begin{cases} *_{2} & j = 2 = k + 1 \\ *_{k} & \text{otherwise} \end{cases}}$$

Typing rules

$$\frac{\Gamma \vdash A : *_{k} \quad k \in \{0, 1, 2, 3\}}{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : *_{k} \quad k \in \{0, 1, 2, 3\}}{\Gamma, y : B \vdash M : A}$$

$$\frac{\Gamma \vdash A : *_{k} \quad \Gamma \vdash B : *_{k} \quad \Gamma \vdash M : A \quad A = B \quad k \in \{0, 1, 2\}}{\Gamma \vdash M : B}$$

Typing rules

$$\frac{e : A \simeq_k B \quad k \in \{0, 1, 2\}}{e^+ : A \rightarrow B \\ e^- : B \rightarrow A}$$

$$\frac{a : A \quad b : B \quad e : A \simeq_k B \quad k \in \{1, 2\}}{e^-(a, b) : (a \simeq_A e^-(b)) \simeq_{k-1} (e^+(a) \simeq_B b) \\ \vec{e}(a) : a \sim_e e^+(a) \\ \overleftarrow{e}(b) : e^-(b) \sim_e a}$$

$$\frac{A : *_k \quad a : A \quad k \in \{0, 1, 2\}}{r(a) : a \simeq_A a}$$

Substitutions

$$\frac{\Gamma, x:A \vdash t : B \quad \Gamma \vdash a : A}{\Gamma \vdash t[a/x] : B[a/x]}$$

| |
|--|
| $\frac{\Gamma, \Delta \vdash t : B \quad [\vec{a}/\vec{x}] \vdash \Gamma \Rightarrow \Delta}{\Gamma \vdash t[\vec{a}/\vec{x}] : B[\vec{a}/\vec{x}]}$ |
|--|

Substitutions

$$\frac{\Gamma, x:A \vdash t : B \quad \Gamma \vdash a^* : a \simeq_A a'}{\Gamma \vdash t[a^*/x] : t[a/x] \sim_{B[a^*/x]} t[a'/x]}$$

| |
|--|
| $\frac{\Gamma, \Delta \vdash t : B \quad [\vec{a}^*/\vec{x}] : [\vec{a}/\vec{x}] \simeq [\vec{a}'/\vec{x}] \vdash \Gamma \Rightarrow \Delta}{\Gamma \vdash t[\vec{a}^*/\vec{x}] : t[\vec{a}/\vec{x}] \sim_{B[\vec{a}^*/\vec{x}]} t[\vec{a}'/\vec{x}]}$ |
|--|

Substitutions

$$\frac{}{\varrho \vdash \Gamma \Rightarrow \emptyset}$$

$$\frac{}{\varrho^* : \varrho \simeq \varrho \vdash \Gamma \Rightarrow \emptyset}$$

Substitutions

$$\frac{[\bar{a}/\bar{x}] \vdash \Gamma \Rightarrow \Delta \quad \Gamma, \Delta[\bar{a}/\bar{x}] \vdash b : B[\bar{a}/\bar{x}]}{[\bar{a}, b/\bar{x}, y] \vdash \Gamma \Rightarrow \Delta, y : B}$$

$$[\bar{a}, b/\bar{x}, y], [\bar{a}', b'/\bar{x}, y] \vdash \Gamma \Rightarrow \Delta, y : B$$

$$[\bar{a}^*//\bar{x}] : [\bar{a}/\bar{x}] \simeq [\bar{a}'/\bar{x}] \vdash \Gamma \Rightarrow \Delta$$

$$\Gamma, \Delta[\bar{a}^*//\bar{x}] \vdash b^* : b \sim_{B[\bar{a}^*//\bar{x}]} b'$$

$$\frac{[\bar{a}^*, b^*//\bar{x}, y] : [\bar{a}, b/\bar{x}, y] \simeq [\bar{a}', b'/\bar{x}, y] \vdash \Gamma \Rightarrow \Delta, y : B}$$

The intended model

Our system has a natural set-theoretic semantics:

- ▶ $\llbracket \text{Prop} \rrbracket = \{0, 1\}$;
- ▶ $\llbracket \text{Set} \rrbracket = \mathbf{V}_\kappa$, for strongly inaccessible κ ;
- ▶ $\llbracket \text{Grpd} \rrbracket =$ the collection of (locally) κ -small groupoids;
- ▶ $p \llbracket \leftrightarrow \rrbracket q = \text{Iff}(p, q)$;
- ▶ $A \llbracket \leftrightarrow \rrbracket B = \text{Iso}(A, B)$;
- ▶ $G \llbracket \Leftrightarrow \rrbracket H = \text{Eq}(G, H)$;
- ▶ $\llbracket (\prod x:A) B \rrbracket = \prod_{a \in \llbracket A \rrbracket} \llbracket B \rrbracket_{x:=a}$
- ▶ $\llbracket (\sum x:A) B \rrbracket = \bigsqcup_{a \in \llbracket A \rrbracket} \llbracket B \rrbracket_{x:=a}$

A strict model

Similar to the previous one, except

- ▶ $\llbracket \text{Grpd} \rrbracket = \mathbf{V}_{\kappa'}$, where $\kappa' > \kappa$.
- ▶ $\llbracket \Leftrightarrow \rrbracket = \llbracket \Leftrightarrow \rrbracket = \llbracket \sim_e \rrbracket := (=_{ZF})$
- ▶ This model actually validates the rule

$$\frac{a \sim_{r(A)} b}{a = b}$$

A meta-theoretic fact

PROPOSITION. Let $\Gamma \vdash A : \text{Prop}$. Let B be such that

$$\Gamma, x:A, \Delta \vdash B : \text{Set}$$

or $\Gamma, x:A, \Delta \vdash B : \text{Grpd}$

Then B is convertible to a term where x does not occur.

Plan

- ▶ Background
- ▶ The system
- ▶ Formalization

The problem of definitional equalities

- ▶ A central issue arising in formalization of type theories is the interpretation of definitional equalities.
- ▶ One approach consists of interpreting *all* equalities propositionally — including beta conversion, and the substitution lemma

$$\llbracket M[N/x] \rrbracket_\rho = \llbracket M \rrbracket_{\rho, x \mapsto \llbracket N \rrbracket_\rho}$$

- ▶ The conversion rule is thereby *not* validated “on the nose”: if A, B are convertible types, the interpretation of $M : A$ is *coerced* from $\llbracket A \rrbracket$ to $\llbracket B \rrbracket$ by identity elimination.
- ▶ Due to this mismatch, it becomes necessary to prove coherence of interpretation with respect to all equalities — major pain in the brain!

”Kipling”-style formalization

- ▶ In “Outrageous but meaningful coincidences”, McBride (2008) shows how dependent type theory can be interpreted in itself preserving all definitional equalities.
- ▶ The heart of the idea: all binders in the language are indexed by the *interpretation* of their domains. In particular, types A in context Γ are families indexed by $\llbracket \Gamma \rrbracket$.
- ▶ In particular, if $x : A \vdash B(x) : *$, and $\vdash (a, a') : A \times A'$, then $\llbracket B(\pi_1(a, a')) \rrbracket = \llbracket B(a) \rrbracket$, since $\llbracket \pi_1(a, a') \rrbracket = \llbracket a \rrbracket$.
- ▶ Using McBride’s technique, we formalized a strict interpretation of $\lambda \simeq$ into a semantic universe defined by induction-recursion.

Recognizing degeneracies

- ▶ Definitional equalities are instances of *degenerate paths*.
- ▶ Once these are added into the syntax (in the form of the $r(t)$ -constructor), a strict interpretation must preserve them as well.
- ▶ This requirement already raises problems when one wants to lift the model to the level of *setoids*.

Strict fibrations

- ▶ A *setoid* is a type A with an equivalence relation $\simeq_A: A \rightarrow A \rightarrow *$.
- ▶ A *fibration* of setoids consists of:
 - ▶ A family of setoids $B(x)$ indexed by $x : A$;
 - ▶ For each $e : a \simeq_A a'$, a setoid isomorphism

$$B(e) : B(a) \simeq B(a')$$

- ▶ The map $e \mapsto B(e)$ must be functorial.
 - ▶ In particular, $B(r(a))$ should be *exactly* the identity isomorphism.
- ▶ This last condition is related to “decidability of degeneracies”.

Our approach

- ▶ The contexts Γ are interpreted by *freely generated setoids*: these are given as the setoid of paths in a graph.
- ▶ The fibration of a setoid over a freely generated one need only specify isomorphisms over the generating edges.
- ▶ This data generates a strict fibration over the generated setoid.
- ▶ Context extension preserves the property of being freely generated.

Conclusion

- ▶ A new type system for reasoning about equalities up to the groupoid level.
- ▶ A strict formalization of λ_{\simeq} .
- ▶ A strict formalization of the λ_{\simeq_2} up to setoids (in progress).
- ▶ A strict formalization of the λ_{\simeq_2} (prospective).

The Shoutout

- ▶ R. Gandy, *On the Axiom of Extensionality*, 1956
- ▶ T. Coquand, *Equality and dependent type theory*.
<http://www.cse.chalmers.se/~coquand/equality.pdf>
- ▶ Tait, *Extensional equality in the classical theory of types*, 1995
- ▶ Altenkirch et al, *Observational type theory*, 2006
- ▶ McBride: *Outrageous but meaningful coincidences*, 2010
- ▶ Martin-Löf: Lectures given at CMU, March 2013