PROOF RELEVANT EQUALITY

$EQ_{Nat} : Nat \to Nat \to \mathcal{U}$ — $EQ_{Nat}\, M\, N$ is inhabited when M is N.

spec: $EQ_{Nat}\, \bar{m}\, \bar{n} = 1$ or $\top$
$EQ_{Nat}\, \bar{m}\, \bar{n} = 0$ or $\bot$ $(m \neq n)$ } meta-reasoning

equality is completed — what does that mean?

Propositions are very different than booleans!

Question: Given $f : Nat \to Nat$ is it the case that
$EQ_{Nat}(M,N) \longrightarrow EQ_{Nat}(fM, fN)$?

ie, do functions on Nat respect EQ_{Nat}?

Answer: Yes. $x, y : Nat \vdash \prod_{f : Nat \to Nat} EQ_{Nat}(x,y) \to EQ_{Nat}(fx, fy)$.

double induction, with forall inside the induction
So induction hypothesis is strong enough.

Proof Sketch:

$x = zero = y$: $\lambda f.\, \lambda \_ .\, refl\,(f\ zero)$.
$EQ_{Nat}(zero, zero) \equiv 1$

$x = zero$, $y = succ(\_)$: $EQ_{Nat}(zero, succ\_) = 0$.
so $\lambda f.\, \lambda z{:}0.\, abort(z)$.

$x = succ\_$, $y = zero$: $EQ_{Nat}(succ\_, zero) \equiv 0$
$\lambda f.\, \lambda z{:}0.\, abort\, z$.

$x = succ\, x'$, $y = succ\, y'$: $EQ_{Nat}(succ(x'), succ(y')) = EQ_{Nat}(x', y')$

IH: $\prod f : Nat \to Nat\ EQ_{Nat}\, x'\, y' \to EQ_{Nat}(f x', f y')$
want to show: $\prod f : Nat \to Nat.\ EQ_{Nat}(x,y) \to EQ_{Nat}(fx, fy)$.
Suppose $f : Nat \to Nat$
$p : EQ_{Nat}(x,y) \equiv EQ_{Nat}(x', y')$
take f in IH to be $f \circ succ$, ie:
$IH\,(f \circ succ)\,(p) : EQ_{Nat}((f \circ succ)(x'), (f \circ succ)(y'))$
($(f \circ succ)(x')$ = fx; $(f \circ succ)(y')$ = fy)

Define EQ for other types:

$EQ_1 \equiv \lambda x.\, \lambda y.\, 1$
$EQ_0 \equiv \lambda x.\, \lambda y.\, 1$
$EQ_{A \times B} \equiv EQ_A \times EQ_B = \lambda x.\, \lambda y.\, EQ_A(fst\, x, fst\, y) \times EQ_B(snd\, x, snd\, y)$
* $EQ_{A+B} \equiv EQ_A + EQ_B = ?$
$EQ_{A \to B} \equiv EQ_A \to EQ_B \equiv \lambda f.\, \lambda g.\, (\prod x,y{:}A.\ EQ_A(x,y) \to EQ_B(fx, gy))$

* $EQ_{\Sigma x:A.B} = ?$
* $EQ_{\Pi x:A.B} = ?$

Question:

$A : \mathcal{U}, x : A \vdash refl_A(x) : EQ_A(x,x)$.

Consider $A \triangleq (Nat \to Nat) \to Nat$

To Show: $F : (Nat \to Nat) \to Nat \vdash \_ : EQ_A(F,F)$

$\_ : \Pi f,g : Nat \to Nat.\ EQ_{Nat\to Nat}(f,g) \to EQ_{Nat}(F(f), F(g))$.

$\_ : \Pi f,g : Nat \to Nat.\ (\Pi x,y : Nat.\ EQ_{Nat}(x,y) \to EQ_{Nat}(fx, gx)) \to EQ_{Nat}(F(f), F(g))$.

Cannot be written in OTT as we have it currently.
Because "Functions" can't be reasoned about.
$A \to B$ is not the type of functions.

What to do?

Define $ELT_A : A \to \mathcal{U} \triangleq \lambda x.\ EQ_A(x,x)$.

with that:

$\Pi x : A.\ \underbrace{ELT_A(x)}_{\text{"valid" "reasonable" "extensional"}} \to EQ_A(x,x)$.

Notation: $M \in A$ iff $\_ : ELT_A(M)$.

So $M = N \in A$ iff $\_ : EQ_A(M,N)$ (presuppose $M \in A$, $N \in A$).

} realizability / type retrieval / HEO / SETOIDS

Example of not well defined element:

Church's Law $\to CL : \Pi F : Nat \to Nat.\ \Sigma e : Nat.\ EQ_{Nat\to Nat}(f, \{e\})$ or $\Pi x : Nat.\ EQ_{Nat}(f(x), \{e\}x)$

Gödel number of turing machine / lambda term.

Fact

$(\Pi x : A.\ \Sigma y : B.\ R(x,y)) \to (\Sigma f : A \to B.\ \Pi x : A.\ R(x, fx))$ ← de Compiler

Consequence: $\Sigma F : (Nat \to Nat) \to Nat.\ \Pi f : Nat \to Nat.\ EQ_{Nat\to Nat}(f, \{F(f)\})$

extensional equality

So:

1. Functions are extensional - No church's law } Constructive Math
   - consistent with classical math
2. Functions are not extensional - Have church's law } recursive Math
   - inconsistent with classical math

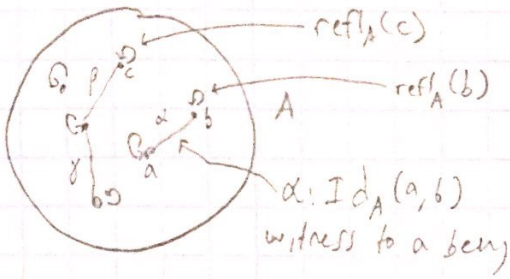
What is equality? "That which everything respects."

IDENTIFICATION TYPE

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash M : A \quad \Gamma \vdash N : A}{\Gamma \vdash Id_A(M,N) : \mathcal{U}}\ F$$

the type of identifications of M and N.

Notation: Id_A is written $M =_A N$ — this is a type



$$\frac{\Gamma \vdash M : A}{\Gamma \vdash refl_A(M) : Id_A(M,M)}\ I$$

self identification

"motive"

$$\cancel{\Gamma \vdash A : \mathcal{U}} \quad \cancel{\Gamma \vdash M : A} \quad \cancel{\Gamma \vdash N : A} \quad \Gamma, x:A, y:A, z:Id_A(x,y) \vdash C_{x,y,z} : \mathcal{U}$$
$$\Gamma \vdash \alpha : Id_A(M,N)$$
$$\Gamma, x:A \vdash c : [x,x,refl_A(x)/x,y,z]C$$

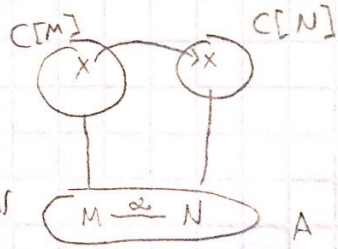
$$\Gamma \vdash J[x,y,z.C](x.c)(\alpha) : [M,N,\alpha/x,y,z]C$$

or

$$\Gamma, x:A, y:A, z:Id(x,y) \vdash J[x,y,z.C](z) : C_{x,y,z}$$

$$J[x,y,z.C](x.c)(refl_A(M)) \equiv [M/x]c : [M,M,refl_A(M)]C \quad \text{computational rule}$$

Special case "transport"



$x:A \vdash C_x : \mathcal{U}$ family of types or predicate

$$\frac{\Gamma \vdash \alpha : Id_A(M,N)}{\Gamma \vdash tr[x.C](\alpha) : [M/x]C \to [N/x]C}$$

if have proof of C for M, can get proof of C for N

$$tr[x.C](refl_A(M)) \equiv id_{C[M]}$$

How is CL an example of an unreasonable function?