$\mathrm{id}\ (A \supset B) = \lambda(\mathcal{E})$

$$\dfrac{\Gamma \atop \mathcal{E}}{A \supset B} = \dfrac{A \supset B \qquad \dfrac{\dfrac{\Gamma \atop \mathcal{E}}{A \supset B} \quad \overline{A}}{B} \supset E}{A \supset B} \supset I$$

$i \cdot \forall f = \lambda(i \times A \cdot f)$

$$\dfrac{\Gamma \atop {\mathcal{E} \atop C} \qquad {\overline{C \quad A} \atop {\otimes \atop B}}}{A \supset B} \supset I \qquad \dfrac{{\overline{\Gamma} \atop {\mathcal{E} \atop C}} \quad \overline{A}}{{\otimes \atop B}}$$

for negative connectives *

- Interpreted as right adjoint functors
  - $[\![* I]\!] := \_^{b}$
  - $[\![* E]\!] := \varepsilon$
  - $[\![* \Rightarrow_R]\!] := \beta_r$
  - $[\![* \Rightarrow_E]\!] :=$ identity expansion

Color Code for Nodes

- —— : Whiteboard
- —— : Exercises!
- —— : Annotations from during lecture
- —— : Annotations from discussion

Q: $\underline{\mathrm{refl}} :\ \Pi x : Nat.\ EQ_{Nat}(x, x)$ (previous lecture)

Rob Harper, Day 4

Proof-relevant equality

$EQ_{Nat} : Nat \to Nat \to \mathcal{U}$ ← Proposition that these 2 numbers are equal

Equality not just some Boolean checks!

spec: informal!
$$\begin{cases} EQ_{Nat}\ \bar{m}\ \bar{m} \equiv \underline{1}\ (\top) & \text{— different notation} \\ EQ_{Nat}\ \bar{m}\ \bar{n} \equiv \underline{0}\ (\bot)\ (m \neq n) \end{cases}$$

$\underline{1}$ → "the type EQ_{Nat}" is inhabited in that case"

$\underline{0}$ → "type NOT inhabited"

$EQ_{Nat}\ M\ N$ is inhabited when M is N.

Side Remark: Assertions are not Boolean tests, but just a piece of code that is executed!

Question: Given $f : Nat \to Nat$ is it the case that for $M, N : Nat$
↳ mathematical function

Is this type inhabitable? → $(?) : EQ_{Nat}(M, N) \to EQ_{Nat}(f\ M, f\ N)$?

"$\forall f : Nat \to Nat$"

i.e. do functions on Nat respect EQ_{Nat}?

⇒ Yes $\quad x, y : Nat \vdash \prod_{f : N \to M} EQ_{Nat}(x, y) \to EQ_{Nat}(f x, f y)$.

Proof: double induction on x, y.

For the proof: strengthen the IH! State the theorem for arbitrary f! Then you can choose f later!

Adjunctions

Antiparallel Functors $F^{:\mathbb{C}\to\mathbb{D}}$ and $G^{:\mathbb{D}\to\mathbb{C}}$ form an adjunction ("$F \dashv G$")
if for any $A : \mathbb{C}$, $B : \mathbb{D}$ there is a natural bijection of hom sets:
(F is left adjoint
G is right adjoint)

- bijection $_{\flat}\left(\dfrac{\mathbb{C}(A \to G(B))}{\mathbb{D}(F(A) \to B)}\right)^{\#}$

- naturality

$$\frac{\mathbb{C} : A' \xrightarrow{a} A \xrightarrow{f = g^{\flat}} G(B) \xrightarrow{G(b)} G(B')}{\mathbb{D} : F(A') \xrightarrow[F(a)]{} F(A) \xrightarrow{g = f^{\#}} B \xrightarrow[b]{} B'}$$

$F : \mathbb{C} \to \mathbb{D}$ and $G : \mathbb{D} \to \mathbb{C}$ form $F \dashv G$ if

different characterization of adjunction

Universal Property of unit:

$\exists \overset{\text{unit of } F \dashv G}{\eta} : id(\mathbb{C}) \longrightarrow F \cdot G$

$\forall f : \mathbb{C}(A \to G(B)).$

$\exists!\, g : \mathbb{D}(F(A) \longrightarrow B).$

$\eta(A) \cdot G(g) = f$

$$\begin{array}{l} \mathbb{C} : \quad A \xrightarrow{\eta(A)} (G \circ F)(A) \xrightarrow{G(g)} G(B), \quad A \xrightarrow{f} G(B) \\ \hline \mathbb{D} \quad F(A) \dashrightarrow_{g} B \end{array}$$

Duality

Universal property of counit

$\exists\, \varepsilon : G \cdot F \to id(\mathbb{D})$

$\forall g : \mathbb{D}(F(A) \to B)$

$\exists! f : \mathbb{C}(A \to G(B))$

$F(f) \cdot \varepsilon(B) = g$

$$\begin{array}{l} \mathbb{C} : \quad A \xrightarrow{f} G(B) \\ \hline \mathbb{D} : \quad F(A) \xrightarrow{g} B, \quad F(A) \xrightarrow{F(f)} (F \circ G) B \xrightarrow{\varepsilon(B)} B \end{array}$$

$$\mathbb{C} \underset{A \supset -}{\overset{- \times A}{\rightleftarrows}} \mathbb{C} \quad (\perp)$$

$$\begin{array}{l} \mathbb{C} \quad X \xrightarrow{\lambda(f)} A \supset B \\ \hline \mathbb{C} \quad X \times A \xrightarrow{f} B, \quad X \times A \xrightarrow{\lambda f \times A} (A \supset B) \times A \xrightarrow{\varepsilon(B)} B \end{array}$$

$$\dfrac{\dfrac{\dfrac{\overline{\Gamma \; A}}{\Gamma \quad B}}{A \supset B}\supset I \qquad A}{B}\supset E \qquad \Longrightarrow_R \qquad \begin{array}{c}\Gamma \; A \\ \otimes \\ B\end{array}$$

local soundness from this property?

Case $x = zero = y$: $\lambda f. \lambda \_.\ refl\ (f\ zero)$.

$EQ_{Nat}(zero, zero) \equiv 1$ ← aka unit aka $\top$

Case $x = zero, y = succ(\_)$: $EQ_{Nat}(zero, succ\ \_) \equiv 0$ aka void aka $\bot$

$\lambda f. \lambda z : 0.\ abort(z)$ ← abort is the only term that is $\bot$!

Case $x = succ(\_), y = zero$: $EQ_{Nat}(succ\ \_, zero) \equiv 0$

$\lambda f. \lambda z : 0.\ abort(z)$

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

$IH : \Pi f : N \to N\ \ \underline{EQ_{Nat}\ x'\ y'} \to EQ_{Nat}(f\ x', f\ y')$

(to show) $TS : \Pi f : N \to N\ \ \underline{EQ_{Nat}\ x\ y} \to EQ_{Nat}(f\ x, f\ y)$

suppose $f : N \to N$

"like reduction" (of the term with this type)

$p : EQ_{Nat}(x, y) \equiv EQ_{Nat}(x', y')$ from the assumptions!

$TS : x : EQ_{Nat}(f\ x, f\ y)$

Take f in IH to be $f \circ \underline{succ}$

$\to IH(f \circ succ)(p) : EQ_{Nat}((f \cdot succ)(x'), (f \cdot succ)(y'))$

$\qquad\qquad \equiv f x \qquad\qquad \equiv f y$

this depends a lot on the code, whether your code really gives you these equality

$\square$

$EQ_1 = \lambda x. \lambda y. \bot$

$EQ_0 = \lambda x. \lambda y. \bot$

$EQ_{A \times B} = EQ_A \times EQ_B = \lambda x. \lambda y.\ EQ_A(fst\ x, fst\ y) \times EQ_B(snd\ x, snd\ y)$

$\boxed{Ex! \to EQ_{A+B} = EQ_A + EQ_B}$

functions are eq if they yield eq. values for eq args!

$EQ_{A \to B} = EQ_A \to EQ_B = \lambda f. \lambda g.\ \underbrace{\Pi x, y : A\ \ EQ_A(x, y)}_{\text{"for } x, y \text{ from } A \text{ that are equal"}} \to EQ_B(f x, g y)$

$\boxed{Ex! \to EQ_{\Sigma x : A. B} = \ ?}$

$\boxed{Ex! \to EQ_{\Pi x : A. B} = \ ?}$

Question: $A : U, x : A \vdash refl_A(x) : EQ_A(x)(x)$?

(Is this type inhabitable for $A : U$?)

consider $A \triangleq (Nat \to Nat) \to Nat$ →

②

$\underline{TS}$: $F : (Nat \to Nat) \to Nat \vdash \_ : EQ_A(F, F)$

expand def of $EQ_{A \to B}$

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

expand def further!

$\boxed{\Pi x, y \; EQ_{Nat}(x, y) \to EQ_{Nat}(f\,x, f\,y)}$

$\_ :$

inhabitable??

$$\boxed{\Pi f, g \; (\Pi x, y : Nat. EQ_{Nat}(x, y) \to EQ_{Nat}(f\,x, f\,y)) \to EQ_{Nat}(F(f), F(g)).}$$

not provable! only contains variables for functions, and we cannot say anything about function equality like we want it so far!

this term is not reducible, since the functions are variables, you don't know anything about them

cannot be written in DTT so far.
(dependent type theory)

Functions are not extensional in this theory!?

$A \to B$ is not a type of functions ~~in extension~~.

extensional functions: the $\underline{\text{observed behavior}}$ of 2 functions is the same

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

$\Pi x : A \quad \underline{ELT_A(x)} \to EQ_A(x, x)$

there could be elements in A that are not "reasonable"

"valid"
"reasonable"
"EXTENSIONAL"
correct term

A is a type for which reflexivity holds

Church's law:

(scientific law!
→ every programming language you can write is eventually like the λ calculus / Turing machines (at least for natural number)

Q: Example for "unreasonable" element?

A:

$\underline{\text{Notation}}$ $M \in A$ iff $\_ : ELT_A(M)$
$M = N \in A$ iff $\_ : EQ_A(M, N)$
presuppose $M \in A, N \in A$.

realizability
type refinement
HEO
SETOID

(Axiom) $CL : \Pi f : N \to N \; \Sigma e : N. \; EQ_{Nat \to Nat}(f, \underline{\{e\}})$
(Church's Law)

$\Pi x : Nat. \; EQ_{Nat}(f(x), \{e\}(x))$ interpret e

e: Gödel number?
"Turing machine code"

$$\boxed{\text{Fact} \quad (\Pi x : A. \Sigma y : B. \; R(x, y)) \to (\Sigma f : A \to B. \; \Pi x : A. \; R(x, \underline{f\,x}))}$$

total relation

we choose an element for every $x : A$

← Ex! (Prove!)

from this restate CL:
"In DTT every total relation contains a functions

$$\Sigma F : (N \to N) \to N.$$

$$\underbrace{\Pi f : N \to N.\ EQ_{Nat \to Nat}\left(f, \{\underbrace{F(f)}_{\text{decompiler}}\}\right)}_{\text{ext. equality}}$$

Apply fact by taking a function that has e as output!

F gives you Gödel number for an arbitrary function! ↳ i.e. with F you good decide extensional eq. of two functions

Alternative points of view:

All ":" in CL would become "∈" but then N would be countable! ???

?

1. Functions are extensional
   × CL
   (Church's law does not hold!)
   consistent with classical math → you cannot construct F in classical math
   } constructive mathematics

2. Functions are not extensional
   ✓ CL
   (Church's law does hold!)
   inconsistent with classical math
   } recursive mathematics ← Russian school

Problem with Setoids: You have to provide equality for M. (↱ impose a notion for equality)
You could do that wrong.

(End of Answer)

What is equality? "That ~~with~~ which everything respects"

Type theory does not know that EQ is equality, i.e. EQ exists outside of type theory, you have to simulate it.

Leibniz Principle: 2 things are equal if you cannot tell them apart no matter what you do to them!

IDENTIFICATION TYPE

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

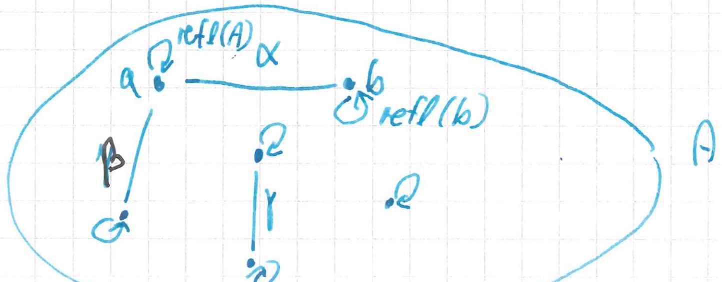
(terms of type A)

← Formation rule! (Introduction rule for the type Id)

Equality is no longer a property of 2 elements, but 2 elements are connected by an identity!

type of identifications of M and N

Notation: $\boxed{M =_A N}$ type!



$\alpha : Id_A(a, b)$
↳ witness of identification from a to b

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathrm{refl}_A(M) : Id_A(M,N)}\, I$$

self-identification

> If this rule was the only introduction rule, refl is the least reflexive relation

Elimination $\cancel{[\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash M : A \quad \Gamma \vdash N : A]}$ $\quad \Gamma, x:A, y:A, z:Id_A(x,y) \vdash C_{x,y,z} : \mathcal{U}$ proposition

"motive"

C may reference x, y, and z (dependent type)

$$\frac{\Gamma \vdash \underline{\alpha} : Id_A(M,N) \qquad \Gamma, x:A \vdash c : \boxed{[x, x, \mathrm{refl}_A(x) / x,y,z]\, C}}{\Gamma \vdash J[x,y,z.C](x.c)(\alpha) : \boxed{[M,N,\alpha / x,y,z]\, C}}\,(E)$$

"proof of equality"

free vars of C substituted

hypothetical "proof using refl" (using same x for both args)

proof term

"mapping out property"

If C is a reflexive relation and α the least reflexive relation (which it is if I from above is the only intro rule), then I can just map out of C.

> rule says: "If we can construct a proof (c) using refl_A, we can use z for the identity proof"

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

$$J[x,y,z.C](x.c)(\mathrm{refl}_A(M)) \equiv \underline{[M/x]c} : [M,N,\mathrm{refl}_A(M)]\,C$$

> If I have a proof of c for M then I can get "proof" for C with N. (assuming identity)

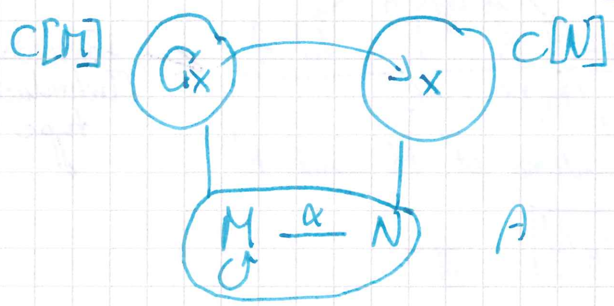
> THE BOXES ARE WHAT MATTERS REALLY

Special case "transport"

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

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

logically equal! But logical equivalence is relatively weak



$$tr[x.C](\mathrm{refl}_A(M)) \equiv id_{C[M]}$$