Equality types of various types

<u>bool</u>: $(\mathrm{true} = \mathrm{true}) \simeq (\mathrm{false} = \mathrm{false}) \simeq \mathbb{1}$ (Similar for $\mathbb{N}$, other inductive types)

$\underline{\Sigma}$: $(s = t) \simeq \left(\sum_{p : \pi_1 s = \pi_1 t} \mathrm{tr}_p\, \pi_2 s = \pi_2 t \right)$ (Similar for $\times$)

What about other types?

$\underline{\Pi}$. $(f = g) \simeq (f \sim g) :\doteq \prod_{a:A} fa = ga$

We cannot prove this (models which do not validate it.)

Instead, we postulate it.

<u>Def</u>. There is a canonical function

$\mathrm{id\text{-}to\text{-}homot} : (f = g) \to (f \sim g)$.

We define

$\mathrm{FunExt} := \mathrm{isEquiv}(\mathrm{id\text{-}to\text{-}homot})$.

<u>Axiom</u>. We assert

$\mathrm{funext} : \mathrm{FunExt}$.

Now we can redefine equivalence by

Def. $\mathrm{isEquiv}(f) := \sum_{g:B\to A} gf = \mathrm{id}_A \times fg = \mathrm{id}_B$.

h-levels

Def. A type T is contractible if

$$\mathrm{isContr}(T) := \sum_{c:T} \prod_{x:T} c = x.$$

Def. For every $n:\mathbb{N}$ and type T, we define $\mathrm{ishlevel}_n(T)$ by induction on n:

$$\mathrm{ishlevel}_0(T) := \mathrm{isContr}(T)$$

$$\mathrm{ishlevel}_{n+1}(T) := \prod_{x,y:T} \mathrm{ishlevel}_n(x=y).$$

(Rijke starts at -2)

Prop. $\mathrm{ishlevel}_1 T \simeq \mathrm{isProp}\, T$

Def. We call a type T a <u>set</u> if $\mathrm{ishlevel}_2 T$.

Def. We call a type T a <u>groupoid</u> if $\mathrm{ishlevel}_3 T$.

Def. We call a type T an <u>n-groupoid</u> if $\mathrm{ishlevel}_{n+2} T$.

⇝ Intuition from homotopy theory:

a ~~type~~ space is a n-groupoid if all m-homotopy groups are trivial for $m > n$.

Prop. (Exercise 5) Each $\mathrm{isHlevel}_n T$ is a proposition.

Lem. $\mathrm{isContr}(T)$ is always a proposition.

Lem. If $\prod_{a:A} \mathrm{isProp}(B\,a)$, then $\mathrm{isProp}\left(\prod_{a:A} B\,a\right)$.

Pf. Suppose we have $f : \prod_{a:A} \mathrm{isProp}(B\,a)$ and $p, q : \prod_{a:A} B\,a$. To show $p = q$, we use function extensionality so that is suffices to prove $p\,a = q\,a$ for all a. But given such an a, $f\,a\,pa\,qa : pa = qa$. $\square$

Pf of prop. Follows by induction. $\square$

Now we can define universes of propositions, sets, etc.

Def. $\mathrm{Prop} := \sum_{T:U} \mathrm{isProp}(T)$

Def. $\mathrm{Set} := \sum_{T:U} \mathrm{isSet}(T)$

Cor. Given two propositions P, Q, we have

$$(P \underset{\mathrm{Prop}}{=} Q) \simeq \sum_{e : P \underset{\mathcal{U}}{=} Q} (\mathrm{tr}_e\, \pi_2 P = \pi_2 Q) \simeq \mathbb{1}$$

(That $\simeq$ is a congruence will become obvious.)

$$\simeq \sum_{e : P \underset{\mathcal{U}}{=} Q} \mathbb{1} \simeq P = Q.$$

(Ex 10.7 b of Rijke)

Prop. In general, for a dependent type $B : A \to \mathcal{U}$ where $\prod_{a:A} \mathrm{isProp}(Ba)$ holds,

$$(a \underset{\sum_{x:A} B}{=} a') \simeq (\pi_1 a \underset{A}{=} \pi_1 a').$$

Thus, we can treat such a Σ-type as a <u>subtype</u> of A.

<u>Universal properties</u>

Prop. There is a canonical map

$$\varepsilon : (A \times B \to Z) \to (A \to B \to Z)$$

given by $\varepsilon f\, a\, b :\doteq f(a,b)$. It is natural in Z in the sense that

$$\begin{tikzcd}
(A \times B \to Z) \arrow[r, "\varepsilon_Z"] \arrow[d, "f_*"'] & (A \to B \to Z) \arrow[d, "f_*"] \\
(A \times B \to Z') \arrow[r, "\varepsilon_{Z'}"] & (A \to B \to Z')
\end{tikzcd}$$

commutes for any $f : Z \to Z'$.

This is an equivalence.

Pf. We can produce an inverse

$$\eta : (A \to B \to Z) \longrightarrow (A \times B \to Z)$$

by doing $\times$-induction. We send

$$\eta\, f\, (a,b) :\doteq f\, a\, b.$$

Then $\varepsilon\eta : (A \to B \to Z) \longrightarrow (A \to B \to Z)$

and $\varepsilon \underbrace{\eta\, f}\, a\, b \doteq (\eta f)(a,b)$

$$\doteq f\, a\, b$$

so $\varepsilon\eta \sim \mathrm{id}$ and $\varepsilon\eta = \mathrm{id}$. (Note we need funext to say this is an equivalence, in the original sense.)

We have $\eta\varepsilon : (A \times B \to Z) \longrightarrow (A \times B \to Z)$

and $\eta\, \underbrace{\varepsilon\, f}\, (a,b) \doteq \varepsilon\, f\, a\, b$

$$\doteq f(a,b)$$

so $\eta\varepsilon \sim \mathrm{id}$ and $\eta\varepsilon = \mathrm{id}$.

Cor. Any P with an equivalence (natural in Z)

$$(P \to Z) \xrightarrow{\sim} (A \to B \to Z)$$

is equivalent to $A \times B$.

Pf. Composing equivalences, we get

$$\varepsilon : (A\times B \to Z) \simeq (P \to Z), \text{ natural in } Z.$$

Now taking $Z := A\times B$, we get $\varepsilon\, id_{A\times B} : P \to A\times B$.

Similarly, we have $\varepsilon^{-1}\, id_P : A\times B \to P$.

$$\begin{tikzcd}
(A\times B \to \overset{A\times B}{Z}) \arrow[r, "\varepsilon"] \arrow[d, "(\varepsilon^{-1} id_P)_*"'] & (P \to \overset{A\times B}{Z}) \arrow[d, "(\varepsilon^{-1} id_P)_*"] \\
(A\times B \to \overset{P}{Z'}) \arrow[r, "\varepsilon"] & (P \to \overset{P}{Z'})
\end{tikzcd}$$

$$\begin{tikzcd}
id_{A\times B} \arrow[r, mapsto] \arrow[d, mapsto] & \varepsilon\, id_{A\times B} \arrow[d, mapsto] \\
\varepsilon^{-1} id_P \arrow[r, mapsto] & id_P \qquad \varepsilon^{-1} id_P \circ \varepsilon\, id_{A\times B}
\end{tikzcd}$$

The commutativity of the above diagram shows

$$\varepsilon^{-1} id_P \circ \varepsilon\, id_{A\times B} = id_P.$$

Similarly, we can show

$$\varepsilon\, id_{A\times B} \circ \varepsilon^{-1} id_P = id_{A\times B},$$

so $P \simeq A\times B$. $\square$

Propositional truncation - first higher inductive type

If A, B are propositions, so are

$A\times B \qquad \neg A \;(A\to\emptyset) \qquad \mathbb{1}$

$A\to B \qquad \emptyset$

We also have that if each $B(a)$ is a proposition, then

$$\prod_{a:A} B(a)$$

is. So the world of propositions looks like first order logic, but w/o $\vee$ and $\exists$.

If we take $\mathbb{1} \vee \mathbb{1}$ or $\sum_{a:\text{bool}} \mathbb{1}$, this is not a proposition.

<u>Def</u>. Given a type T, we say a proposition $\|T\|$ together with a function $\eta : T \to \|T\|$ is a <u>propositional truncation</u> if for every proposition P,

$$\eta^* : (\|T\| \to P) \to (T \to P)$$

is an equivalence.

I.e., $\|T\|$ is a propositional truncation if every $T \to P$ factors uniquely as $T \xrightarrow{\eta} \|T\| \to P$.

<u>Lem</u>. A propositional truncation of a type T is unique up to equivalence.

<u>Ex</u>. If T is inhabited, then $\|T\| \simeq \mathbb{1}$.

<u>Pf</u>. We take $\eta : T \to \mathbb{1}$ to be the unique map.

We produce an inverse to η^*.

$$\varepsilon : (T \to P) \longrightarrow (\mathbb{1} \to P)$$
$$f \longmapsto * \mapsto ft$$

where $t : T$ (hypothesis).

Then $\varepsilon \eta^* : (\mathbb{1} \to P) \longrightarrow (\mathbb{1} \to P)$

$$f \longmapsto * \longmapsto \varepsilon(\eta^* f) *$$
$$\doteq \varepsilon (f \eta) *$$
$$\doteq (f \eta)\, t$$
$$\doteq f *$$

so $\varepsilon \eta^* = \mathrm{id}$.

Similarly $\eta^* \varepsilon : (T \to P) \longrightarrow (T \to P)$

$$f \longmapsto t \longmapsto (\eta^* \varepsilon) f t$$
$$\doteq ft$$

so $\eta^* \varepsilon = \mathrm{id}$.

Therefore, $\mathbb{1}$ is a propositional truncation for T.

<u>Rules</u>. (Higher inductive type)

$\|-\|$-form

$$\frac{T \text{ type}}{\|T\| \text{ type}}$$

$\|-\|$-intro

$$\frac{t : T}{\|t\| : \|T\|}$$

$$\frac{s, t : \|T\|}{e_{s,t} : s =_{\|T\|} t}$$

$\|-\|$-elim

$$\frac{\begin{array}{l} x : \|T\| \vdash D x \text{ type} \\ t : T \vdash d_0 t : D\|t\| \\ s, t : \|T\|,\ u : Ds,\ v : Dt \\ \qquad \vdash d_1(s,t) : \mathrm{tr}_{e_{s,t}} u =_{Dt} v \end{array}}{x : \|T\| \vdash \|-\|\text{-ind}\,(d_0, d_1) : Dx}$$

Prop. This provides propositional truncations for any type.

Now we can set

$$\exists_{x:A} Bx := \| \sum_{x:A} Bx \|$$