<u>Rappel</u> A Proposition si $(x,y:A) \to x=y$

A set if $(x,y:A)$ the type $x=y$ is a proposition.

<u>Def</u> A groupoid is a type A s.t for all $x,y:A$ $x=y$ is a set.

<u>Examples</u>
- Sets are groupoids
- The type Set of sets $\left(\left((X:\mathcal{U}) \times \mathrm{isSet}(X)\right)\right)$ is a groupoid.

Pb We want to be to see any type as e.g a proposition.

→ We assume a universal way to do this

Def For A a type a 'propositional truncation' is a type $\|A\|$ s.t.

$[-] : A \to \|A\|$ $\qquad x : A, \ [x] : \|A\|$

$\|A\|$ is a proposition. a family of proposition

And for $P : \|A\| \to \mathcal{U}$ we the map

$[-]^* : \left((x : \|A\|) \to P(x) \right) \longrightarrow \left((x : A) \to P[x] \right)$ an equivalence

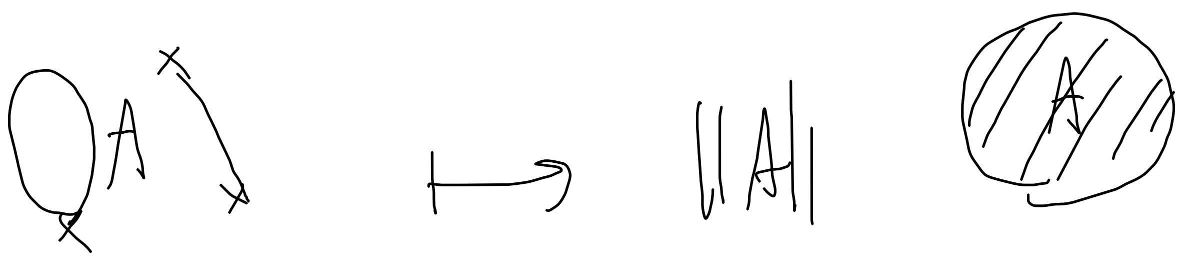
Rq It is equivalent to say that

$[-] : A \to \|A\|$ is initial among

~~por~~ propositions B with $f : A \to B$

We assume such truncation exists.

In practice it means if have $\|A\|$ and we want to prove a proposition, we can assume A.



E.g. Recall $(x,y:A) \to x=y$ means A proposition

then $(x,y:A) \to \|x=y\|$ — A path connected

$$\boxed{\begin{array}{ll} \text{Similarly} & A \to \|A\|_0 \text{ universal with } \|A\|_0 \text{ a set} \\ & A \to \|A\|_1 \text{ ——— } \|A\|_1 \text{ a groupoid} \end{array}}$$

Intuition $\|A\|_0$ is the set of path connected component.

$\|A\|_1$ is the fundamental groupoid of the space A.

A a space its fundamental groupoid

objects : points in A

morphisms : path in A up to homotopy

<u>Lemma</u> A a type, $x, y : A$

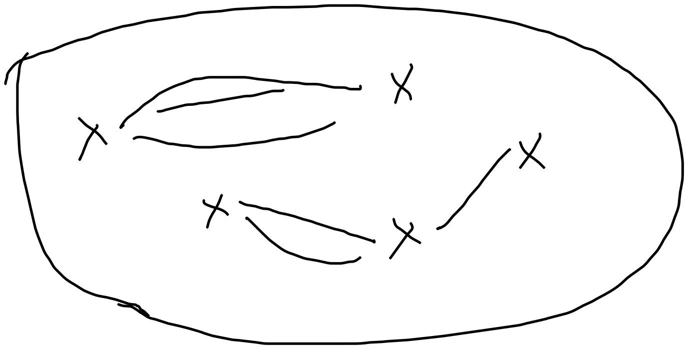
$$[x] =_{\|A\|_0} [y] \simeq \|x =_A y\|$$

$$\left([x] =_{\|A\|} [y] \simeq \#\right)$$

$$[x] =_{\|A\|_1} [y] \simeq \|x =_A y\|_0$$

Part 3

- We give definitions of:
  sets propositions groupoids



- We assumed universal way to build prop/set/groupoids

$\boxed{\text{Rq truncations are supported by the interpretation of types as homotopy types}}$

Higher inductive types

$\|A\| : \mathcal{U}$

$[-] : A \to \|A\|$

trunc : $\mathrm{isProp}(\|A\|)$

Part 4 | Groups & groupoids

Intro

Classically

A groupoid is a category where all morphisms are invertible

A group is ~~a g~~ equivalent to a groupoid with one object.

Our goal

Formulate this in HoTT

Def A group (in HoTT) is a set G
with $\_\times\_ : G \times G \to G$
$\_^{-1} : G \to G$
s.t. ...

Def A pointed type X is $X : \mathcal{U}$ with $* : X$

Def For X a pointed type we define a group:

$$\pi_1(X) :\equiv \| * =_X * \|_0$$

first homotopy group of X.

$$\boxed{\pi_1 : \text{pointed type} \longrightarrow \text{groups} \qquad \text{Can we restrict } \pi_1 \text{ to an equivalence?}}$$

① $\pi_1(\|X\|_1) = \pi_1(X)$ for X pointed type

<u>Proof</u> $\pi_1(\|X\|_1) \equiv \|[*] =_{\|X\|_1} [*]\|_0 \simeq \|\,\|* =_X *\|_0\,\|_0$

$\simeq \|* =_X *\|_0 \equiv \pi_1(X)$

→ we should assume X a groupoid

② If X not path-connected

(then $\pi_1(X)$ forgets everything about X_2)

→ we should assume X connected

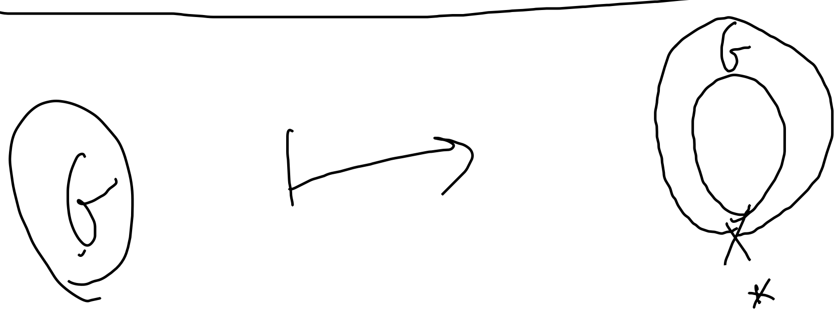
Theorem

π_1 : pointed connected groupoids $\longrightarrow$ Groups

is an equivalence

We call B : Groups $\longrightarrow$ pointed connected groupoids the inverse

$\boxed{\text{Name: } BG \text{ is the classifying space/type for } G.}$



Examples What is $B\pi_1(X)$ for X pointed type?

① X connected, then $B\pi_1(X) \simeq \|X\|_1$

Proof $\pi_1(\|X\|_1) \simeq \pi_1(X) \simeq \pi_1 B\pi_1(X)$

So $\|X\|_1 \simeq B\pi_1(X)$

② X is a groupoid then $B\pi_1(X) \simeq (x:X) \times \|x = *\|$

Proof $\pi_1\big((x:X)\times\|x=*\|\big) = \pi_1(X)$

So $(x:X)\times\|x=*\| = B\pi_1(X)$

What if you remove the "groupoid condition"?

$\boxed{\text{We know groups are sets of the form } x =_X x \text{ for } X \text{ connected groupoid.}}$

Def An **∞-group** is a type X with a pointed connected type Y s.t.

$$X \simeq (* =_Y *)$$

$\boxed{\begin{array}{l}\text{E.g. A topological group gives an } \infty\text{-group.} \\ \quad ⚠ \text{ not all } \infty\text{-groups are of this form}\end{array}}$

Theorem

$\boxed{\text{Aut Aut} = \Omega}$

$\mathrm{Aut} :$ pointed connected type $\longrightarrow$ ∞-group

is an equivalence.

Proof

$\mathrm{Aut} : \left(X : \mathcal{U}, * : X\right) \longmapsto \left(* =_X *, (X, *), \mathrm{id}_{* =_X *}\right)$

$\vdots$

$\left(Y : \mathcal{U}, X : \mathcal{U}, * : X, (* = *) \simeq Y\right)$

$X : \underbrace{\mathcal{U}_*}_{\text{pointed type}} \longrightarrow \left(Y : \mathcal{U}, X : \mathcal{U}_*, * = * = Y\right)$

$\wr\!\wr$

$\left(X : \mathcal{U}_*, Y : \mathcal{U}, (* = *) =_{\mathcal{U}} Y\right)$

Rq I have

Aut : printed type $\longrightarrow$ ∞-Groups

$\boxed{\text{I want some basic concepts of group theory}}$

Def $G, G' : \infty$-Groups

$$\mathrm{Hom}_{\infty\text{Groups}}(G, G') :\equiv \left(f : BG \longrightarrow BG' \right) \times (f * = *)$$

Lemma G, G' are groups this is equivalent to the usual definition.

Def An action of $G : \infty$-Groups on $X : \mathcal{U}$ is a map in

$$\mathrm{Hom}_{\infty\text{Group}}(G, \mathrm{Aut}(\mathcal{U}, X))$$

for $X : \mathcal{U}$.

Eg $\quad \mathrm{Aut}(X,*) :\equiv * =_X *$ as infinity group.

For $X : \mathcal{U}$

$\mathrm{Aut}(\mathcal{U}, X) :\equiv X =_{\mathcal{U}} X$ as ∞-group

$$X \simeq X$$

so this is the usual automorphism group.

$\boxed{\sim \ \pi_1(\mathcal{U}, X) \text{ automorphisms.}}$

<u>Lemma</u> If G is a group and X a set.
This equivalent to usual definition of action of groups.

<u>Eg</u> $B\,\mathrm{Aut}(X)$ for X pointed type
$\simeq (x : X) \times \|x = *\|$

Proposition G : ∞-group

$$\left(\text{Actions of } G\right) \simeq \left(BG \to \mathcal{U}\right)$$

$$P(*) : \mathcal{U} \longleftarrow\!\shortmid \; P : BG \to \mathcal{U}$$

$$\mathrm{Aut}(P) : \underset{\substack{\wr\\ G}}{\mathrm{Aut}\,BG} \to \mathrm{Aut}(\mathcal{U}, P(*))$$

Proof $\quad (X : \mathcal{U}) \times \mathrm{Hom}_{\infty}(G, \mathrm{Aut}(\mathcal{U}, X))$

$$\simeq (X : \mathcal{U}) \times \left(P : BG \to B\mathrm{Aut}(\mathcal{U}, X)\right) \times P(*) = X$$

$$\simeq (X : \mathcal{U}) \times \left(P : BG \to \mathcal{U}\right) \times \left[(x : BG) \to \| P_x = X \|\right] \times P_* = X$$

$X : \mathcal{U}$ and $P_* = X$ implies $(x : BG) \to \| P_x = X \|$

(we know $\| x = * \|$, we want $\| P_x = X \|$,

we can assume $x = *$, then $\| P_* = X \|$ is true)

$$\simeq \; (X : \mathcal{U}) \times (P : BG \to \mathcal{U}) \times (P_* = X)$$

$$\simeq \; BG \to \mathcal{U}$$

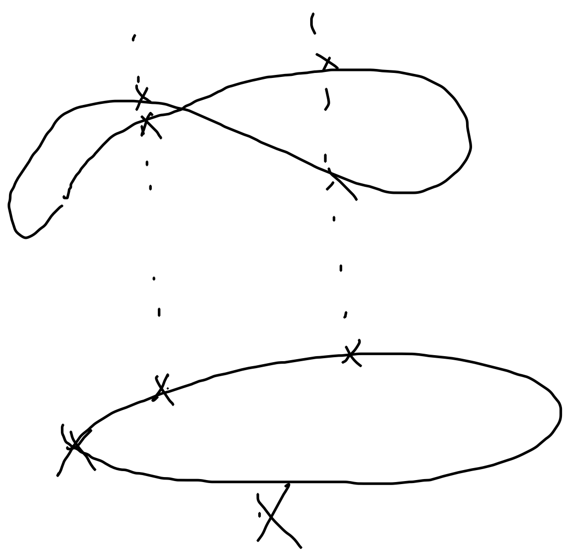
Summary

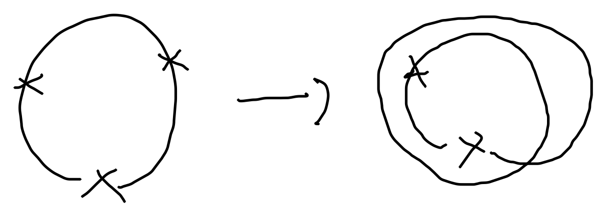
$$\begin{tikzcd} \text{pointed connected types} \arrow[r, bend left, "\mathrm{Aut}"] & \infty\text{-Groups} \arrow[l, bend left, "B"] \end{tikzcd} \qquad \begin{tikzcd} \text{pointed connected groupoids} \arrow[r, bend left, "\pi_1"] & \text{Groups} \arrow[l, bend left, "B"] \end{tikzcd}$$

We defined morphisms and actions

Part 5 Applications

Def a covering of X a type is a map to X whose fibers are sets





Theorem X is path connected

$$\left(\text{Coverings of } X\right) \simeq \left(\pi_1(X)\text{-actions}\right)$$

Proof $\mathrm{Cover}(X) \simeq X \to \mathrm{Set}$ (Grothendieck correspondence)

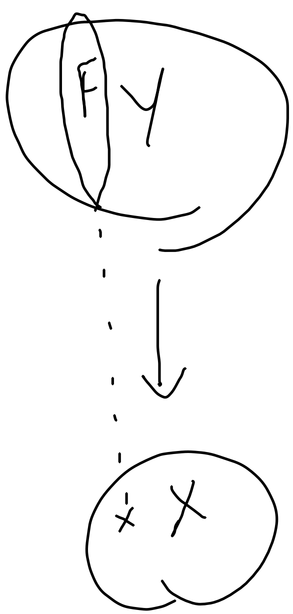
$\simeq \|X\|_1 \to \mathrm{Set}$ (Set is a groupoid)

$\simeq B\pi_1 X \to \mathrm{Set}$ (X connected)

$\simeq \pi_1(X)\text{-action}$

<u>Definition</u> $X, F : \mathcal{U}$, an F-bundle over X is an inhabitant of

$$(Y : \mathcal{U}) \times (f : Y \to X) \times (x : X) \to \| \mathrm{fib}_f(x) = F \|$$



$(R_y \quad \mathrm{fib}_f(x) = F)$

then $f = \pi : Y \times X \to X$

Theorem

$$\left(F\text{-bundle over } X \right) \simeq \left(X \to B\mathrm{Aut}(U,F) \right)$$

(recall Grothendieck correspondence

$$\left(\text{maps over } X \right) \simeq \left(X \to U \right) \quad)$$

Proof:

$$(Y:U) \times (f : Y \to X) \times \| \mathrm{fib}_f(x) = F \|$$

$$\simeq (P : X \to U) \times (x:X) \to \| Px = F \|$$

$$\simeq X \to (Y:U) \times \| Y = F \| \simeq X \to B\mathrm{Aut}(U,F)$$

In the notes section Applications, two often theorems like this:

- We classify
  - G-principal bundle over X.
  - Extensions of ∞-groups.

We

Synthetic Homotopy Theory works very well here.

- $\mathcal{U}$ classifies all maps.
- truncations works well. ($\|X\|_2$ ...)
- ∞-Groups have the 'easy' definition in HoTT

Usually ∞-groups involves infinite towers of equations

I left out:

- Higher Inductive types
  - → inductive types with constructors in identity types.
  - → e.g.
    - truncations,
    - BG classifying space,
    - CW complex (spheres, torus, Klein bottle)
    - homotopy pushouts.
    - ...
- To compute identity types, you need 'encode-decode' method

$$[x] =_{\|X\|_0} [y] \simeq \|x =_X y\|$$

Friday at 14 at EPIT

Egbert Rijke Lecture on Synthetic HoTT