last time $\prod$ $\sum$ $\mathrm{Id}_A(a,b)$ notation: $a =_A b$

dep prod, dep sum, identity/equality/paths

usual "A iff B" i.e. "$f : A \to B$" "$g : B \to A$" ($\exists$)

but there is a stronger notion: $A \simeq B$ "iso of iso"

add conditions: $f \cdot g = id$, $g \cdot f = id$ ← equivalence of types

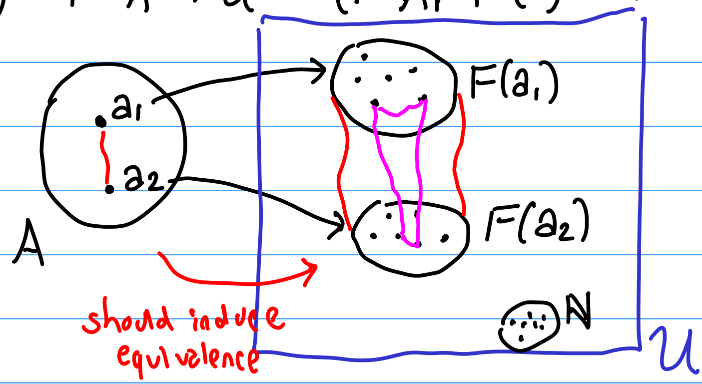
Univalence axiom: $(A \simeq_{\mathcal{U}} B) \simeq (A =_{\mathcal{U}} B)$

What is a family of types?

1) $F : A \to \mathcal{U}$ $\quad (x : A \vdash F(x) : \mathcal{U})$

isomorphism of categories rarely useful — equivalence is useful.

family



should induce equivalence

category theory

Jeffrey set theoretic view of things

2) $\sum x : A . F(x)$ total space

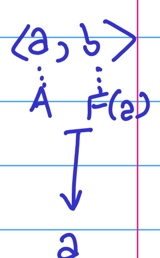
the fiber of F over a

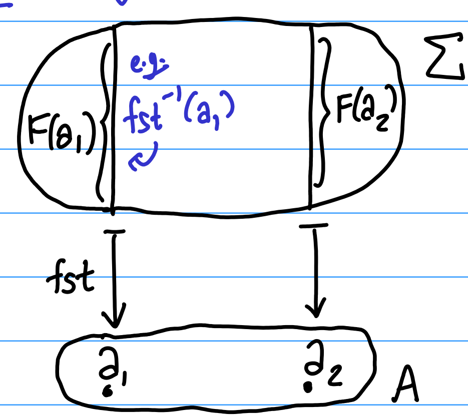
(drawn out into a line)

fibration

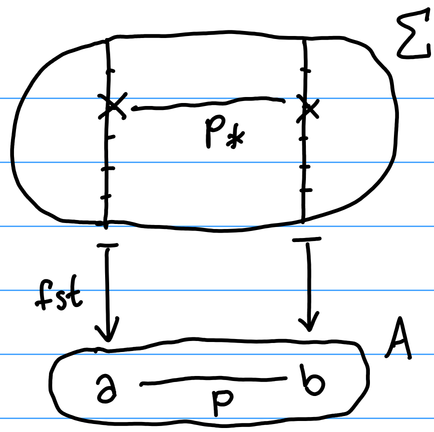
display map

$A \xleftarrow{\text{fst}} \sum x : A . F(x)$





"the views are dual; the arrows have been reversed"



path lifting property

Identity Type

0) $$\boxed{\dfrac{a : A}{\mathrm{refl}_A(a) : \mathrm{Id}_A(a,a) \quad a =_A a}} \;=\text{I-refl}$$

everyone has this

1) extensional identity

stresses identity as mere proposition

$$\frac{p : a =_A b}{a \equiv b : A} \;(=\text{-E-ext})$$

e.g. $\lambda x. x \equiv \lambda x. 0 + x : \mathbb{N} \to \mathbb{N}$

2) intensional identity (stress on Id_A as data)

stresses identity as data (type)

want to say: only element in it is refl: an induction principle

$$\frac{p : a =_A b \qquad x{:}A, y{:}A, z{:}\, x =_A y \vdash P(x,y,z) \qquad x : A \vdash q : P(x,x,\mathrm{refl})}{J[x,y,z.P](x.q\,;p) : P(a,b,p)}$$

motive

$\mathrm{rec}^{\mathrm{Id}}$

not a judgment

note: notation is confusing

path induction $\quad J(x.q;\ \mathrm{refl}_A(a)) \equiv [a/x]q : P(a,a,\mathrm{refl}_A(a))$

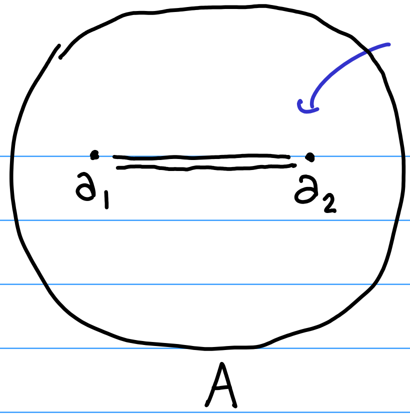
case-analysis on one case

Fact: now can show $=_A$ is an equivalence relation

1) $=_A$ is symmetric
2) $\Pi x,y : A.\ x =_A y \to y =_A x$
3) $x : A,\ y : A,\ z : x =_A y \vdash \mathrm{sym}(z) : y =_A x$

aka z^{-1}

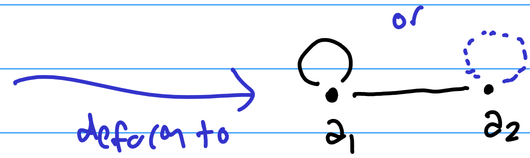
s.t. $\quad \mathrm{sym}(\mathrm{refl}_A(a)) \equiv \mathrm{refl}_A(a)$



equates (also drawn squiggly)

note: abstracted from geometric considerations (synthetic)

deform to



or ← self-loop

notation: loops don't actually go anywhere

Informally, it suffices to assume that $z = \mathrm{refl}_A(x)$

$$x : A \vdash \mathrm{refl}_A(x) : x =_A x$$

thus $\mathrm{sym}(z) := J_{x,y,\, x=y}(x.\, \mathrm{refl}_A(x);\, z)$

$$\mathrm{sym}(\mathrm{refl}_A) \equiv J_{x,y,\, x=y}(x.\, \mathrm{refl}_A(x);\, \mathrm{refl}_A(a)) \equiv \mathrm{refl}_A(a) \ \checkmark$$

ETT ~ homotopy theory of sets

the lesson of HoTT is that types are not sets (they're weak ∞-groupoids)

Prop 1) $=_A$ is transitive

2) $\prod x,y,z : A.\ x =_A y \to y =_A z \to x =_A z$

3) $x:A, y:A, z:A, u : x =_A y, v : y =_A z \vdash \mathrm{trans}(u,v) : x =_A z$) — written $u \cdot v$

s.t. $\mathrm{trans}(\mathrm{refl}(a), \mathrm{refl}(a)) \equiv \mathrm{refl}(a)$

[Trick: to show $x \leq y$ iff $\forall z$ if $z \leq x$ then $z \leq y$]

↑ naturality

Yoneda lemma on preorders

Makes proofs easier!

STS: $x:A, y:A, u : x =_A y \vdash g_u : \prod z{:}A.\ y =_A z \to x =_A z$

$$\mathrm{trans}(u,v) := g_u(z)(v) : x =_A z$$

$$g_{\mathrm{refl}}(\cancel{\mathrm{refl}}) \equiv \cancel{\mathrm{refl}}$$

"even if it's wrong, I fooled you"

Note: this is not the double induction proof

$$g_u := J_{x,y.\,(\prod z:A.\ y=z \to x=z)}(w.\, \lambda z. \lambda r : u = z.\, r;\, u)$$

says laws

pre-groupoid

$\mathrm{refl}(a)$ (unit)

p^{-1} (inverse)

$p \cdot q$ (mult)

not a group, since the types are not the same (it is a groupoid)

groupoid

$\mathrm{refl}^{-1} = \mathrm{refl}$

$\mathrm{refl} \cdot q = q$

$p \cdot \mathrm{refl} = p$

$p \cdot (q \cdot r) = (p \cdot q) \cdot r$

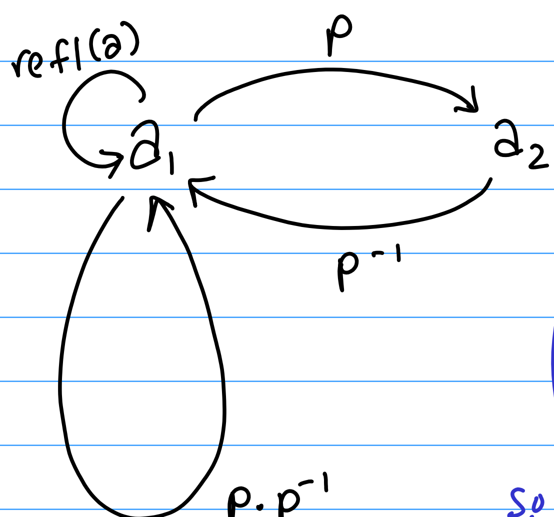
$p \cdot p^{-1} = \mathrm{refl}$

$p^{-1} \cdot p = \mathrm{refl}$

equality?

what about p^{-1-1}?

these all hold



$a : A \quad b : A$

$p : a =_A b \quad p^{-1} : b =_A a$

$\mathrm{refl}_A(a) : a =_A a$

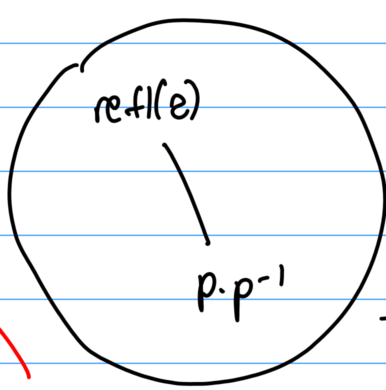
$p \cdot p^{-1} : a =_A a$

this is a type!

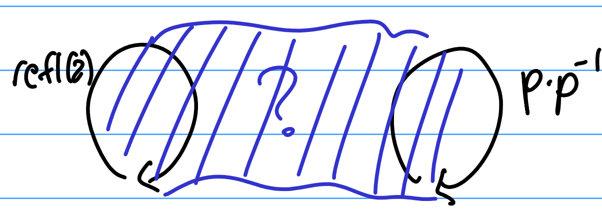
$\Omega_A(a)$

loop space

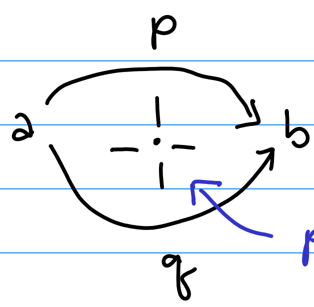
So we can talk about their equality



$\Omega_A(a)$



$$\begin{tikzcd} a \arrow[r, bend left=50, "p"{name=P}] \arrow[r, bend right=50, "q"'{name=Q}] & b \arrow[from=P, to=Q, phantom, "\Downarrow\alpha"] \end{tikzcd}$$



punctured, cannot manage

groupoid up to higher homotopy

What are these proofs? α requires inverses, concatenations, etc in the loop space.

weak ∞-groupoid

"Nothing really holds, it only holds up to a bigger lie. Well, as long as you carry these lies to infinity you're fine; it's a Ponzi scheme that runs out. 'Well, what could go wrong?'"

It's not well-founded. There's no spot where something utterly becomes true... unless you truncate. If I demand that these hold definitionally (the groupoid laws), you get a strict groupoid, at any dimension you wish. For example, the fundamental group of a loop space is the zero-truncation of a loop space. That truncation is the starting point of algebraic topology.