Introduction to Homotopy type theory

Andrej Bauer

EPIT 2020 1

Six short lectures with exercises:

1. Martin-Löf type theory
2. Identity types
3. Homotopy levels
4. Equivalences
5. Higher-inductive types
6. Univalence

Backround reading:

- Egbert Rijke's textbook - Chapter I (see #resources on Discord)
- HoTT book - Chapter 1
- Video lectures (see EPIT GitHub repo)

Part 1: Martin-Löf Type Theory

- Foundational
  - It does not rely on logic, set theory or category theory
  - Mathematics can be carried out in it
- A theory of constructions
  - How mathematical objects are built & manipulated
  - Not about what is the case (that would be logic)
  - "Construction" can mean many things
  - We take the geometric view:
    construction of points & paths in spaces

Basic concepts

Type / Space	A type
Element / Point	$t : A$

Type & point may depend on other points

$$\underbrace{x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n}_{\text{context } \Gamma} \vdash B(x_1, \ldots, x_n) \text{ type}$$

$x_i : A_i$ parameter of type A_i

Examples from real life:

1. $\mathbb{R}^n$ depends on $n : \mathbb{N}$ — $n : \mathbb{N} \vdash \mathbb{R}^n$ type
2. $[a,b]$ depends on $a, b \in \mathbb{R}$ and $a \leq b$ — $a : \mathbb{R}, b : \mathbb{R}, p : (a \leq b) \vdash [a,b]$ type

Equality	$A \equiv B$	(judgementally) equal types
	$s \equiv_A t$	(judgementally) equal points

Type theory as a formal system

Judgement forms

$$\Gamma \vdash A \text{ type}$$
$$\Gamma \vdash t : A$$
$$\Gamma \vdash A \equiv B$$
$$\Gamma \vdash s \equiv_A t$$

Rules of inference

$$\frac{P_1 \quad P_2 \quad \cdots \quad P_n}{C}$$

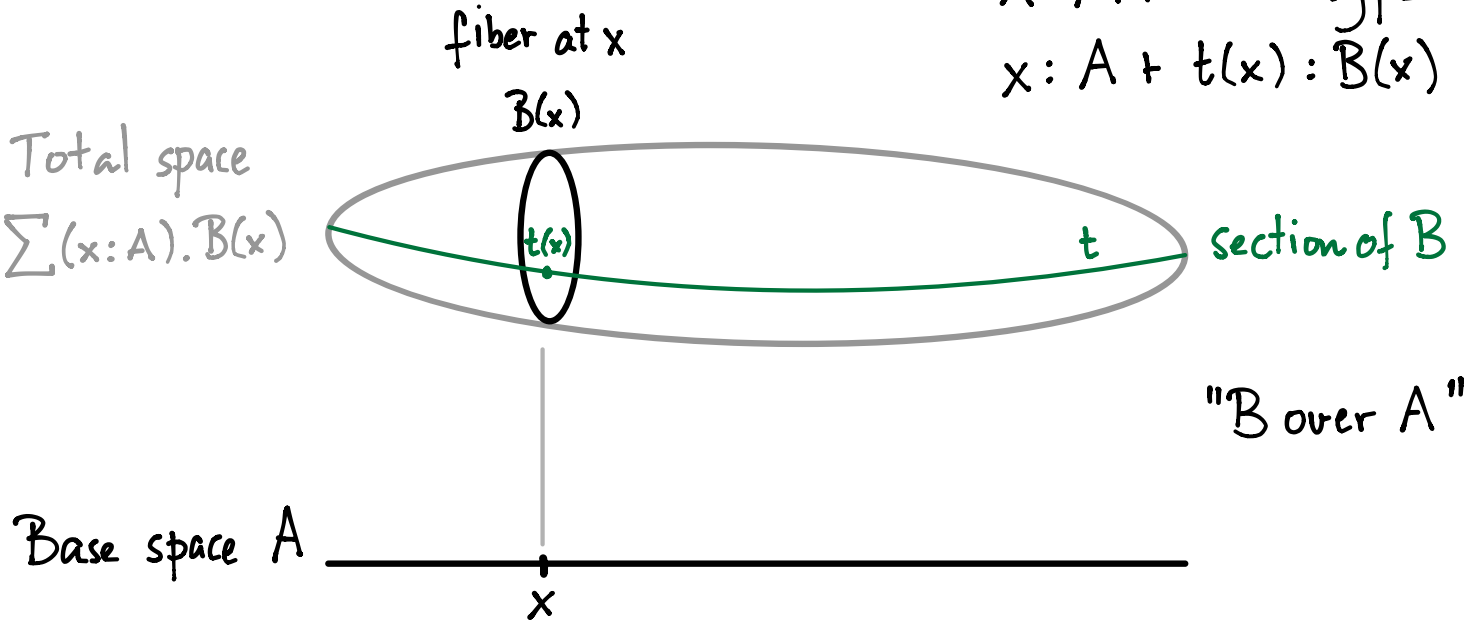
premises

conclusion

The geometric picture

$x : A \vdash B(x)$ type

$x : A \vdash t(x) : B(x)$



Basic constructions

Dependent sum $\Sigma(x:A).B(x)$

- elements : ordered pairs (s,t) where $s:A$, $t:B(s)$
- operations : projections π_1 & π_2
- equations $\pi_1(s,t) \equiv_A s$

  $\pi_2(s,t) \equiv_{B(s)} t$

  $(\pi_1 u, \pi_2 u) \equiv_{\Sigma(x:A).B(x)} u$

Binary product $A \times B := \Sigma(x:A).B$

(B does not depend on A)

Dependent product $\Pi(x:A).B(x)$

- elements: functions $\lambda x:A.e$ "x maps to e"

  $x \mapsto e$

- operation: application

$$\frac{\Gamma \vdash s : \Pi(x:A).B(x) \qquad \Gamma \vdash t : A}{\Gamma \vdash s\,t : B(t)}$$

- equations:

$$(\lambda x:A.e)\,t \equiv_{B(t)} e[x:=t]$$

(substitution)

$$\lambda x:A.\,s\,x \equiv_{\Pi(x:A).Bx} s$$

"η-rule" ↕ inter-derivable

$$\frac{\Gamma, x:A \vdash u(x) \equiv_{B(x)} v(x)}{\Gamma \vdash (\lambda x:A.u(x)) \equiv_{\Pi(x:A).B(x)} (\lambda x:A.v(x))}$$

extensionality rule

Function space $A \to B := \Pi(x:A).B$

(B does not depend on A)

Natural numbers $\mathbb{N}$

- elements: $0 : \mathbb{N}$ and $\dfrac{n : \mathbb{N}}{S\,n : \mathbb{N}}$
- induction principle:

  Given a type P over $\mathbb{N}$

  $$ind_{\mathbb{N}} : P\,0 \to (\Pi(k:\mathbb{N}).\,P\,k \to P(S\,k)) \to \Pi(n:\mathbb{N}).\,P\,n$$

- equations

  $$ind_{\mathbb{N}}\ s\ t\ 0 \equiv_{P\,0} s$$

  $$ind_{\mathbb{N}}\ s\ t\ (S\,n) \equiv_{P(S\,n)} t\ n\ (ind_{\mathbb{N}}\ s\ t\ n)$$

Finite types

$\mathbb{0}$ empty type:

$$\frac{\vdash t : \mathbb{0} \qquad \vdash A \text{ type}}{\vdash \text{exfalso}\, t : A}$$

$\mathbb{1}$ unit type: element $() : \mathbb{1}$

equation $s \equiv_{\mathbb{1}} t$

$\mathbb{2}$ booleans: $\text{false}, \text{true} : \mathbb{2}$

Construction by cases: given type P over $\mathbb{2}$

$$\frac{\vdash b : \mathbb{2} \qquad \vdash s : P(\text{true}) \qquad \vdash t : P(\text{false})}{\vdash \text{case}(b, s, t) : P(b)}$$

$\text{case}(\text{true}, s, t) \equiv_{P(\text{true})} s$

$\text{case}(\text{false}, s, t) \equiv_{P(\text{false})} t$

Universe $\mathcal{U}$ – a type of types

- elements are types (not all of them)
- closed under Σ, Π and identity types

We can have many universes: $\mathcal{U}_0, \mathcal{U}_1, \mathcal{U}_2, \ldots$

Universes may be <u>cummulative</u>: if $A : \mathcal{U}_n$ then $A : \mathcal{U}_{n+1}$

Observation: A dependent type B over A is a map

$$B : A \longrightarrow \mathcal{U}$$

Exercises

① Construct a point of

$$(\prod(x:A).\sum(y:B).C\,x\,y) \to \sum(f:A\to B).\prod(x:A).C\,x\,(f\,x)$$

where A and B are types and $C : A \to B \to \mathcal{U}$.

② Use $\mathrm{ind}_{\mathbb{N}}$ to define $\mathrm{double} : \mathbb{N} \to \mathbb{N}$
which doubles a number, e.g.: $\mathrm{double}\,(S(S0)) \equiv S(S(S(S0)))$.

③ Define $\mathrm{halve} : \mathbb{N} \to \mathbb{N}$
which halves a number (rounded down).

Part 2: Identity types

Given a type A and points $s, t : A$,
we form the type

$$Id_A(s,t) \qquad \text{the identity type (path type)}$$

We think of $p : Id_A(s,t)$ as a path from s to t.
We shall also use the notation

$$s =_A t \quad \text{or just} \quad s = t$$

(not to be confused with $s \equiv_A t$)

For any $t : A$ there is

$$\text{idpath}_A t : t =_A t \qquad \text{also:} \quad \text{refl}_A t : t =_A t$$

Next we would like to have

Homotopy invariance

If there is a path $p : s =_A t$ then any construction that uses s can be transformed along p to use t instead.

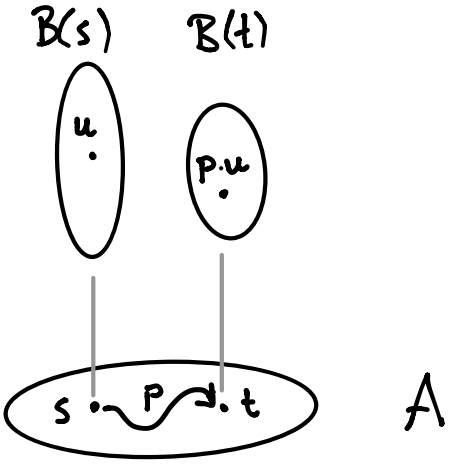
First attempt:

Given $B : A \to \mathcal{U}$

$s, t : A$

$p : s =_A t$

$u : B(s)$



we form the transport

$$p.u : B(t)$$

Problem: B may depend on x, y and p, not just x.

Solution: path induction

Suppose we would like to construct a point of

$$\prod(x:A)(y:A)(p:x =_A y).\ B(x,y,p)$$

Then it suffices to construct a point of

$$d:\prod(z:A).\ B(z,z,\mathrm{idpath}\ z).$$

We write

$$J(d,x,y,p):B(x,y,p)$$

for the point so constructed.

Equality:

$$J(d,s,s,\mathrm{idpath}\ s)\equiv d(s).$$

Example: path concatenation

Given $p : x =_A y$ and $q : y =_A z$, construct

$$p \cdot q : x =_A z$$

$$\_ \cdot \_ : \prod(x:A)(y:A)(p:x=y) \underbrace{\prod(z:A)(q:y=z).\ x=z}_{B(x,y,p)}$$

We apply path induction:

$$\prod(x:A)\, B(x,x,\mathrm{idpath}\, x)$$

$$\prod(x:A)\prod(z:A)(q:x=z).\ x=z$$

$$\lambda x\, z\, q.\ q$$

Answer:

$$p \cdot q := J((\lambda x\, z\, q.\ q), x, y, p)$$

Equation:

$$\mathrm{idpath}_x \cdot q \equiv J((\lambda x\, z\, q.\ q), x, x, \mathrm{idpath}\, x) \equiv q$$

Transport:

$$\prod(x,y:A)(p:x =_A y).\ B(x) \to B(y)$$

Path induction:

$$\prod(z:A).\ B(z) \to B(z)$$

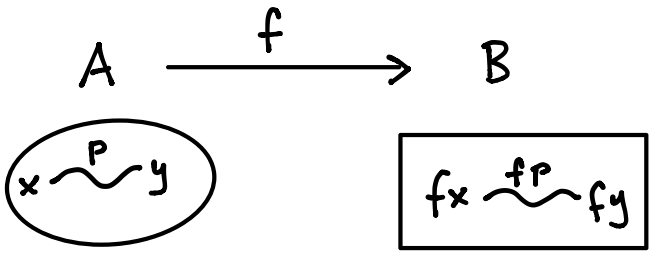
$$\lambda\ z\ \ u\ .\ u$$

Transport: $p.u := J(\lambda z\, u.u, x, y, p)$

$$\mathrm{idpath}_x.u \equiv u$$

Action on paths

Given $f : A \to B$



$$f(-) : \prod(x,y:A)(p : x =_A y).\ fx =_B fy$$

By path induction:

$$\prod(z:A).\ fz =_B fz$$

$$\lambda z.\ \mathrm{idpath}_{fz}$$

Answer: $f(p) = J(\lambda z.\ \mathrm{idpath}_{fz}, x, y, p)$

$$f(\mathrm{refl}_x) = \mathrm{refl}_{fx}$$

Exercises:

① Construct a path between paths:

$$p \cdot \mathrm{idpath}_y =_{x =_A y} p$$

where $x, y : A$ and $p : x =_A y$

② Given a path $p : x =_A y$ construct $p^{-1} : y =_A x$ and show that $\mathrm{idpath}_x^{-1} = \mathrm{idpath}_x$.

③ Suppose

$$u, v : \sum (x:A).\, B(x)$$

$$p : \pi_1 u =_A \pi_2 v$$

$$q : p_*(\pi_1 u) =_{B(\pi_1 v)} v_2$$

are given. Construct a path $u =_{\Sigma(x:A).B(x)} v$

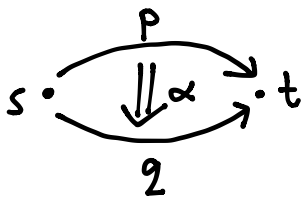
Part 3: Homotopy levels

Higher-dimensional paths:

$t : A$ point

$p : s =_A t$ path

$\alpha : p =_{s=_A t} q$ 2-path



(and so on)

Sometimes the higher-dimensional paths collapse above dimension n.

We start with the collapse of points:

Definition: A type A is **contractible** when

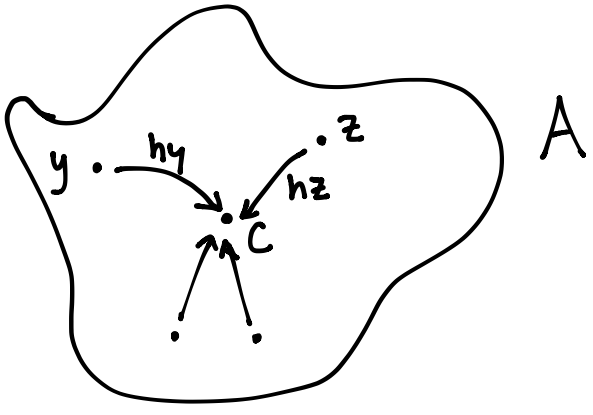
$$\mathsf{isContr}\ A :\equiv \sum(x:A).\prod(y:A).\ x =_A y$$

has a point.

When $(c,h) : \mathsf{isContr}\ A$:

- $c : A$ is the **center of contraction**
- $h : \prod(y:A).\ c =_A y$ is a homotopy from const_c to id_A

(observe that $h : \prod(x:A).\ \mathsf{const}_c\, x =_A \mathsf{id}_A\, x$)



Example: $\mathbb{1}$ is contractible

center: $()$

homotopy: $\prod(x:\mathbb{1}).\ () =_{\mathbb{1}} x$ is given by $\lambda x:\mathbb{1}.\ \mathrm{idpath}_{()}$

(We used the fact that $\mathrm{idpath}_{()} : () =_{\mathbb{1}} x$ because $() \equiv_{\mathbb{1}} x$.)

We now generalize to all levels.

Definition

For $n \geq -2$ define the notion of **n-type**:

(-2)-type $A :\equiv$ isContr A

$(n+1)$-type $A :\equiv \prod(x,y:A).\ n\text{-type}\,(x =_A y)$

Fact: n-type $A \to (n+1)$-type A

Special cases of interest:

→ (-1)-types are called **propositions**:

- equivalent formulation:

  $$\mathrm{isProp}\ A := \prod(x,y:A).\ x =_A y$$

- any two points are equal (connected by a path)
- all higher identity types are contractible
- examples: $\mathbb{0}$, $\mathbb{1}$

→ 0-types are called **sets**:

- equivalent formulation:

  $$\mathrm{isSet}\ A := \prod(x,y:A)\prod(p,q:x=_A y).p=q$$

- parallel paths are homotopic
- examples: $\mathbb{0}$, $\mathbb{1}$, $\mathbb{2}$, $\mathbb{N}$

→ 1-types are called **groupoids**

- no interesting 2-paths

Property vs. structure

We say that $P : A \to \mathcal{U}$ is a property on A when

$$\prod(x:A).\ \mathrm{isProp}(P_x).$$

(In general, P is a structure on A.)

Example:

- The structure of a monoid on a set M:

$$\mathrm{Monoid}\ S :=$$

$$\sum(m : S \times S \to S)(e : M).$$

$$\left(\prod(x,y,z:S).\ m(m(x,y),z) = m(x,m(y,z))\right) \times$$

$$\left(\prod(x:S).\ (m(x,e) = x) \times (m(e,x) = x)\right)$$

- The property of a monoid being a group:

$$\mathrm{isGroup}(S, m, e, \alpha, \beta) :=$$

$$\sum(i : S \to S).\ \prod(x:S).\ (m(x,ix) = e) \times (m(ix,x) = e)$$

- The structure of a group on a set S

$$\mathrm{Group}\ S :\equiv \sum(M : \mathrm{Monoid}\ S).\ \mathrm{isGroup}(M)$$

Truncation:

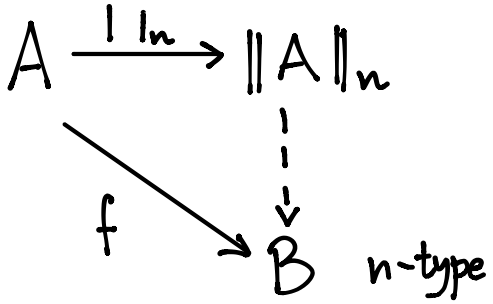
An operation which turns a type A into an n-type.

$$\frac{A \text{ type}}{\|A\|_n \text{ type}} \qquad \frac{t : A}{|t|_n : \|A\|_n}$$

$$\frac{A \text{ type}}{n\text{-trunc } A : n\text{-type } \|A\|_n}$$

Universal property of $\|A\|_n$:

every map $f : A \to B$ into an n-type B factors uniquely through $\|A\|_n$

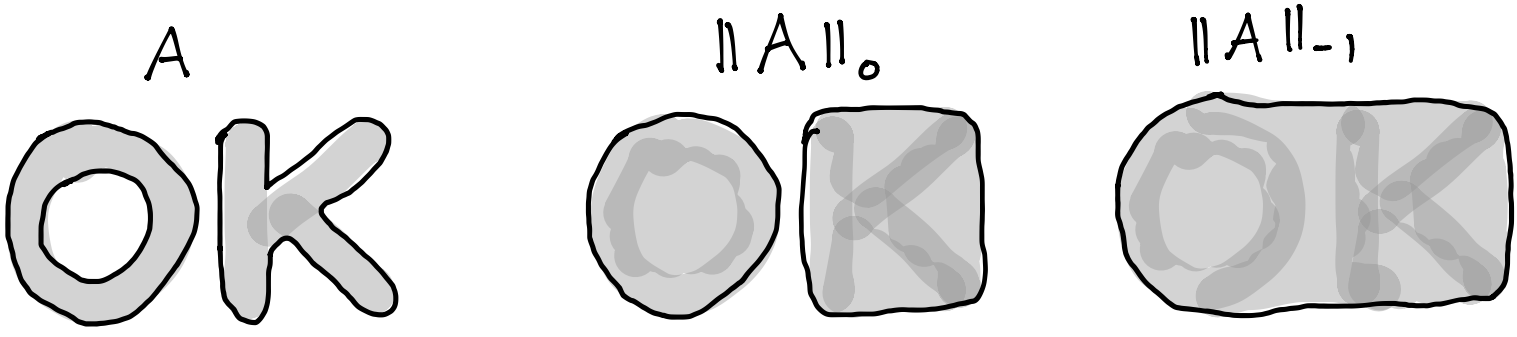


How do we express the above in type theory?

Special cases of interest

Set truncation $\|A\|_0$ the space of connected components of A

A is connected when $\|A\|_0$ is a proposition



Propositional truncation $\|A\|_{-1}$

We use it to define logic

$$\top := \mathbb{1}$$

$$\bot := \mathbb{0}$$

$$\neg P := P \to \bot$$

$$P \wedge Q := P \times Q$$

$$P \Rightarrow Q := P \to Q$$

$$P \vee Q := \|P + Q\|_{-1} \quad \text{(where } A+B \text{ is disjoint sum)}$$

$$\forall x:A.P(x) := \prod(x:A).P(x)$$

$$\exists x:A.P(x) := \|\sum(x:A).P(x)\|_{-1}$$

These satisfy the rules of intuitionistic logic.

Caveat:

In order to show that $\to$ and $\prod$ preserve propositions we need function extensionality.

Importance of truncation for mathematics

- Surjection vs. retraction: $f : A \to B$

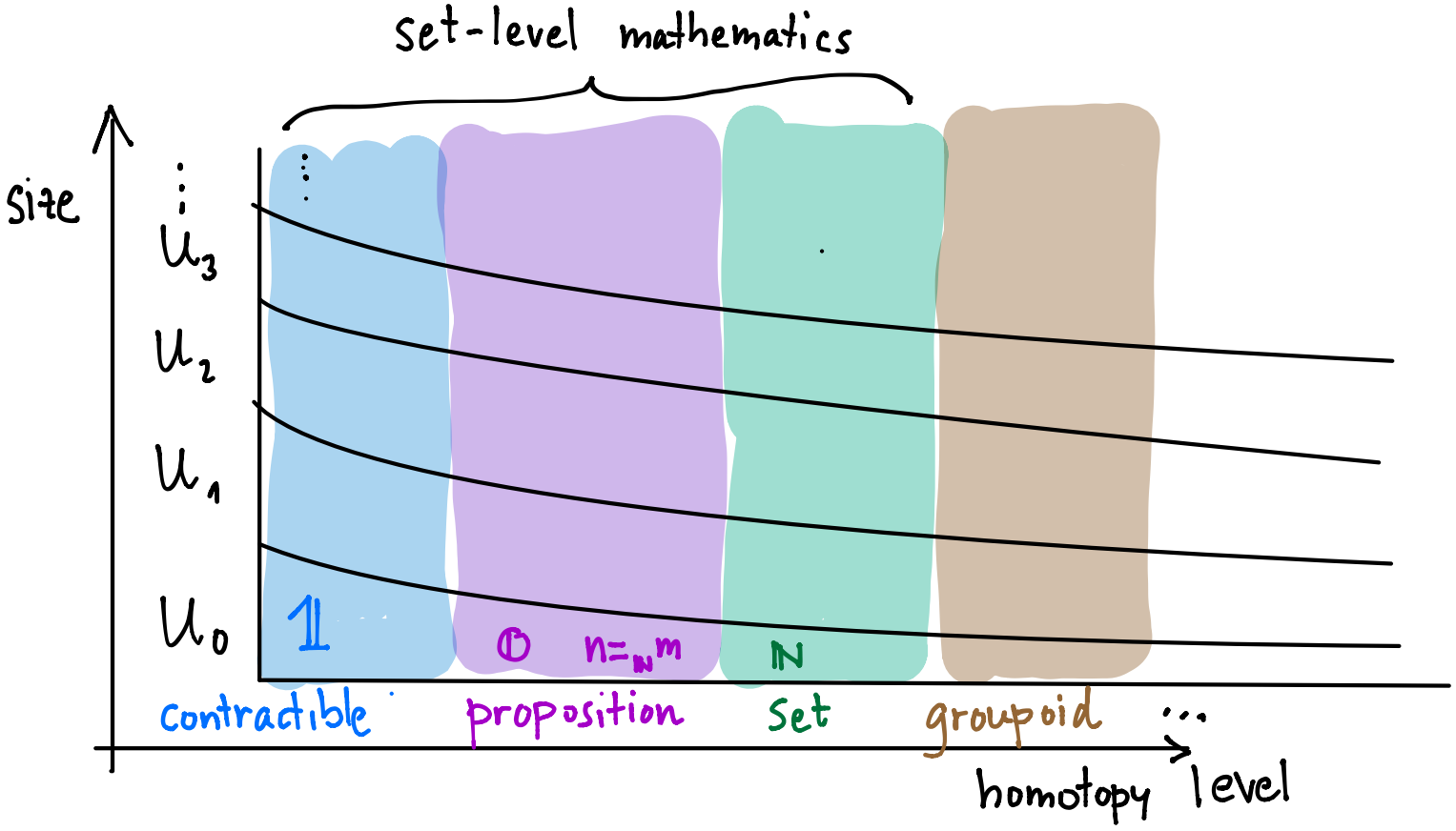
  $\prod(y:B).\ \sum(x:A).\ fx =_B y$ "f has a right inverse"

  $\prod(y:B).\ \| \sum(x:A).\ fx =_B y \|$ "f is surjective"

- Loop space vs. fundamental group

  $\Omega(A,a) := (a =_A a)$ loop space at base point $a:A$

  $\pi_1(A,a) := \| \Omega(a,a) \|_0$ fundamental group



Exercises:

① $\mathsf{isContr}\ A \to \Pi(x,y:A).\ \mathsf{isContr}\ (x =_A y)$

② Is contractibility a property or a structure?

③ $\Pi(x:A).\ \mathsf{isContr}(\Sigma(y:A).\ x =_A y)$

Part 4: Equivalences

Function extensionality

Given maps $f, g : \Pi(x:A).\, B(x)$ define

$$f \sim g \;:=\; \Pi(x:A).\, fx =_{Bx} gx$$

We have:

$$f =_{\Pi(x:A).Bx} g \to f \sim g \qquad \text{(exercise)}$$

How about the other way?

Function extensionality:

$$f \sim g \to f =_{\Pi(x:A).Bx} g$$

It turns out that univalence implies function extensionality.

Given a map $f : A \to B$, how do we say that it is an equivalence, or an isomorphism?

(1) "f has a two-sided inverse"

$$\mathsf{isIso}\, f := \sum (g : B \to A).\, (f \circ g \sim \mathrm{id}_B) \times (g \circ f \sim \mathrm{id}_A)$$

(2) "Every $y : B$ has exactly one preimage"

$$\mathsf{isEquiv}\, f := \prod (y : B).\ \mathsf{isContr}\left(\sum (x : A).\ f x =_B y\right)$$

(3) "f has a left- and right-sided inverses"

$$\mathsf{isBiinv}\, f := \left(\sum (g : B \to A).\, f \circ g \sim \mathrm{id}_B\right) \times \left(\sum (h : B \to A).\ h \circ f \sim \mathrm{id}_A\right)$$

Observations:

- isEquiv is a property
- isIso is a structure in general
- isBiinv is a property & equivalent to isEquiv

Define:

$$A \cong B := \sum (f : A \to B).\, \mathsf{isIso}\, f$$

$$A \simeq B := \sum (f : A \to B).\, \mathsf{isEquiv}\, f$$

We prefer $\simeq$ because isEquiv is a property.

There is a map $A \cong B \to A \simeq B$
(every iso can be rectified to an equivalence)

Equivalence is equivalence:

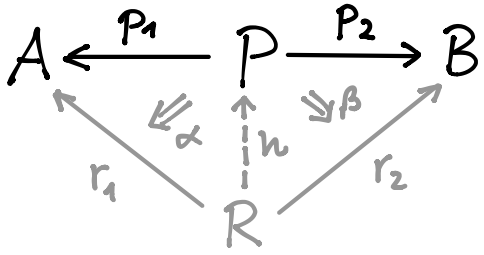
$$A \simeq A$$

$$A \simeq B \to B \simeq A$$

$$A \simeq B \to B \simeq C \to A \simeq C$$

Equivalences & universal properties

Recall the notion of a product diagram



How to express its universal property?

$$e_R : (R \to P) \to \sum (r_1 : R \to A)(r_2 : R \to A).(p_1 \circ h = r_1) \times (p_2 \circ h = r_2)$$

$$e_R(h) := (p_1 \circ h,\ p_2 \circ h,\ \mathit{idpath},\ \mathit{idpath})$$

$$\mathit{isEquiv}(e_R) = \prod (r_1, r_2, \alpha, \beta).\ \mathit{isContr}\left(\sum (h : R \to P).\ e_R(h) = (r_1, r_2, \alpha, \beta)\right)$$

The inverse map e_R^{-1} computes h from $(r_1, r_2, \alpha, \beta)$.
The rest provides uniqueness of h.

Exercises:

① (a) If P and Q are propositions then

$$(P \to Q) \times (Q \to P) \to P \simeq Q$$

(b) If P and Q are propositions then the map you defined in (a) is an equivalence.

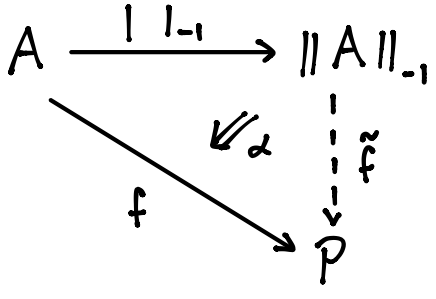
(c) Homework: If X and Y are sets then

$$(X \simeq Y) \simeq \sum (f : X \to Y) . \text{Iso } f$$

② Use equivalences to express the fact that

$$|\ |_{-1} : A \longrightarrow \|A\|_{-1}$$

is the universal map from A to propositions:



For any proposition P and $f : A \to P$ there is a unique $\tilde{f} : \|A\|_{-1} \to P$ such that $\tilde{f} \circ |\ |_{-1} = f$.

Part 5: Univalence

Function extensionality:

$$(f =_{A \to B} g) \simeq \prod(x:A).\, fx =_B gx$$

Equality on $A \times B$:

$$(u =_{A \times B} v) \simeq (\pi_1 u =_A \pi_1 v) \times (\pi_2 u =_B \pi_2 v)$$

These are <u>extensionality principles</u>:

"Two objects are equal when their parts (extensions) are equal."

Do we have such a principle for universes?

There is a map

$$\mathrm{idtoequiv} : \prod(A,B:\mathcal{U}).\; A =_{\mathcal{U}} B \to A \simeq A$$

Defined by path induction

$$\mathrm{idtoequiv}\ C\ c\ \mathrm{idpath}_c := (\mathrm{id}_c, \ldots..)$$

<u>Definition</u>:

A universe $\mathcal{U}$ is <u>univalent</u> if idtoequiv is an equivalence.

$$(A =_{\mathcal{U}} B) \simeq (A \simeq B)$$

"Identity is equivalent to equivalence."

Axiom of univalence: Every universe is univalent.

In practice we use the inverse of idtoequiv:

$$ua : (A \simeq B) \longrightarrow A =_{\mathcal{U}} B$$

Given an equivalence $e : A \simeq B$ we get $ua(e) : A =_{\mathcal{U}} B$, and we know

$$ua(e).x =_B e(x)$$

Some consequences of univalence:

- function extensionality
- it allows us to compute homotopy groups
- it implies that

$$\mathrm{Set}_{\mathcal{U}} := \sum(A : \mathcal{U}).\ \mathrm{isset}\ A$$

  is a groupoid (and not a set).

- it implies that isomorphic groups/rings/modules are equal ("Structure identity principle")
- many other mathematically desirable facts

Exercises

(1) Show that the type of true propositions

$$\sum (A : \mathcal{U}).\ \mathsf{isProp}\ A \times A$$

is contractible.

(2) Show that $\sum (A : \mathcal{U}).\ \mathsf{isSet}\ A$ is not a set.
Hint: $(2 \simeq 2) \simeq 2$.