

# Introduction to Homotopy type theory

Andrej Bauer

EPIT 2020<sup>1</sup>

Six short lectures with exercises:

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

## Background 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

$x_1 : A_1, x_2 : A_2, \dots, x_n : A_n + B(x_1, \dots, x_n)$  type

$\underbrace{\qquad\qquad\qquad}_{\text{Context } \Gamma}$

$x_i : A_i$  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

$\Gamma \vdash A \text{ type}$

$\Gamma \vdash t : A$

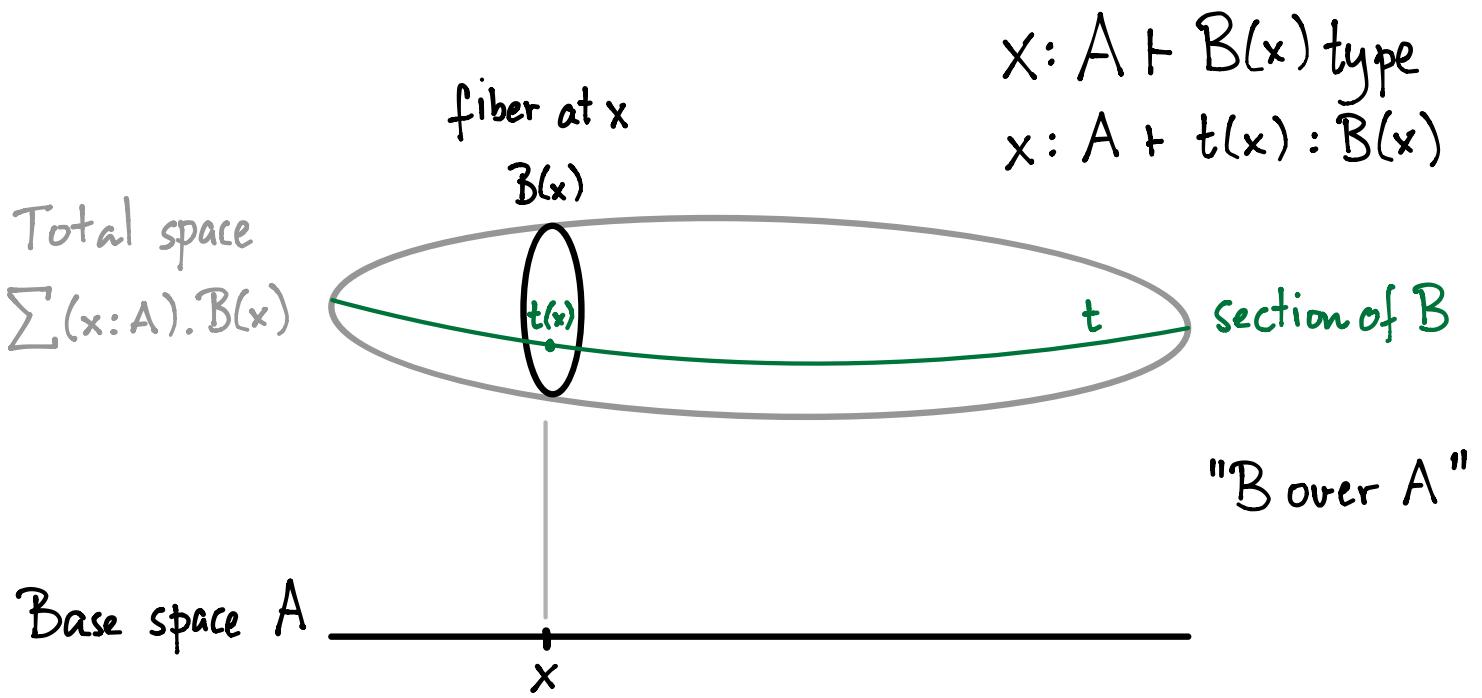
$\Gamma \vdash A \equiv B$

$\Gamma \vdash s \equiv_A t$

## Rules of inference

$$\frac{P_1 \quad P_2 \quad \dots \quad P_n}{C} \quad \begin{array}{l} \text{premises} \\ \text{conclusion} \end{array}$$

# The geometric picture



# Basic constructions

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

- elements: ordered pairs  $(s, t)$  where  $s:A, t:B(s)$
- operations: projections  $\pi_1$  &  $\pi_2$
- equations  $\pi_1(s, t) \equiv_A s$   
 $\pi_2(s, t) \equiv_{B(s)} t$   
 $(\pi_1 u, \pi_2 u) \equiv_{\sum(x:A). B(x)} u$

## Binary product $A \times B := \sum(x:A). B$ ( $B$ does not depend on $A$ )

## Dependent product $\prod(x:A).B(x)$

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

- operation: application

$$\frac{\Gamma \vdash s : \prod(x:A).B(x) \quad \Gamma \vdash t : A}{\Gamma \vdash st : B(t)}$$

- equations:

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

$$\frac{\lambda x:A.sx \equiv_{\prod(x:A).Bx} s}{\Gamma, x:A \vdash u(x) \equiv_{B(x)} v(x)} \quad \text{extensionality rule}$$

"η-rule"  
 ↑ · inter-  
 ↓ derivable

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

Function space  $A \rightarrow B := \prod(x:A).B$

(B does not depend on A)

## Natural numbers $\mathbb{N}$

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

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

$$\text{ind}_{\mathbb{N}}: P_0 \rightarrow (\prod(k:\mathbb{N}). P_k \rightarrow P(Sk)) \rightarrow \prod(n:\mathbb{N}). P_n$$

- equations

$$\text{ind}_{\mathbb{N}} s t 0 \equiv_{P_0} s$$

$$\text{ind}_{\mathbb{N}} s t (Sn) \equiv_{P(Sn)} t n (\text{ind}_{\mathbb{N}} s t n)$$

## Finite types

① empty type:  $\frac{\vdash t : \emptyset}{\vdash \text{exfalso} : A}$

1 unit type: element  $() : \mathbb{1}$   
equation  $s \equiv_{\mathbb{1}} t$

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

Construction by cases: given type  $P$  over  $2$

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

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

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

## Universe $\mathcal{U}$ - a type of types

- elements are types (not all of them)
- closed under  $\Sigma, \Pi$  and identity types

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

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

Observation: A dependent type  $B$  over  $A$  is a map

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

# Exercises

① Construct a point of

$$(\prod(x:A). \sum(y:B). C \times y) \rightarrow \sum(f:A \rightarrow B). \prod(x:A). C \times (fx)$$

where A and B are types and  $C:A \rightarrow B \rightarrow U$ .

② Use  $\text{ind}_N$  to define  $\text{double}: N \rightarrow N$

which doubles a number, e.g.:  $\text{double}(S(S0)) \equiv S(S(S(S0)))$ .

③ Define  $\text{halve}: N \rightarrow N$

which halves a number (rounded down).

## Part 2: Identity types

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

$\text{Id}_A(s, t)$  the identity type  
(path type)

We think of  $p : \text{Id}_A(s, t)$  as a path from  $s$  to  $t$ .  
We shall also use the notation

$s =_A t$  or just  $s = t$   
(not to be confused with  $s \equiv_A t$ )

For any  $t : A$  there is

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

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.

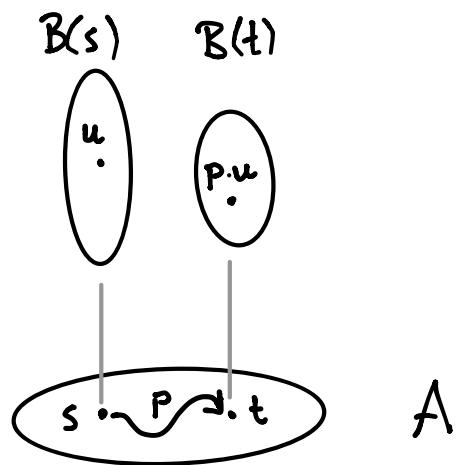
First attempt:

Given  $B: A \rightarrow U$

$s, t: A$

$p: s =_A t$

$u: B(s)$



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

$$\mathsf{TT}(x:A)(y:A)(p:x=_A y). B(x,y,p)$$

Then it suffices to construct a point of

$$d : \mathsf{TT}(z:A). B(z,z,\mathsf{idpath}\ z).$$

We write

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

for the point so constructed.

Equality:

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

Example: path concatenation

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

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

$$\_ \cdot \_ : \prod(x:A)(y:A)(p:x=y) \prod(z:A)(q:y=z) . x = z$$

$\underbrace{\hspace{10em}}_{B(x,y,p)}$

We apply path induction:

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

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

$$\lambda x z q \cdot q$$

Answer:

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

Equation:

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

## Transport:

$$\prod(x, y : A)(p : x =_A y) . \mathcal{B}(x) \rightarrow \mathcal{B}(y)$$

Path induction:

$$\prod(z : A) . \mathcal{B}(z) \rightarrow \mathcal{B}(z)$$

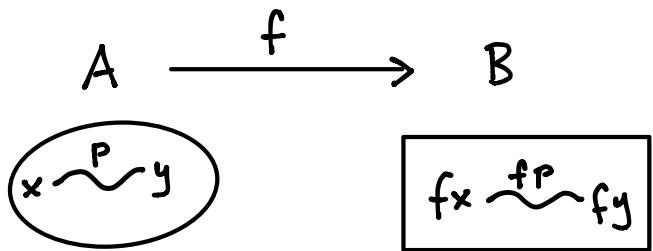
$$\lambda z \ u \ . \ u$$

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

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

## Action on paths

Given  $f: A \rightarrow B$



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

By path induction:

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

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

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

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

## Exercises:

① Construct a path between paths:

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

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

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

③ Suppose  $u, v : \sum(x:A). B(x)$

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

$$q : p \cdot (\pi_1 u) =_{B(\pi_1 u)} v$$

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

## Part 3: Homotopy levels

Higher-dimensional paths:

$t : A$  point       $\cdot t$   
 $p : s =_A t$  path       $s \xrightarrow{p} t$   
 $\alpha : p =_{s =_A t} q$  2-path       $s \xrightarrow{\alpha} t$   
(and so on)

Sometimes the higher-dimensional paths collapse above dimension  $n$ .

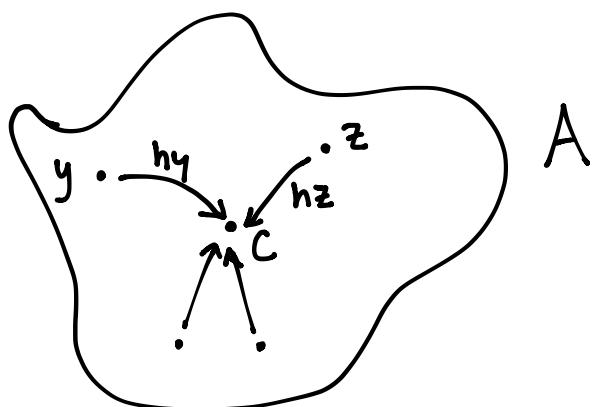
We start with the collapse of points:

Definition: A type  $A$  is contractible when  
 $\text{isContr } A := \sum_{x:A} \text{TT}(y:A) . x =_A y$   
 has a point.

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

- $c : A$  is the center of contraction
- $h : \text{TT}(y:A) . c =_A y$  is a homotopy from  $\text{const}_c$  to  $\text{id}_A$

(observe that  $h : \text{TT}(x:A) . \text{const}_c x =_A \text{id}_A x$ )



Example:  $\mathbb{1}$  is contractible

center: ()

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

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

We now generalize to all levels.

### Definition

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

$(-2)$ -type  $A := \text{isContr } A$

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

Fact:  $n\text{-type } A \rightarrow (n+1)\text{-type } A$

## Special cases of interest:

→ (-1)-types are called propositions:

- equivalent formulation:

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

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

→ 0-types are called ssets:

- equivalent formulation:

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

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

→ 1-types are called groupoids

- no interesting 2-paths

# Property vs. structure

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

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

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

## Example:

- The structure of a monoid on a set M:

Monoid S :=

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

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

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

- The property of a monoid being a group:

isGroup(S, m, e, α, β) :=

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

- The structure of a group on a set S

Group S :=  $\sum(M: \text{Monoid } S). \text{isGroup}(M)$

# Truncation:

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

$$\frac{\text{A type}}{\|A\|_n \text{ type}}$$

$$\frac{t : A}{\|t\|_n : \|A\|_n}$$

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

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

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

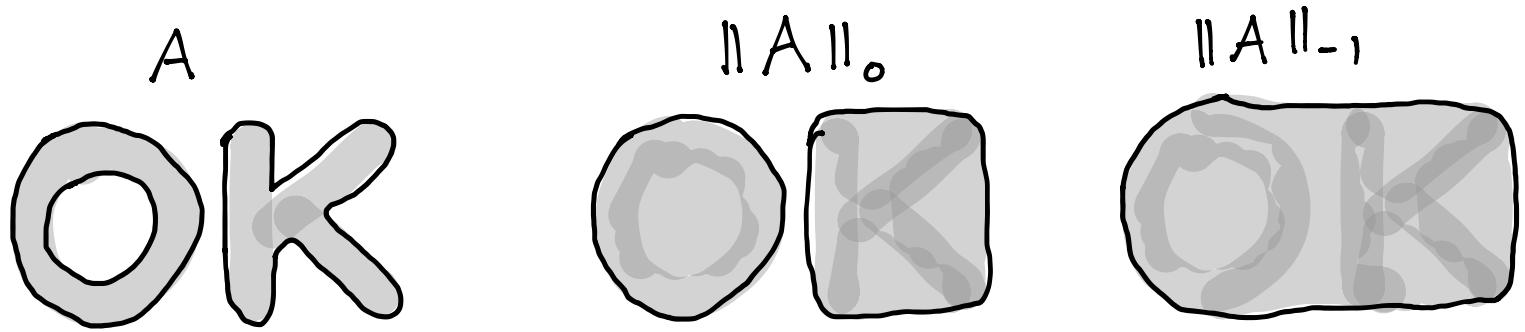
$$\begin{array}{ccc}
 A & \xrightarrow{\quad \mid_{\ln} \quad} & \|A\|_n \\
 & \searrow f & \downarrow \\
 & B & \text{n-type}
 \end{array}$$

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

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

$$\perp := \emptyset$$

$$\neg P := P \rightarrow \perp$$

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

$$P \Rightarrow Q := P \rightarrow Q$$

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

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

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

These satisfy the rules of intuitionistic logic.

Caveat:

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

# Importance of truncation for mathematics

→ Surjection vs. retraction:  $f : A \rightarrow B$

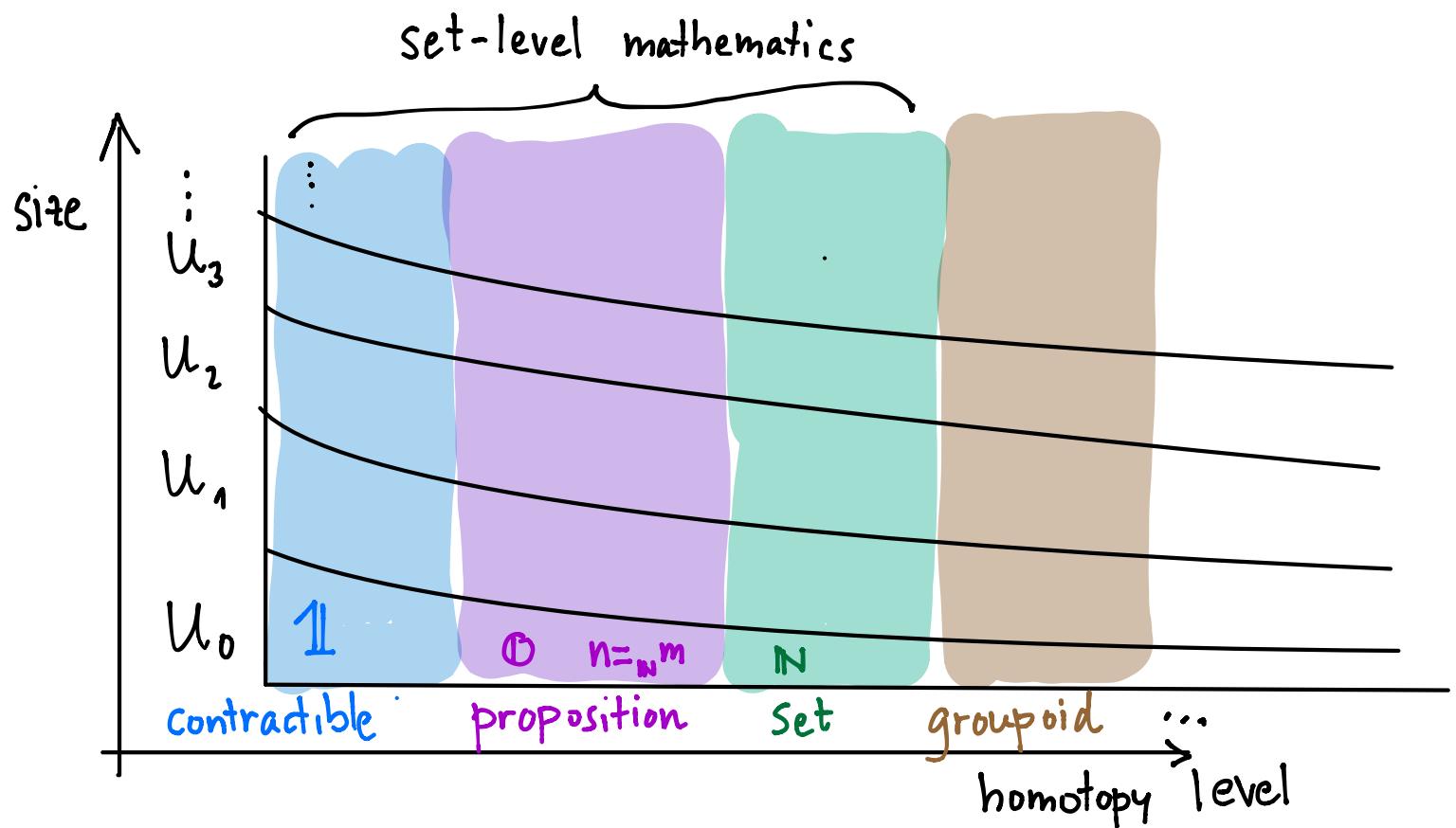
$$\text{TT}(y:B) . \sum(x:A) . f x =_B y \quad \text{"f has a right inverse"}$$

$$\text{TT}(y:B) . \parallel \sum(x:A) . f x =_B y \parallel \quad \text{"f is surjective"}$$

→ Loop space vs. fundamental group

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

$$\pi_1(A, a) := \parallel \Omega(a, a) \parallel_0 \quad \text{fundamental group}$$



## Exercises:

- ①  $\text{isContr } A \rightarrow \prod(x,y:A). \text{isContr}(x=_A y)$
- ② Is contractibility a property or a structure?
- ③  $\prod(x:A). \text{isContr}(\sum(y:A), x=_A y)$

## Part 4: Equivalences

### Function extensionality

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

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

We have:

$$f =_{\prod(x:A). Bx} g \rightarrow f \sim g \quad (\text{exercise})$$

How about the other way?

Function extensionality:

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

It turns out that univalence implies function extensionality.

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

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

$$\text{isIso } f := \sum(g: B \rightarrow A). (f \circ g \sim \text{id}_B) \times (g \circ f \sim \text{id}_A)$$

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

$$\text{isEquiv } f := \text{TT}(y: B). \text{isContr}(\sum(x: A). fx =_B y)$$

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

$$\text{isBiinv } f := (\sum(g: B \rightarrow A). f \circ g \sim \text{id}_B) \times (\sum(h: B \rightarrow A). h \circ f \sim \text{id}_A)$$

Observations:

- $\text{isEquiv}$  is a property
- $\text{isIso}$  is a structure in general
- $\text{isBiinv}$  is a property & equivalent to  $\text{isEquiv}$

Define:

$$A \cong B := \sum(f: A \rightarrow B). \text{isIso } f$$

$$A \simeq B := \sum(f: A \rightarrow B). \text{isEquiv } f$$

We prefer  $\simeq$  because  $\text{isEquiv}$  is a property.

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

Equivalence is equivalence:

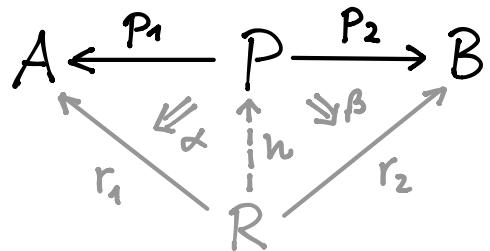
$$A \simeq A$$

$$A \simeq B \rightarrow B \simeq A$$

$$A \simeq B \rightarrow B \simeq C \rightarrow A \simeq C$$

# Equivalences & universal properties

Recall the notion of a product diagram



How to express its universal property?

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

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

$$\text{isEquiv}(e_R) = \text{TT}(r_1, r_2, \alpha, \beta) . \text{isContr}(\sum(h : R \rightarrow P) . e_R(h) = (r_1, r_2, \alpha, \beta))$$

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

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

(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

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

② Use equivalences to express the fact that

$$I_{\perp_1}: A \longrightarrow \|\mathcal{A}\|_{\perp_1}$$

is the universal map from  $A$  to propositions:

$$\begin{array}{ccc} A & \xrightarrow{I_{\perp_1}} & \|\mathcal{A}\|_{\perp_1} \\ & \searrow f & \downarrow \tilde{f} \\ & & P \end{array}$$

for any proposition  $P$  and  $f: A \rightarrow P$  there is a unique  $\tilde{f}: \|\mathcal{A}\|_{\perp_1} \rightarrow P$  such that  $\tilde{f} \circ I_{\perp_1} = f$ .

## Part 5: Univalence

Function extensionality:

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

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 extensionality principles:

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

Do we have such a principle for universes?

There is a map

$$\text{idtoequiv} : \prod(A, B : \mathcal{U}). A =_u B \rightarrow A \simeq A$$

Defined by path induction

$$\text{idtoequiv} \circ \text{idpath}_c := (\text{id}_c, \dots)$$

Definition:

A universe  $\mathcal{U}$  is univalent if  
 $\text{idtoequiv}$  is an equivalence.

$$(A =_u B) \simeq (A \simeq B)$$

"Identity is equivalent to equivalence."

Axiom of univalence: Every universe is univalent.

In practice we use the inverse of  $\text{idtoequiv}$ :

$$ua : (A \simeq B) \longrightarrow A =_u B$$

Given an equivalence  $e : A \simeq B$  we get  $ua(e) : A =_u B$ ,  
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

$$\text{Set}_u := \sum(A : u). \text{isSet } A$$

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  
 $\sum(A : U) . \text{isProp } A \times A$   
is contractible.
- ② Show that  $\sum(A : U) . \text{isSet } A$  is not a set.  
Hint:  $(2 \simeq 2) \simeq 2$ .