

Introduction to Homotopy Type Theory

Nicolai Kraus

Midlands Graduate School 2021, 12–16 April

Ways to introduce (homotopy) type theory

↙ Agda

```
x--mono : ∀ {x y z : Brw} → y ≤ z → x • y ≤ x • z
x--mono ≤-zero = ≤-zero
x--mono (≤-trans y≤w w≤z) = ≤-trans (x--mono y≤w) (x--mono w≤z)
x--mono {x} (≤-succ-mono {y} {z} y≤z) = +x-mono x (x--mono {x} y≤z)
x--mono {x} {y} (≤-cocone f {k = k} y≤z) with decZero x
... | yes x≡zero = subst (λ z → z ≤ zero) (sym (zero · y≡zero y)) • cong
... | no x≠zero = ≤-cocone (λ n → x • f n) {k = k} (x--mono y≤z)
x--mono {x} (≤-limiting f f≤z) with decZero x
... | yes x≡zero = ≤-zero
... | no x≠zero = ≤-limiting (λ n → x • f n) λ k → x--mono (f≤z k)
x--mono (≤-trunc p q i) = ≤-trunc (x--mono p) (x--mono q) i
```

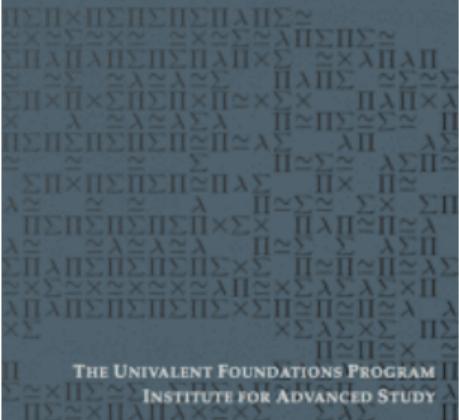
↙ inference rules

$$\frac{\Gamma, x:1 \vdash C : \mathcal{U}_i \quad \Gamma \vdash c : C[\star/x] \quad \Gamma \vdash a : 1}{\Gamma \vdash \text{ind}_1(x.C, c, a) : C[a/x]} \text{ 1-ELIM}$$

↙ natural
math language

Homotopy Type Theory

Univalent Foundations of Mathematics



Prerequisites

- Basic examples of types

Nat Bool Unit Empty

- Universe(s)

Type

$A, B : \text{Type}$

- Function types

$A \rightarrow B$

- Dependent function types

if $C : A \rightarrow \text{Type}$, then

- Binary products (pairs) and coproducts (sums)

$A \times B$

$A \sqcup B$

$(a : A) \rightarrow C_a$

- Dependent pairs

$\sum_{(a:A)} C_a$

$[(a : A) \rightarrow C_a]$

- Inductive types

see above

Program = Proof

Exercise: Formulate a type which says that there are infinitely many primes
(or simply: arbitrarily large primes)!

$$\begin{aligned} \text{isPrime : } & \text{Nat} \rightarrow \text{Type} \\ \text{isPrime}(n) := & ((m \ k ; N) \rightarrow (m \cdot k = n) \rightarrow (m=1) \oplus (m=n)) \\ & \times (n > 1) \\ \text{bigPrimes : } & (n: \text{Nat}) \rightarrow \sum_{(p: \text{Nat})} \text{isPrime}(p) \\ \text{bigPrimes} := & (\text{exercise}) \qquad \qquad \qquad \times p > n \end{aligned}$$

Caveat: judgments (meta-theoretic)

- A type "A is a valid type"
- $a : A$ "a is term of type A"
- $a \equiv b$ "a and b are the same" $(\lambda x. f x) y$
- $a : \equiv b$ same, by definition $\equiv f y$

We cannot ask or prove these statements *in* the language.

Exercise: Which of the following can we ask internally?

- (1) Is 8 a natural number? no, b/c $8 : \text{Nat}$ is judgment
- (2) Is 8 a prime number? yes, b/c $\text{isPrime}(8)$
- (3) Is "hello" a prime number? no, $\text{isPrime}("hello")$
is not a type

(Identity a.k.a. equality a.k.a. identification a.k.a. path) types

- if $a, b : A$ then $a = b$ type "formation"
- given $a : A$ we have $\text{refl} : a = a$ "introduction"
- given $C : (a, b : A) \rightarrow (a = b) \rightarrow \text{Type}$ "elimination"
and $C a a \text{refl}$ (for all $a : A$)
we get $C a b p$ (for all $a, b, p : a = b$)] aka path induction
- applying this rule and asking for the A refl case gives back the assumption

$$\text{sym} : (a, b : A) \rightarrow a = b \rightarrow b = a \quad \text{"computation"}$$

here:

$$\text{Sym } a \underset{a}{\cancel{=}} b \underset{b}{\cancel{=}} \text{refl} := \text{refl} \quad C a b p := (b = a)$$

$$\text{computation: Sym } a a \text{refl} \equiv \text{refl}$$

Model: types as spaces

The homotopical interpretation of dependent type theory works roughly like this:

judgment interpretation

A type “ A is a space”

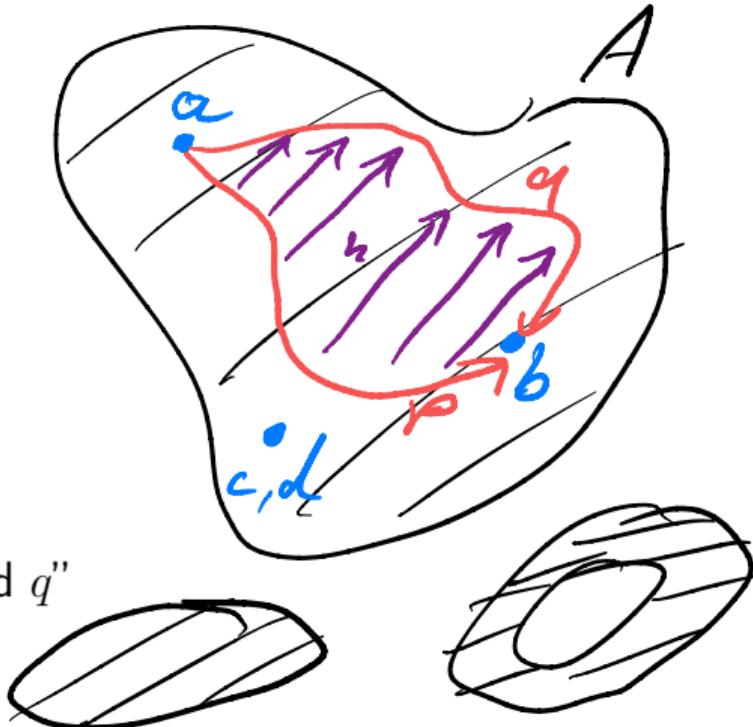
$a : A$ “ a is a point in space A ”

$c \equiv d$ “ c and d are the same point”

$p : a = b$ “ p is a path between a and b ”

$h : p = q$ if p and q are equalities $a = b$:
“ h is a homotopy between p and q ”

↑
path in
 $\alpha = b$



Types as spaces, examples: symmetry, transitivity

$\text{sym} : (a\ b : A) \rightarrow a = b \rightarrow b = a$

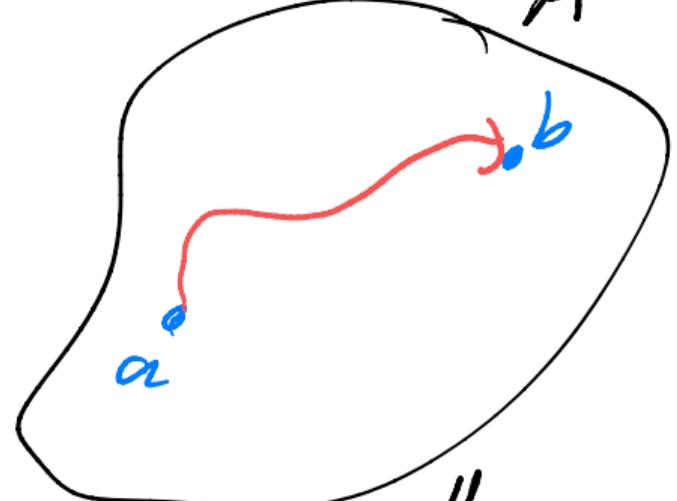
~~$\text{sym } a \not\sim b$~~

$\text{trans} : (a\ b\ c : A) \rightarrow$

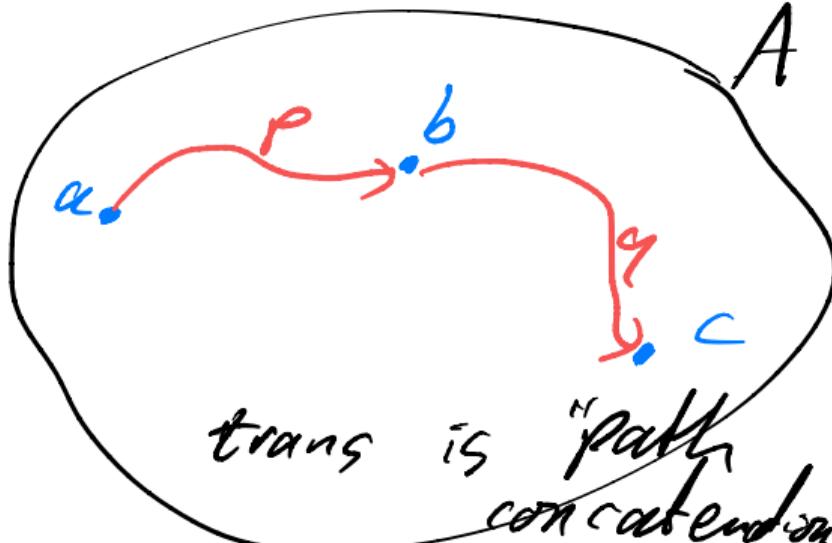
$a = b \rightarrow b = c \rightarrow a = c$

~~$\text{trans } a \not\sim c$~~

~~$a \not\sim q \rightarrow b \not\sim q$~~



sym is "path reversal"



trans is "path concatenation"

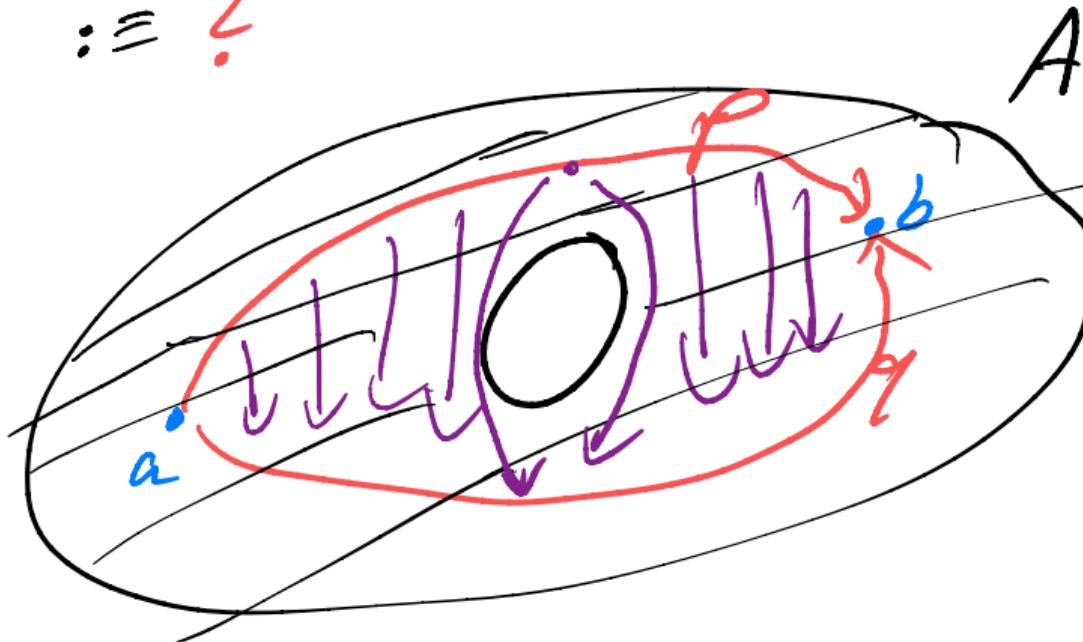
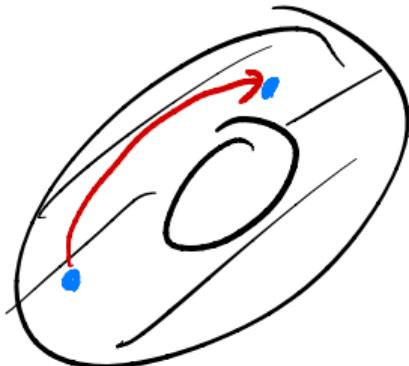
Types as spaces, example: uniqueness of identity proofs

Can we prove this for all types A ?

VIP

$\text{uip} : (a b : A) \rightarrow (p q : a = b) \rightarrow p = q$

~~uip a b~~ ~~p q~~ $\vdash ?$



A quiz

given in addition:

Given $A : \text{Type}$ with $x, y : A$

$B : (a, b : A) \rightarrow (p : a = b) \rightarrow \text{Type}$

$C : (p : x = y) \rightarrow \text{Type}$

$D : (a : A) \rightarrow (p : a = y) \rightarrow \text{Type}$

$e_1 : (a : A) \rightarrow B \ a \ a \ \text{refl}_a$

$e_3 : D \ \text{refl}_y$

Below are three statements which we may want to prove. Which is (probably) the most difficult and which the easiest?

exercise₁ : $(a, b : A) \rightarrow (p : a = b) \rightarrow B \ a \ b \ p$

exercise₂ : $(p : x = y) \rightarrow C \ p$

exercise₃ : $(a : A) \rightarrow (p : a = y) \rightarrow D \ a \ p$

after path induction, we
need to show:

$(a : A) \rightarrow B \ a \ a \ \text{refl}_a$

(nothing we can do)

$D \ \text{refl}_y$

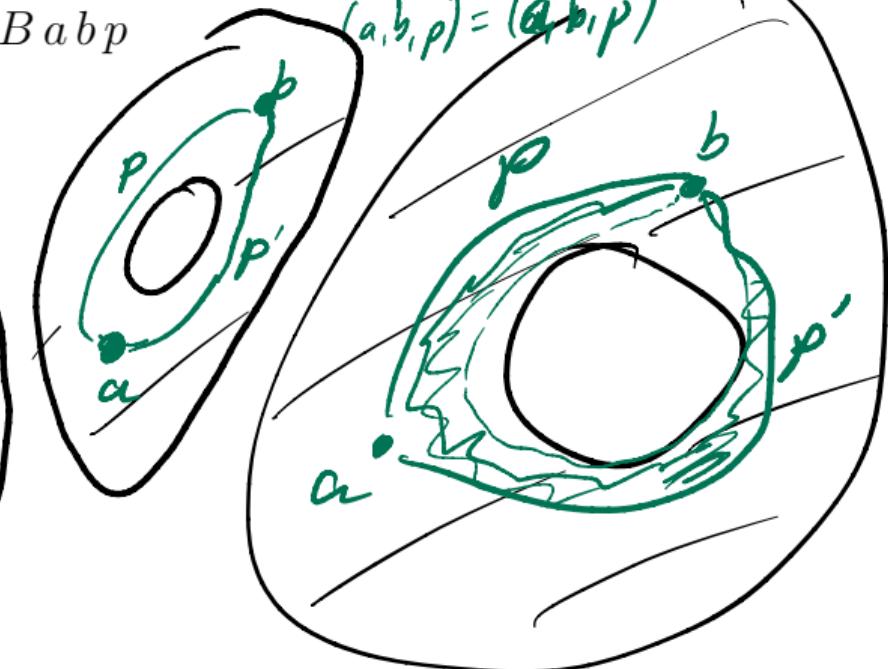
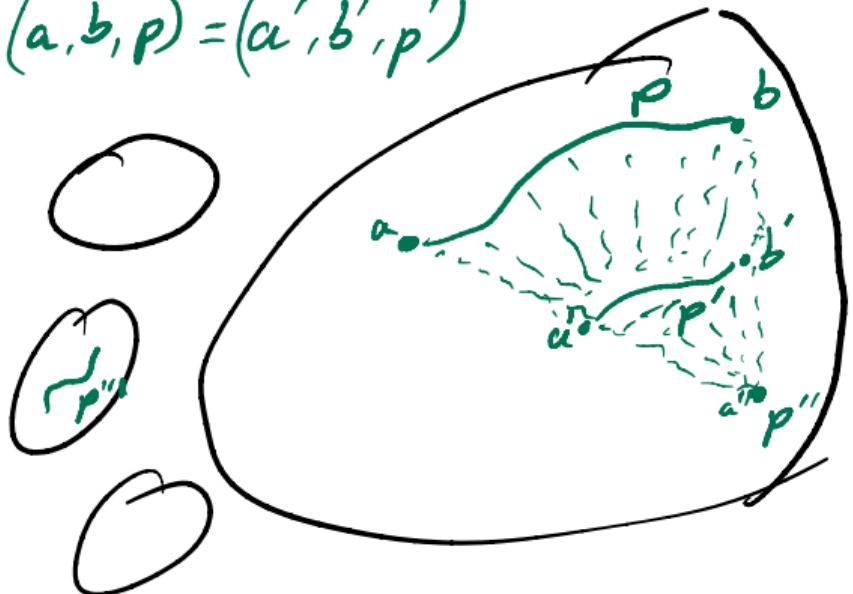
A quiz (2)

Given $A : \text{Type}$ with $x, y : A$

$B : (a b : A) \rightarrow (p : a = b) \rightarrow \text{Type}$

exercise₁ : $(a b : A) \xrightarrow{(a:A)} (p : a = b) \rightarrow B a b p$

$$(a, b, p) = (a', b', p')$$



caveat:
cannot show $p = p'$
can show
 $(a, b, p) = (a', b', p')$

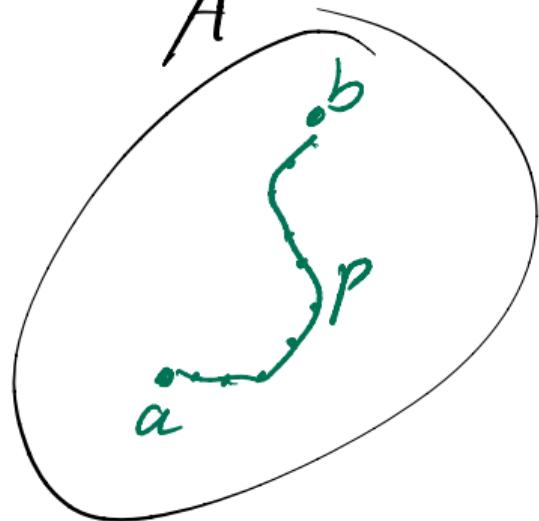
Types as spaces, examples: map on paths

Given $f : A \rightarrow B$, we have:

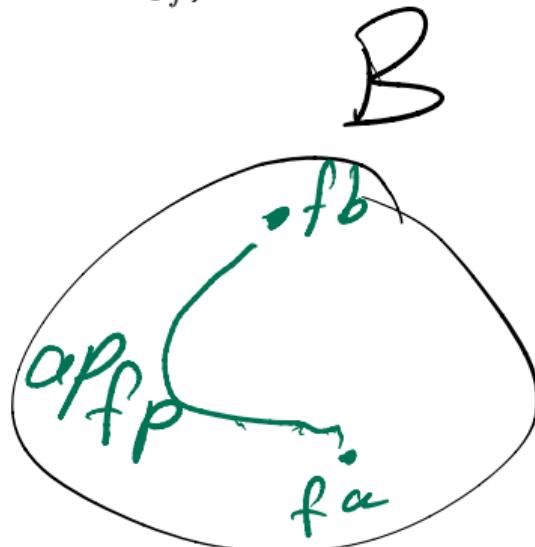
$$\text{ap}_f : (a b : A) \rightarrow a = b \rightarrow f a = f b$$

(a.k.a. cong_f)

$$\text{ap}_f \ a \cancel{b} \ \cancel{\text{refl}}_a : \equiv \text{refl}_f$$



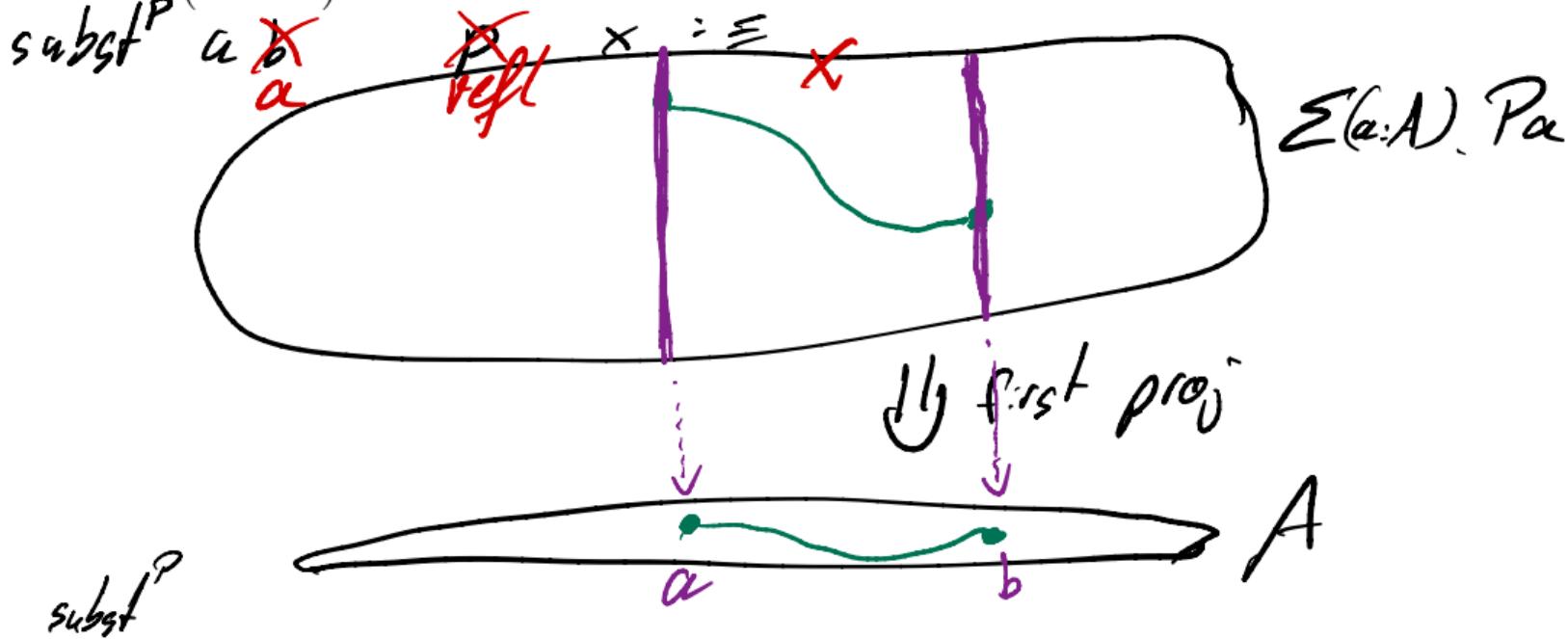
$$f \Rightarrow$$



Types as spaces, example: substitution (a.k.a. transport)

Given a type A and $P : A \rightarrow \text{Type}$, we have:

$$\text{subst}^P : (a, b : A) \rightarrow a = b \rightarrow P a \rightarrow P b$$



Isomorphisms and Equivalences

When is $f : A \rightarrow B$ an isomorphism?

$$\begin{aligned} \text{isIso}(f) &:= \exists g : B \rightarrow A \\ &\quad \in \alpha : (a:A) \rightarrow g(fa) = a \\ &\quad \in \beta : (b:B) \rightarrow f(gb) = b \end{aligned}$$

question: given $\alpha : A$, how do you show $f(g(fa)) = fa$.

solution: $\beta(fa)$ or $\alpha f(ga)$

\Rightarrow ambiguity /
which we usually
want to avoid

eg: in category
theory

$$h \xrightarrow{g} \xrightarrow{f} l$$
$$h \circ (g \circ f) = (h \circ g) \circ f$$