Introduction to Homotopy Type Theory

Department of Computer Science
University of Verona

Iosif Petrakis

Summer Term 2017

Overview of this Course

Part I: A homotopic approach to Intensional Type Theory (ITT)

Chapter 1: Basic types and type constructors in ITT

Chapter 2: Applications of path-induction

Part II: Homotopic extensions of Intensional Type Theory

Chapter 3: The Univalence Axiom (UA) of Voevodsky

Chapter 4: Higher Inductive Types (HITS)

(Chapter 5: Path spaces)

- ITT
- ITT + EXT
- ITT + UA
- ITT + HITS

ITT + UA + HITS = HoTT

- A seminar talk by Dr. Xu is planned on the semantics of HTT in cubical sets and on implementation issues. (June 17).
- Main literature: HoTT book (https://homotopytypetheory.org) → many authors → one year → shorter paths

+ extras, if necessary

Brief Explanation of the basic terms

ITT : It's called intensional (contrast vs to extensional) because equality is not trivial (→ to be explained!)
or Martin-Löf's ITT
or constructive IT

a theory of types (Russell, Church...) created by Martin-Löf in order to provide new foundations for constructive mathematics (Bishop).

Main characteristics : • extensive use of inductive definitions
• no separation between logic and mathematics (like in set-theoretic foundation with 1st order logic)
• it consists of rules, it has no axioms.

It turned out to be a fundamental functional programming language and was connected more to Computer Science rather than mathematics.

HoTT : Through homotopy-theoretic semantics for ITT → w/ conforming the UA
• Awodey – Warren 2009
• Kapulkin – Lumsdaine – Voevodsky 2012
• Coquand, Bezem, Huber et al 2013,16

types in ITT are interpreted as homotopical objects

so we can use ITT to prove theorems in homotopy theory of some kind of topol. spaces

and

we can use homotopy theory to discover things about the types of ITT + (UA + HITs)

• In this course : • Informal ITT (syntax)
• Informal semantics (of) within ITT

for other work : **Naive Homotopy Type Theory** (Holypos)

• (IMPORTANT) : ∃ many variants of ITT (univ., K-axiom etc.) see Notes & ch.s in HoTT-book (Intro, Index and here) ↓ VF Axiomatic HoTT

Aims of this course

0. Experience with inductive defs (written: "writer" above "Experience")

1. At the end you can study the HoTT-book on your own, without help

2. —"— that —"— research papers on HoTT

3. To understand basic issues on the foundations of mathematics

4. To realise that foundations of math can influence math and the converse.

Bureaucracy of this course

- email: petrakis@math.lmu.de
  www.mathematik.uni-muenchen.de/~petrakis/
  Room 1.63B, Office hour Thursday 10:30–11:30 AM
  Italian email-account ....
- website of the course: you receive an email from P.J.
- Weekly Exs: Every Thursday you get new exs (this week on Monday?)
  —"— 1:30–2:30 we discuss the exs of the previous week. You participate in
  A group of 2-3 students may present the solutions? More to get used to the language of HoTT.
- Exams: Exs + Oral Exams (+ Tessina)? (Please think about it (further)) and tell me what you
  If written exam, then bonus to the top.
  Please think and tell me in 2-weeks if you prefer to do oral exams in late June / early July.
- Mo 06.03 9:30–11:30, not on 08 and 09.03 due to a conference I have to attend.
  Room? I'll give you the Gos then!!

Thursday 13.04. I won't be here.

- ~~My office hour: Wed 1:30–2:30, My office: ...~~
- 4 hours on Wednesday?? Weather: sharp $3 \times 45 = 135$ min $= 90 - 20$ min break $- 45$
  $4 \times 45 = 180$ min $= 90 - 30$ min —"— $- 90$
- Again in June I'll inform you.
- Please send me your emails! (or take them from PS?)
- Questions?

Part I

A homotopic approach to Intensional Type Theory

Chapter 1

Basic types and type constructors in Intensional Type Theory

Section 1.1 : Propositions and judgements

- A proposition is a statement that can be proved (eg. in ZFC : $a \in A$), or used as a hypothesis in a proof. (implication)
  A judgement is a statement that can be used in the ambient context of a proposition, that can be derived, but cannot be negated, used as a hypothesis in a proof etc. (i.e. it cannot be treated as a proposition)

- The basic judgements of ITT : $a : A$
  - "a is an object of type A" ("the term a is of type A")
  - "a is a proof of A"
  - "a is a point in space A"

  (You cannot say "$1 = A$")

  $a \equiv b : A$ "a, b are "definitionally" equal objects of A" ("by definition", "judgementally")

  or $a \equiv_A b$ (with context $A \equiv B : \mathcal{U}$)

  (The only judgement of 1st-order logic : "P has a proof")

  $1 \equiv_{\mathbb{N}} \mathrm{succ}(0)$

  $f(3) \equiv_{\mathbb{N}} 6$, if $f(x) \equiv 2x$, $f : \mathbb{N} \to \mathbb{N}$, if I have a type $\mathbb{N}$, ...

- Every "term" a of ITT comes with its type A, only $a : A$, for some A (unlike statement $\forall$ is not a term)
  No terms "points in isolation" (In ZFC $x \in \mathbb{R}$, and only "$x \in V$", can be constructed naively)

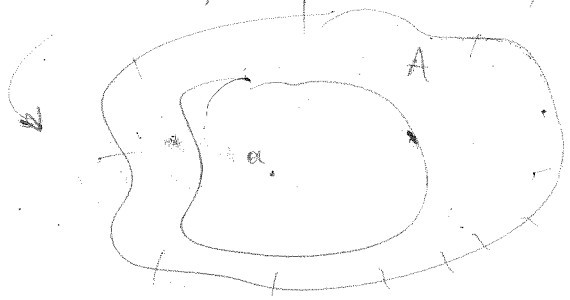
  you can say $\neg(x =_{\mathbb{R}} y)$ (propositional equality, it is a proposition)

- In Intensional Type Theory (ITT) there is a notion of equality $x =_A y$, for $x, y : A$ other than $x \equiv_A y$, which is responsible for the higher dimensional structure of its types.

- In Extensional Type Theory (ETT) no distinction between $x \equiv_A y$ and $x =_A y$.

  (here type-checking is not decidable, as in ITT)
  (with less interesting semantics)

- Both were first studied, and introduced by Per Martin-Löf.



The homotopy points of the homotopic approach to ITT

but you can't say $a : \mathcal{U}$ (only in ZFC)

open, see section on universe.

Section 12 : Universes

- In set theory was very helpful to think about a universe V of all sets, and treat it as a set again.

  (and natural from the perspective of a naive notion of set)

  But Russell's paradox : $A = \{x \in V \mid \neg(x \in x)\}$

  $A \in A \Leftrightarrow \neg(A \in A)$

  showed that V cannot be considered a set, only a "class".

In ITT this paradox cannot be reproduced, since $\neg(A : A)$ or $\neg(x : x)$ has no meaning - you cannot negate a judgement.

- But also in ITT is extremely helpful and natural to consider, initially a type $\mathcal{U}$, the universe where all other types "are objects of". In ITT

  $\mathcal{U}$ is a type

  (see its proof in ML's paper : An Intuitionistic theory of types !!)

  (maybe we discuss this proof)

To avoid circularities (actually a version of the Burali-Forti paradox was formulated in ITT by Girard, which showed the necessity of univ) a hierarchy of universes

$\mathcal{U}_0, \mathcal{U}_1, \mathcal{U}_2, \ldots$ ← In MLTT, following Grothendieck) hierarchy of universes !!

all of them behave as types, under the following Rules

(RU1): $\dfrac{m < n}{\mathcal{U}_m : \mathcal{U}_n}$; (some naive notion of naturals is used here !)

(RU2): $\dfrac{A : \mathcal{U}_m \quad (\text{small type} \leftarrow \text{large type}), \quad m < n}{A : \mathcal{U}_n}$ (a type that includes $\mathcal{U}$ in larger eg $A \to \mathcal{U}$ ← universe)

(the universes are "cumulative")

no unique typing (unpleasant) "for space-reasons"

- Usually we work with $\mathcal{U}$, or $\mathcal{U}$ and $\mathcal{U}'$, its successor.

Remark 12.1 If $A, B : \mathcal{U}$ (usually we omit the index) then A, B are "objects" a type and the equality $A = B : \mathcal{U}$ special cases of the initial judgement

or $A \equiv_{\mathcal{U}} B$

is meaningful.

(RU3): Conversion rule for types in a universe : $\dfrac{a : A \quad A : \mathcal{U}, \quad B : \mathcal{U}, \quad A \equiv_{\mathcal{U}} B}{a : B}$ (here)

(of course we have such a rule for every $\mathcal{U}_n$ and $\mathcal{U}_n$)

- We can give examples of $A \equiv B$ after introducing type-families

  i.e. $P : U \to U$

  If $P \equiv \lambda(A:U).\ A$, then

  $P(A) \equiv A$

- If $P : A \to U_0$, then I cannot get $P : A \to U_1$, since there is no rule that permits it. What I can do though is

  $[\ ]\ (a:A).\ P(a)\ ] : A \to U_1$

  (see Notes + Ch1 from HoTT-book).

  (• If we wanted to show $P(A)$ for every A, we would use this induction principle Ind$_U$)

- Remark 1.2.2: • Usually when we define a type we include an induction axiom which guarantees that the type under definition is the "closest" with the required properties. Universes are not defined inductively this way here, (OPEN UNIVERSE) there are other versions of HTT for which such an axiom is given, since we want to keep U (a $U_0, U_1, \ldots$) open-ended spaces, so that we can add new basic types or use new type constructions if necessary. (This was Martin-Löf's original approach to U too)

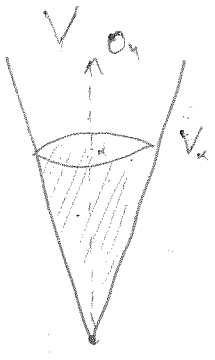
  • This will be essential in Part II, chapter 4, where HITs are going to be introduced, "open basic types".

- Remark 1.2.3: In (axiomatic) ZFC the universe V is described as the following cumulative hierarchy of sets:

  $V_0 = \emptyset$

  $V_{\alpha+1} = \mathcal{P}(V_\alpha)$

  $V_\lambda = \bigcup_{\alpha<\lambda} V_\alpha$, λ limit ordinal



With the axiom of foundation every set is well-founded, and there is $\in$

- transfinite induction on On: $P(0)$, $P(\alpha) \Rightarrow P(\alpha+1)$,

  $(P(\alpha))_{\alpha<\lambda} \Rightarrow P(\lambda)$

- to show $\forall_{x \in V}\ P(x)$: $\forall_{\alpha \in On}\ (\underbrace{\forall_{x \in V_\alpha}\ (P(x))}_{\Phi(\alpha)})$

- i.e. V in ZFC is not open-ended, it is a (CLOSED) universe

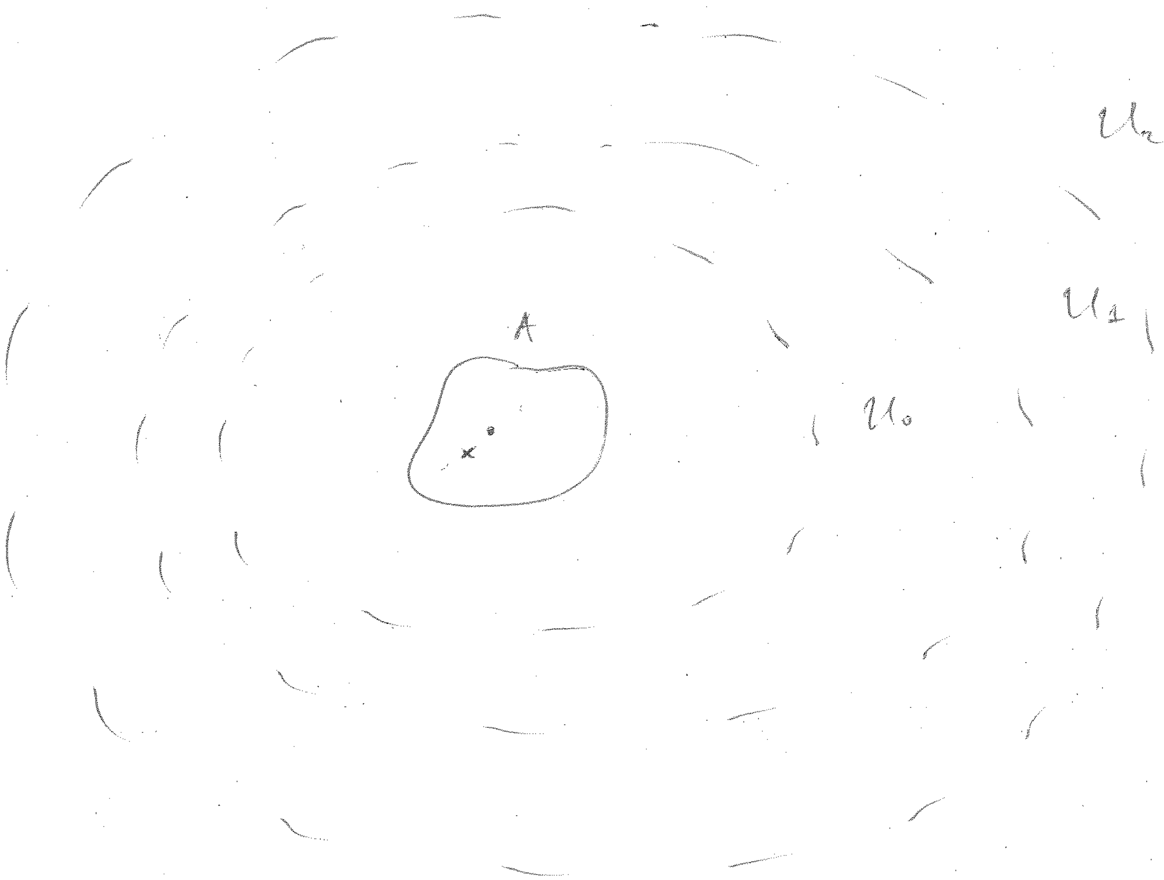
- With the set-constructors $A \mapsto \mathcal{P}(A)$
  $A, B \mapsto A \cup B$
  $A, \varphi \mapsto \{x \in A \mid \varphi(x)\}$
  pairing, replacement
  ($\emptyset$) definable $\{x \neq x\}$
  · infinite set

out with only one basic set

all basic sets used in mathematics are going to be defined in V.

In ITT we have many basic types, and we may add more, if we want

- We have more than one universes
- (V can be "embedded" in HoTT)
- The type constructors of ITT correspond to the set-constructors of ZFC



- Similarity between ZFC and ITT: first the axioms, then their models.
  (unlike group-axioms, the group models preexisted)

- $\mathcal{U}' = \mathcal{U}_{n+1}$, if $\mathcal{U} = \mathcal{U}_n$ (standard notation)
  $\mathcal{U}, \mathcal{U}', \mathcal{U}'', \mathcal{U}''', \ldots$
  $\mathcal{U} : \mathcal{U}' : \mathcal{U}''$;

can we avoid theory of $\mathbb{N}$?

No problem: for the theory of $\mathbb{N}$ you need the first universe $\mathcal{U}$ only

So, one develops $\mathbb{N}$, and comes back to RU_1, RU_2.

Section 1.3: Function Types (a non-dependent function type)

Not inductively defined but with the following rules: (no for we can form $U \to U$, $U \to U'$, $U \to (U \to U)$ etc)
but with no diff of some basic types very soon will have more interesting examples at hand.)
but this might be too early for, before introducing new basic types

- Form$_{A \to B}$ $\quad \dfrac{A, B : \mathcal{U}}{A \to B : \mathcal{U}}$

- Intr$_{A \to B}$ $\quad \dfrac{x : A \;,\; b(x) : B \quad \text{(if we have seen a new judgement)}}{\left(x : A \;,\; b : B\right) \text{(given } x \text{ this means)}}$

$$[\lambda (x : A).\, b(x)] : A \to B$$

by explicit definition

- definition of an element of $A \to B$ via λ-abstraction (Church)
- the rule introduces an element of the new type ("canonical")

Implication introduction

- Elim$_{A \to B}$ (or Eval)

$$\frac{f : A \to B \;,\; x : A}{f(x) : B}$$

- the rule eliminates the elements of the new type: it explains how the terms of the type can be used in derivations

here we get eliminate since it is not explicitly

- Comp$_{A \to B}$ $\quad \dfrac{(x : A \;,\; b : B) \;,\; a : A}{[\lambda (x : A).\, b](a) \equiv b(a)}$

, where $\quad b(a) \equiv b[a/x]$

- the comp-rule is judgemental equality
- the comp-rule explains what happens when the elim-rule applies to the intro-rule. Also known this specific one as β-conversion (compatibility between Intro + Elim) or β-reduction

- Uniq$_{A \to B}$ $\quad \dfrac{f : A \to B}{f \equiv \lambda (x : A).\, f(x)}$

η-conversion or η-expansion (alpha)

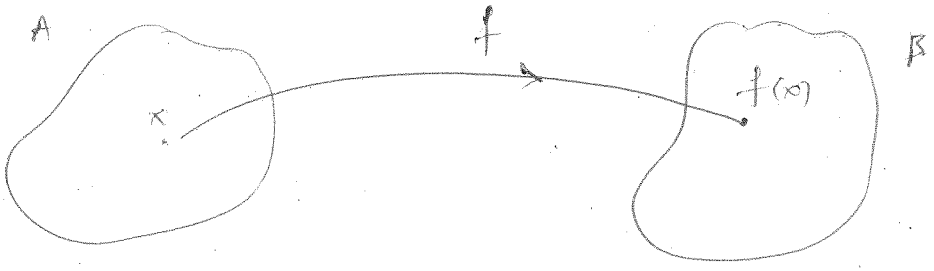
- this optional uniqueness principle is a judgemental equality that explains how every element of the type is uniquely determined by the results of the elim-rule applied to it.

Here: Not from f
use elim on $f(x)$
this means to compute an element of B given an element of A namely f.
Elim-intro$_{A \to B}$ of this method gives f again.

- sometimes we write $\lambda x.\, b$ for simplicity

- This pattern of rule given for the definition of a type in general.
If the type is defined inductively, then Elim (+ Comp) replaced by Ind.
~~Clearly Comp and Uniq rules are meaningful only when Intro is not inductive!!~~

- $\mathrm{Univ}_{A\to B}$ will be explained in the proof of EXT through the HIT: Interval



- Examples: . $\overline{b}_A \equiv \lambda(x:A).\, b$, where $b:B$, is the constant function b on A. ($b(x) \equiv b$, for every x)

. $\mathrm{id}_A : A \to A$, $\mathrm{id}_A \equiv \lambda(x:A).\, x$

- Remark 1.3.1: The definition of $A \to B$, eg $\mathrm{Univ}_{A\to B}$, is not "uncircular", for you have an operation $x \mapsto b(x)$, and though if you write down an element of $A \to B$. You don't know what a function is, you rely on a notion of correspondence. (somehow intuitively grey) CIRCLE IT in the future.

- Definition 1.3.2: $f \equiv \lambda(x:A).\, f(x)$ $\quad f:A\to B$ $\quad A,B,C:\mathcal{U}$

$g \equiv \lambda(y:B).\, g(y)$ $\quad g:B\to C$

$g \circ f \equiv \lambda(x:A).\, g(f(x)) \quad : A \to C$

$\lambda(x:A).\, b(x)$ ~ $\lambda(y:B).\, c(y)$ ~ $\lambda(x:A).\, c(x)$

- Exercise: $f : A \to A$

$f \circ \mathrm{id}_A \equiv \lambda(x:A).\, f(\mathrm{id}_A(x))$

$\equiv \lambda(x:A).\, f(x)$

$\equiv f$

Here we use: $b(x) \equiv c(x)$, for every $x:A$

then $\lambda(x:A).\, b(x) \equiv \lambda(x:A).\, c(x)$

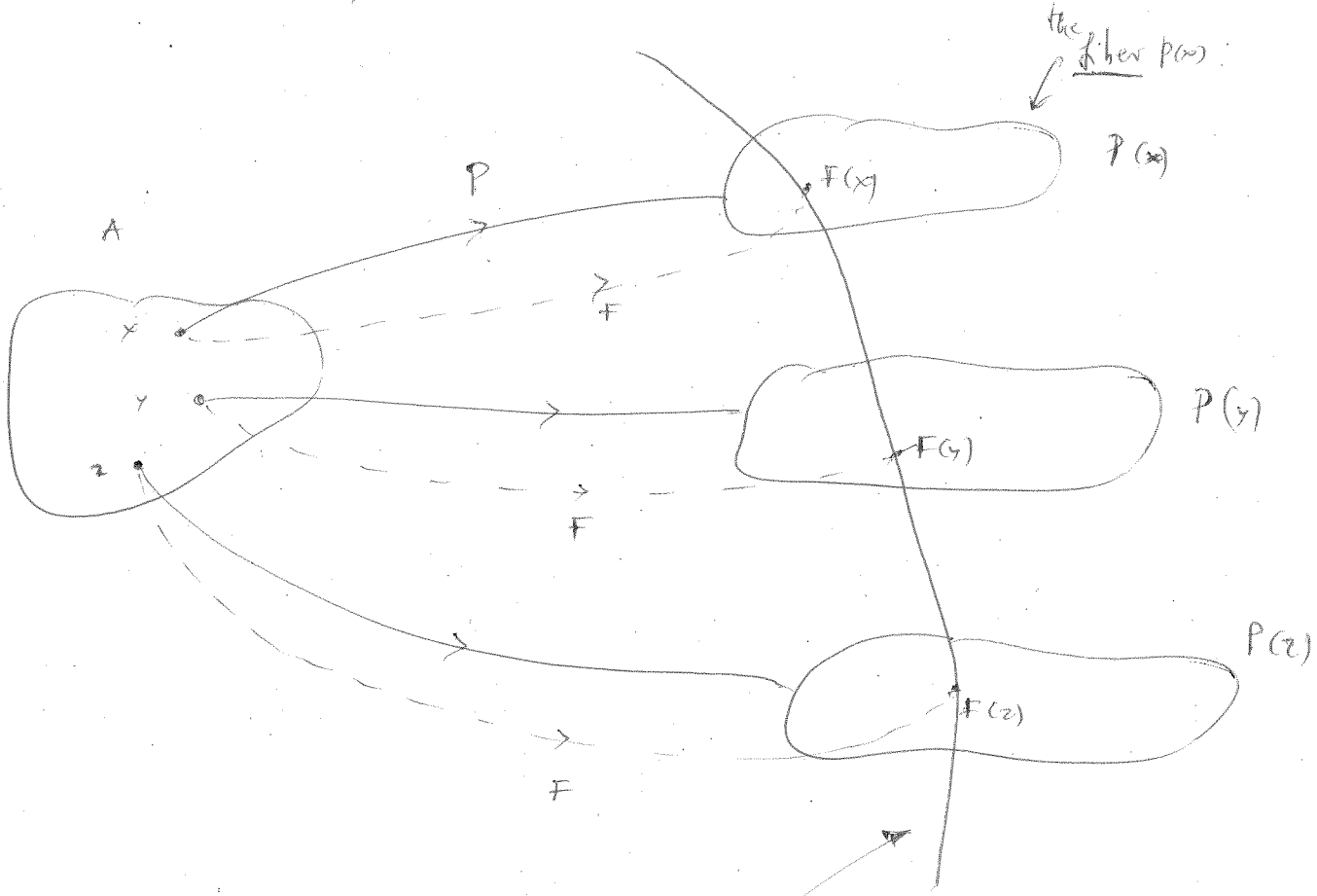
(Judgemental extensionality), how we can add as a rule. (Just to be used, as for what I know, since I'll use it $f \circ \mathrm{id}_A \equiv f$ later)

- Conventions: . $A \to B \to C \equiv A \to (B \to C)$

- If $g : A \to B \to C$, then $g(x,y) \equiv g(x)(y)$, where $x:A$, $y:B$.

Definition 1.3.3 A function $P : A \to \mathcal{U}$, where $A : \mathcal{U}$, is called a <u>type family</u>, or a <u>predicate</u> [it corresponds to the homotopic notion of fibration, here $\mathcal{U}$ is "parametrized" on a space by A]. In set theory it corresponds to a family of sets].

If $u : P(x)$, for some $x : A$, then "P is true for x".



- One can introduce the judgement "$\Gamma, x : A \vdash P(x) : \mathcal{U}$" (and this is used in the definition of $\to$ in Π)

  P can be regarded as a family of types indexed over A (by x)

  Intensional

  That's why Martin-Löf's Type Theory is a <u>dependent type theory</u> (apart from simple types, types dependency is allowed !)

The right picture of $F : \prod_{x:A} P(x)$ is a <u>section</u> of the fibration $P : A \to \mathcal{U}$

• Section 14: Dependent function types

• Form $_{\prod_{x:A} P(x)}$

$$\frac{A : \mathcal{U} \qquad P : A \to \mathcal{U}}{\prod_{x:A} P(x) : \mathcal{U}}$$

• Intro $_{\prod_{x:A} P(x)}$

$$\frac{P : A \to \mathcal{U} \qquad (x : A \,,\, b(x) : P(x))}{[\lambda(x:A).\, b(x)] : \prod_{x:A} P(x)}$$

• Elim $_{\prod_{x:A} P(x)}$

$$\frac{F : \prod_{x:A} P(x) \,,\quad a : A}{F(a) : P(a)}$$

• Comp $_{\prod_{x:A} P(x)}$ →

$$\frac{P : A \to \mathcal{U} \,,\quad a : A \quad (x : A,\, b(x) : P(x))}{[\lambda(x:A).\, b(x)](a) \equiv b(a)}$$

where $b(a) \equiv b(a/x) : P(a)$

• Uniq $_{\prod_{x:A} P(x)}$ →

$$\frac{F : \prod_{x:A} P(x)}{F \equiv \lambda(x:A).\, F(x)}$$

• Go back to figure of p. 7, for the visualization of F (different colour?)

• Some people (Schreichtenberg) object to their use: "these are more formulas rather than types".

• Homotopically speaking: $F : \prod_{x:A} P(x)$ is a section of the fibration P.
interpretation

it captures the fact that some things happen "fiberwise", for each $P(x)$.

i.e., P is "fiberwise inhabited". (we'll see many examples of this)

• **Remark 14.1** If $P : A \to \mathcal{U}$ is $\overline{B}_A$ i.e. $P(x) \equiv B$, for every $x:A$, then

$$\prod_{x:A} P(x) \equiv \prod_{x:A} B \equiv A \to B$$

i.e., every function in $A \to B$ can be seen as a dependent function w.r.t. the constant type family B. (It will be used in the future).

No coincidence that the rules were so similar.

· **Remark 1.9.2** In the previous definition we used the $A \to B$ type constructor, in the form of $P : A \to \mathcal{U}$.

One can follow a general principle according to which each (new) type constructor should be introduced independently from all other type constructors, and define

$\prod_{x:A} P(x)$ independently from $A \to B$

eg $\dfrac{A : \mathcal{U} \qquad (x:A,\ B(x):\mathcal{U})}{\prod_{x:A} B(x) : \mathcal{U}}$ Form$_\Pi$

$\dfrac{x:A \qquad b(x) : B(x)}{[\lambda(x:A).b(x)] : \prod_{x:A} B(x)}$ intro$_\Pi$ (i.e., we $x \mapsto B(x)$ in the intuitive sense and not refer to $P : A \to \mathcal{U}$.

..., Then one can define $A \to B$ as a special case of $\prod_{x:A} B$.

But we find better for introducing to use both definitions and "in between" the aforementioned general principle (cf. in HoTT-book. More on the appendix of HoTT, p. 428).

· **Example:** $P : \mathcal{U} \to \mathcal{U}$ is defined by $P(A) \equiv A \to A$, for every $A : \mathcal{U}$

$$id_{\mathcal{U}} : \prod_{A:\mathcal{U}} (A \to A)$$

$$id_{\mathcal{U}}(A) \equiv id_A \quad , \text{ for any } A : \mathcal{U}$$

· **Remark:** Note that if $A : \mathcal{U}$ we cannot give any $F : \prod_{x:A} P(x)$, because we have not defined yet types $P(x)$ that depend on $x : A$.

Though the next introduced equality types $x =_A a$, $x =_A y$, we'll have many such P_i, therefore F_i, at hand.

· **Standard Convention of notation** $\prod_{x,y:A} \equiv \prod_{x:A} \prod_{y:A}$

(i.e. An empty (non-universe))

3. Let $A, B, C, \mathcal{U}$

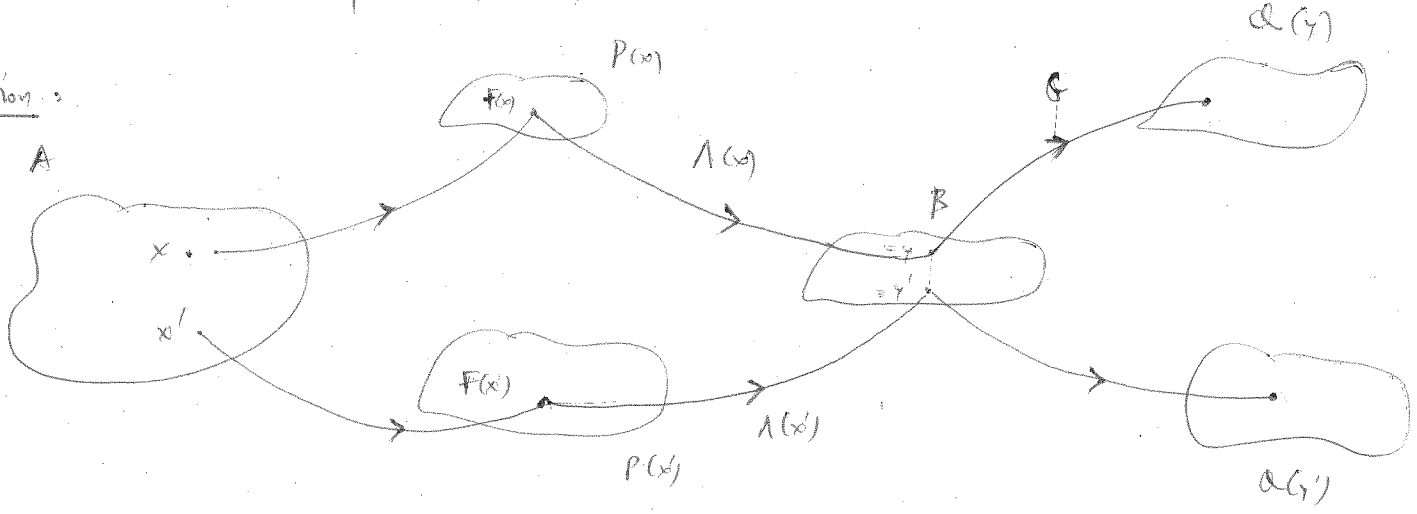
$P : A \to \mathcal{U}$ $\quad Q : B \to \mathcal{U}$

$F : \prod_{x:A} P(x)$ $\quad G : \prod_{y:B} Q(y)$

Given the above data, find formula and conditions, such that an operation $G \circ F : \prod_{x:A} (\ )$ can be defined to put after $P \equiv \overline{B}_A$ and $Q \equiv \overline{C}_B$, then

$G \circ F \equiv$ the composition of the functions $G \circ F : A \to C$, with those conditions.

Solution:



Let $\Lambda : \prod_{x:A} (P(x) \to B)$ i.e. $\Lambda(x) : P(x) \to B$, be given

$R : A \to \mathcal{U}$

$R(x) \equiv Q\left(\Lambda(x)\left(F(x)\right)\right)$

$\left(G \circ_\Lambda F\right)(x) \equiv G\left(\Lambda(x)\left(F(x)\right)\right)$

$\left(G \circ_\Lambda F\right) : \prod_{x:A} R(x)$

If $P \equiv \overline{B}_A$ and $Q \equiv \overline{C}_B$, then

$F : A \to B$ $\quad G : B \to C$

$\Lambda : \prod_{x:A} (B \to B) \equiv A \to (B \to B)$

define $\Lambda(x) \equiv id_B$, for every $x : A$.

then $G \circ_\Lambda F \equiv G \circ f$.

□

Examples / Exercises

1. Let the universes $\mathcal{U}$, $\mathcal{U}'$, and let $A : \mathcal{U}$.
Define a $P : A \to \mathcal{U}'$ and dependent function over A with the above data given (i.e. an element $\prod_{x:A} P(x)$)

Solution : $P : A \to \mathcal{U}'$

$P(x) \equiv \mathcal{U}$, for every $x : A$

We want $F : \prod_{x:A} P(x) \equiv \prod_{x:A} \mathcal{U} \equiv A \to \mathcal{U}$

$F(x) \equiv A$, for every $x : A$.

2. Let $A, B, C : \mathcal{U}$ $P : B \to \mathcal{U}$

$f : A \to B$ and $G : \prod_{y:B} P(y)$

With P from given above
Define an operation $\circ f$ $\prod_{x:A}$ for some $C : \mathcal{U}$ s.t. $G \circ f$ is related to standard composition functions where $P \equiv \bar{C}_B$.

Solution : Let $Q : A \to \mathcal{U}$ be defined by $Q(x) \equiv P(f(x))$, for every $x : A$

$G \circ f : \prod_{x:A} Q(x) \equiv \prod_{x:A} P(f(x))$ where

$(G \circ f)(x) \equiv G(f(x))$

If $P \equiv \bar{C}_B$, then $G : B \to C$ and $G \circ f$ is the standard composition of f.

• **Section 1.5: Equality types** (inductively defined) → All the rest types + type constructors are going to be defined inductively!!
IMPORTANCE: Grasp of the concept via the induction principle (corresp.) (like ℕ).
(INDUCTIVE MATHEMATICS — construction of concepts — Fully ... ℕ!!)

Relaxed/coarser notion of equality

• Form $x =_A y$ (better → $(=_A)$)

$$\frac{A : \mathcal{U}, \quad x, y : A}{x =_A y : \mathcal{U}}$$

$x =_{(A)} y$ is also called "identity" type ($Id_A(x,y)$), or "path" type ($Path_A(x,y)$), of identifications, or paths between x, y (in A). [Note that "equality of proofs" is rather mysterious]

• Intro $x =_A x$ (better → $(=_A)$)

$$refl_{(A)} : \prod_{x:A} (x =_A x)$$

, usually we write $refl$ only, when A is obvious

i.e., $refl(x) : x =_A x$

the standard notation is $refl_x$ for $refl(x)$

• The canonical element $refl_x$ of $x =_A x$ corresponds to the constant path.

• Without, or with UA one can show that there are paths, elements in $x =_A x$ which are not the $refl_x$, i.e., ITT can be interpreted as if all paths (loops) are refl. But our informal interpretation is that this is not the case in general.

• Ind$_{=_A}$ If $C : \prod_{x,y:A} \prod_{p : x =_A y} \mathcal{U}$ (or $C : \prod_{x,y:A} (x =_A y \to \mathcal{U})$)

is a dependent family of types in $\mathcal{U}$, i.e., $C(x, y, p) : \mathcal{U}$

and if

$$c : \prod_{x:A} C(x, x, refl_x)$$

, i.e., $c(x) : C(x, x, refl_x)$, for every $x:A$

is a dependent function, then there is a dependent function

$$F : \prod_{x,y:A} \prod_{p : x =_A y} C(x, y, p)$$

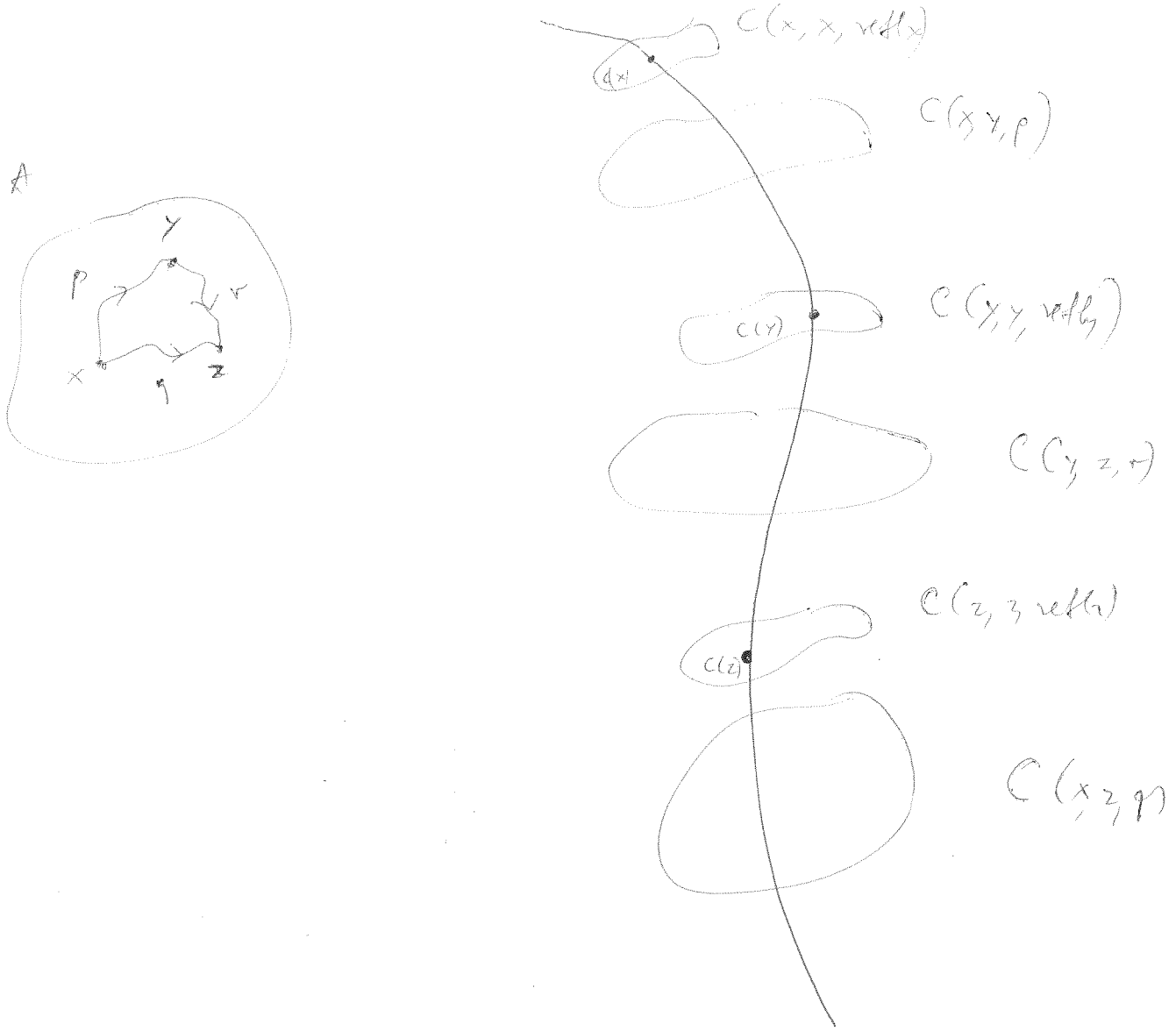
s.t.

$$F(x, x, refl_x) \equiv c(x)$$

, for every $x : A$.

• Ind$_{=_A}$ is also called (unbased) path-induction, or the J-rule. ✓✓ YZ 33–94

• it was introduced by Martin-Löf in 1973 (published 1975) An Int. theory of types, predicative part, Logic Colloquium 1973. (In the HoTT-book it is wrongly mentioned that it was introduced much later in (1979) — 25 years — me) ⇐ → CHECK THIS



$$I \to A \simeq \sum_{x,y:A} (x =_A y)$$

. Exercises let $A, B : \mathcal{U}$

1. If $f : A \to B$, there is $F : \prod_{x,y:A} ((x =_A y) \to B)$ s.t. $F(x,x,\mathrm{refl}_x) \equiv f(x)$.

Solution: Let $C : \prod_{x,y:A} \prod_{p : x =_A y} \mathcal{U}$ be defined by $C(x,y,p) \equiv B$, for every $x,y:A$ $p : x =_A y$

Let $c : \prod_{x:A} C(x,x,\mathrm{refl}_x) \equiv \prod_{x:A} B \equiv A \to B$ be defined by $c \equiv f$

By path-induction there is $F : \prod_{x,y:A} \prod_{p : x =_A y} B$ s.t. $F(x,x,\mathrm{refl}_x) \equiv c(x) \equiv f(x)$.

$A : \mathcal{U}$

Def: $R : A \to A \to \mathcal{U}$ $\quad R(x,y) : \mathcal{U}$ a "relation" on A

Let R be reflexive i.e., $R(x,x)$ is inhabited ("true") for every $x : A$

i.e., there is $\quad r : \prod_{x:A} R(x,x) \qquad r(x) : R(x,x)$

Define $\quad C(x,y,p) \equiv R(x,y) : \mathcal{U}$, for every $p : x =_A y$.

i.e., $\quad C : \prod_{x,y:A} \prod_{p : x =_A y} \mathcal{U}$ is independent from $p : x =_A y$

and $\quad C(x,x,\mathrm{refl}_x) \equiv R(x,x)$, hence

$$r : \prod_{x:A} C(x,x,\mathrm{refl}_x)$$

By path-induction there is $\quad f : \prod_{x,y:A} \prod_{p : x =_A y} R(x,y)$ s.t.

$$f(x,x,\mathrm{refl}_x) \equiv r(x)$$

i.e., "if R is reflexive, then $R(x,y)$ holds for all $x,y : A$ s.t. $x =_A y$"

i.e., "$=_A$ is the least reflexive relation on A" $\quad$ i.e. $\prod_{x,y:A} (x =_A y \to R(x,y))$ is inhabited

, since $\quad f(x,y,p) : R(x,y)$, for any $x,y : A$ and $p : x =_A y$.

R is rel on A : $\equiv R : A \to A \to \mathcal{U}$

Def $\quad R$ is reflexive $:\equiv \prod_{x:A} R(x,x)$ is inhabited

is symmetric $:\equiv \prod_{x,y:A} (R(x,y) \to R(y,x))$ is inh.

, transitive $:\equiv \prod_{x,y,z:A} (R(x,y) \to R(y,z) \to R(x,z))$ is inhabited

$=_A : A \to A \to \mathcal{U}$

$=_A(x,y) \equiv x =_A y$

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

Clearly if not $R(x,y)$ then $R(x,y) \equiv 0$

empty type

ex: when 0 →

Chapter 2

Basic applications of path-induction

In this chapter we present applications of path-induction. The general "technique" of applying path-induction in the proof of a proposition is the following:

Step 1: Create a type $\mathcal{C}(x, y, p)$ out of the given data $A : U$, $x, y : A$, $p : x =_A y$, and the property to be proven i.e., translate what is to be proven into a type $\mathcal{C}(x, y, p)$.

Step 2: Define the typed family $\mathcal{C} : \prod_{x,y:A} \prod_{p:x=_A y} \mathcal{U}$ through Step 1.

Step 3: Define $\varsigma : \prod_{x:A} \mathcal{C}(x, x, \mathrm{refl}_x)$ by unfolding $\mathcal{C}(x, x, \mathrm{refl}_x)$. Most of the times the definition of $\varsigma(x)$ is straightforward after the unfolding.

Step 4: Use the conclusion of path-induction on the previously defined $\mathcal{C}$ and ς to get the object that proves the proposition.

2.1 Uniqueness

Proposition 2.1.1. *Let $\mathcal{C} : \prod_{x,y:A} \prod_{p:x=_A y} \mathcal{U}$ be a dependent family of types in $\mathcal{U}$, and $\varsigma : \prod_{x:A} \mathcal{C}(x, x, \mathrm{refl}_x)$ be a dependent function. If $F, G : \prod_{x,y:A} \prod_{p:x=_A y} \mathcal{C}(x, y, p)$ are dependent functions such that*

$$F(x, x, \mathrm{refl}_x) :\equiv \varsigma(x) \equiv: G(x, x, \mathrm{refl}_x),$$

for every $x : A$, then the type

$$\prod_{x,y:A} \prod_{p:x=_A y} F(x, y, p) =_{\mathcal{C}(x,y,p)} G(x, y, p)$$

is inhabited.

Proof. We define the dependent family of types $\mathcal{D} : \prod_{x,y:A} \prod_{p:x=_A y} \mathcal{U}$ by

$$\mathcal{D}(x, y, p) :\equiv F(x, y, p) =_{\mathcal{C}(x,y,p)} G(x, y, p).$$

To define a dependent function $\mathfrak{q} : \prod_{x:A} \mathcal{D}(x, x, \mathrm{refl}_x)$ we need to have

$$\mathfrak{q}(x) : \mathcal{D}(x, x, \mathrm{refl}_x) \equiv F(x, x, \mathrm{refl}_x) =_{\mathcal{C}(x,x,\mathrm{refl}_x)} G(x, x, \mathrm{refl}_x) \equiv \varsigma(x) =_{\mathcal{C}(x,x,\mathrm{refl}_x)} \varsigma(x).$$

Thus we define

$$\mathfrak{q}(x) :\equiv \mathrm{refl}_{\varsigma(x)}.$$

- Ind$_{=_A}$ is the most fundamental (and powerful) proof-tool in HTT with amazing consequences

- Ind$_{=_A}$ packaged into a single function:

$$\mathrm{Ind}_{=_A} : \prod_{C : \prod_{x,y:A} \prod_{p : x =_A y} \mathcal{U}} \; \prod_{c : \prod_{x:A} C(x,x,\mathrm{refl}_x)} \left(\prod_{x,y:A} \prod_{p : x =_A y} C(x,y,p) \right) \qquad (1)$$

$$\simeq \; \mathrm{Ind}(C,c) : \prod_{x,y:A} \prod_{p : x =_A y} C(x,y,p)$$

s.t. $$\mathrm{Ind}_{=_A}(C,c,x,x,\mathrm{refl}_x) \equiv c(x) \qquad (2)$$

The function Ind$_{=_A}$ is traditionally called J. ($\simeq$, $J(C,c)$)

Remark 1.5.1: (1) can be understood as $=_A$-Elim ($x =_A x$ is "eliminated," and F is constructed)

(2) " " $=_A$-Comp (how to eliminate Ind$_{=_A}$ as a function in applied to the input-rule, $x \mapsto \mathrm{refl}_x$)

The same pattern is followed in all the rest inductive definitions: the Ind$_{\mathrm{Type}}$ which generally expresses the way to define elements of type $\prod_{x:\mathrm{Type}} P(x)$, can be splitted into the

- Ind$_{\mathrm{Type}}$ = packaging function
- computation rule

Recall that uniqueness principles are optional.

So all Types introduced will follow the following pattern of rules:

- Form$_{\mathrm{Type}}$
- Intro$_{\mathrm{Type}}$
- Elim$_{\mathrm{Type}}$
- Comp$_{\mathrm{Type}}$
- (Uniq$_{\mathrm{Type}}$)

(Ind$_{\mathrm{Type}}$ may Types inductive)

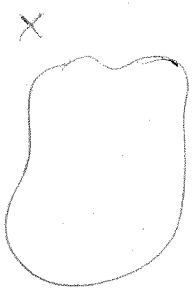
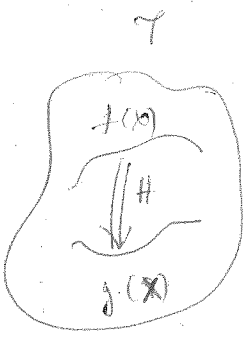
- Interpretations of Path-induction: (a) Homotopic (in HoTT-book) → Through Voevodsky's notion (by Capriotti 2017)] related investigations
  (b) Categorical (via Yoneda lemma: § 5.8 in HoTT-book, p. 172)
  (c) Type-Theoretic // via the general theory of inductive definitions (can)
  (James Ladyman + Stuart Presnell: Identity in HoTT, part I: The justification of Path Induction 2014, see also 2016-paper of them; unfortunately foundation for mathematics / in use of exotic not-universal terms like homotopy) (we'll discuss this later!)

(d)? Topological ??

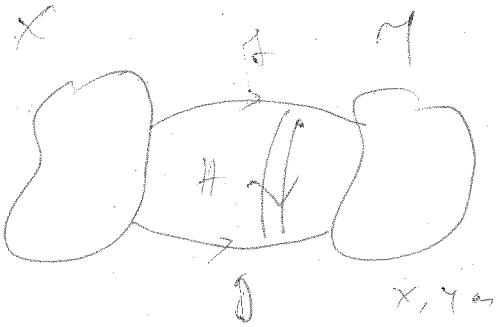
Remark 1.5.2 : The homotopic interpretation of Thm

"How can $T(x,y,p)$ depend only on $T(x,x,\text{refl}_x)$, when in principle there are more paths between x and x than refl_x ?" The hard interpretation is that

- $X, Y, \in \text{Top}$
- $f, g : X \xrightarrow{\text{cont}} Y$ the "standard notation" !
- $H : f \sim g$ H is a homotopy between f, g
  if $H : X \times [0,1] \xrightarrow{\text{cont}} Y$ s.t.
  $H(x,0) = f(x)$, $\forall x \in X$
  $H(x,1) = g(x)$, $\forall x \in X$

or better



X, Y = points
f, g = paths

- $x =_A y$ the "set" of paths from x to y.
- $x =_A x$ is the set of loops at x
- refl_x = the constant loop

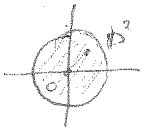
Later we'll define a type $\|A\|_0$ and we'll show $\left(x =_{\|A\|_0} y\right) \simeq \sum_{\varphi : Z \to A} \| \varphi(z) = x \times \varphi(z) = y \|$
but yet the "correct" picture of a path for an element z is

- $X \sim Y$ (homotopy equivalent) if there are $f : X \to Y$ and $g : Y \to X$ s.t.
  $(g \circ f) \sim id_X$ and $(f \circ g) \sim id_Y$
- $\sim$: equal
- $[X]_\sim = \{Y \in \text{Top} \mid Y \sim X\}$: the homotopy type of X
- Homotopy theory doesn't distinguish between homotopy equivalent spaces, as topology doesn't distinguish between homeomorphic spaces, and hence homotopy types rather than top. spaces are the basic objects of homotopy theory.

Ex. $\mathbb{D}^2 = \{(x,y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$ not homeomorphic to $\{(0,0)\}$ (as bijection from $\mathbb{D}^2$ to $\mathbb{O}$, but top. spaces)

$\mathbb{D}^2 \sim \{(0,0)\}$: $f : \mathbb{D}^2 \to \mathbb{O}$ $f(x,y) = (0,0)$, $\forall (x,y) \in \mathbb{D}$

$g : \mathbb{O} \to \mathbb{D}^2$ $g((0,0)) = (0,0)$, or any other point in $\mathbb{D}$



$f \circ g\,(0,0) = (0,0) = id_{\mathbb{O}}(0,0)$ $(g \circ f)(x,y) = (0,0)$, $\forall x,y$ $\Rightarrow$ f, g are continuous

$H : (g \circ f) \sim id_{\mathbb{D}^2}$, $H : \mathbb{D}^2 \times [0,1] \to \mathbb{D}^2$ s.t. $H(z,0) = (g \circ f)(z) = (0,0)$
$H(z,1) = id_{\mathbb{D}^2}(z) = z$

Define $H(z,t) = tz$, if $z = (x,y)$ — two (x,y) s.t.

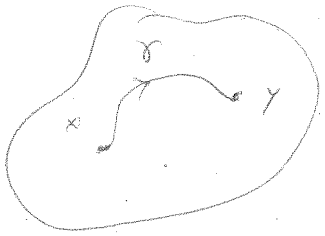
we need $H' : \mathbb{D} \times [0,1] \to \mathbb{D}$
$H'((0,0),0) = (f \circ g)(0,0) = (0,0)$
$H'((0,0),1) = id_{\mathbb{D}}((0,0)) = (0,0)$
hence $H'((0,0),t) = (0,0)$, $\forall t$.

$H' : (f \circ g) \sim id_{\mathbb{O}}$,

$H' : \mathbb{D}^2 \times [0,1] \to \mathbb{D}^2$ s.t. $H(z,0) = (f \circ g)(z) = (0,0)$
$H(z,1) = id_{\mathbb{O}}(z) = (0,0)$
Define $H(z,t) = (0,0)$ $\forall z,t$

- Let $x, y : X$, $\gamma : [0,1] \xrightarrow{\text{cont}} X$ with $\gamma(0) = x$, $\gamma(1) = y$

$\overline{x}_{[0,1]} : [0,1] \to X$ with $\overline{x}_{[0,1]}(t) = x$, $\forall t \in [0,1]$



Then is $H : \gamma \sim \overline{x}_{[0,1]}$

We define $H : [0,1] \times [0,1] \xrightarrow{\text{cont}} X$ s.t. $H(s,0) = \gamma(s)$, $\forall s \in [0,1]$ and

$H(s,1) = x = \overline{x}_{[0,1]}(s)$, $\forall s \in [0,1]$.

Let $H(s,t) := \left(1 - t\right) \gamma(s) + t\, \overline{x}_{[0,1]}(s)$

$H(s,0) = \gamma(s) + 0 = \gamma(s)$, $H(s,1) = 0 + \overline{x}_{[0,1]}(s) = x$.

If we want to define $H' : \overline{x}_{[0,1]} \sim \gamma$ we $H'(s,t) = \gamma(s \cdot t)$

$H'(s,0) = \gamma(0) = x$, $H'(s,1) = \gamma(s)$.

- Hence: • Since $\gamma \sim \overline{x}_{[0,1]}$, a property of $\overline{x}_{[0,1]}$ that respects homotopy (i.e. H is invariant under homotopy) will hold for γ. Thus to show that γ has such a property it suffices to show it for $\overline{x}_{[0,1]}$ (This is the homotopy-theoretic counterpart to Ind$_{\mathbb{N}}$)!!
  - If γ can be retracted to $\overline{x}_{[0,1]}$ (also in figure, "shrinked to")

- Remark 1.5.3: If one adds the following reflection rule:

$$\frac{p : x =_A y}{x =_A y}$$

then one gets extensional TT out of ITT.
(ETT)
Of course, we don't accept to have this rule

- Remark 1.5.4: Later will define the HIT interval I and will show that

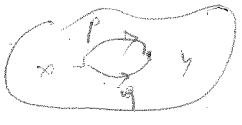
$$x =_A y \simeq \sum_{f : I \to A} \left(f(0_I) =_A x\right) \times \left(f(1_I) = y\right)$$

and we regain the usual topological picture of path on $\gamma : [0,1] \to A$.

- Streicher Axiom K:

$$\frac{p : x =_A x}{p = \mathrm{refl}_x}$$

$x \Rightarrow p = q$, for $p, q : x =_A y$



$p \cdot q^{-1}$ reduces to $p = q$

Section 5.6: $\mathrm{Ind}_=$? ⇒ Recall path-induction

there is $f : \prod_{x:A} \prod_{p:x=_A x} \dots$

(Exercise from HoTT-book)

$f(x, \mathrm{refl}_x) \equiv \mathrm{refl}_{\mathrm{refl}_x}$

Remark 4.6.1: $\left(\dfrac{p : x =_A x}{p = \mathrm{refl}_x}\right)$ vs. (2a) $\left(\dfrac{p : x =_A x}{p = \mathrm{refl}_x}\right)$ $\mathrm{Str}(x)$

- Attempt with $\mathrm{Ind}_=$ elements: $C(x, y, p)$, no explicable
- This corresponds to the principle:

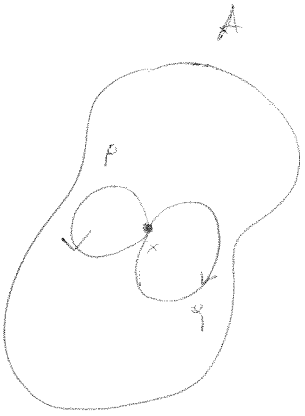
$\mathrm{Ind}_='$ {

$C : \prod_{x:A} \prod_{p:x=_A x} \mathcal{U}$ $\quad C(x,p) : \mathcal{U}$

$c : \prod_{x:A} C(x, \mathrm{refl}_x)$ $\quad c(x) : C(x, \mathrm{refl}_x)$

Then there is $F : \prod_{x:A} \prod_{p:x=_A x} C(x,p)$ vs.

$F(x, \mathrm{refl}_x) \equiv c(x)$ }



They define $C(x,p) \equiv (p = \mathrm{refl}_x)$

$C(x, \mathrm{refl}_x) \equiv (\mathrm{refl}_x = \mathrm{refl}_x)$

$c(x) \equiv \mathrm{refl}_{\mathrm{refl}_x} : (\mathrm{refl}_x = \mathrm{refl}_x)$

and then we expect $f \equiv \mathrm{refl}_{\mathrm{refl}}$ to follow.

{ W.r.t. the homotopic interpretation the loop p cannot be shrunked to the constant path. "We cannot cut it", starting from a point in the loop, and "go to x". }

($S^1 \times \Psi$ arises in S^1 a lot.)

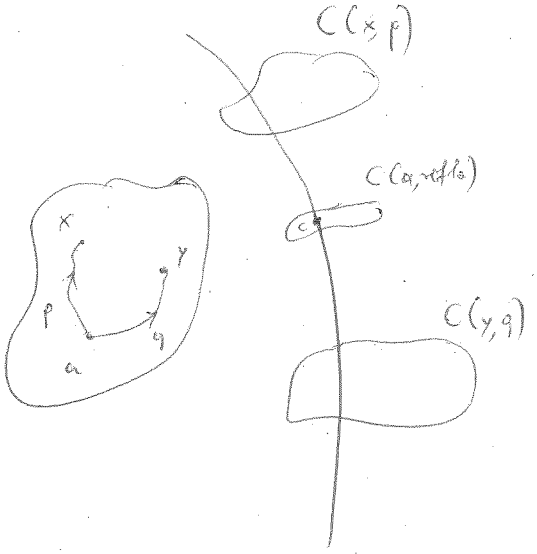
⇒ there is no homotopical reason to accept $\mathrm{Ind}_='$ (it seems from this that dependent K is incompatible to the homotopy-interpretation).

- We'll see that there is a theorem in HoTT (!) which says $\neg(\Rightarrow \mathrm{Ind}_=')$ always holds.
- Also we can actually say towards this direction in the following from a p-induction.

(Paulin-Mohring 1993) **Based path-induction**: $A:\mathcal{U}$ $a:A$

If $C : \prod_{x:A} \prod_{p:a=_A x} \mathcal{U}$ i.e. $C(x,p):\mathcal{U}$

and $c : C(a, \mathrm{refl}_a)$,



then there is $f : \prod_{x:A} \prod_{p:a=_A x} C(x,p)$ s.t.

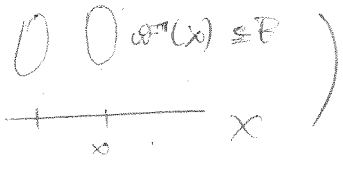
$$f(a, \mathrm{refl}_a) \equiv c$$

(If is very useful, sometimes more than Ind$_=$, they are provably equivalent).

- **Homotopical interpretation of b-Ind$_=$**: again the path $p : a =_A x$ is retracted (continuously) to the constant path refl_a.

Fibrations

- pInd$_=$ is the type-theoretic version of **Serre's theorem** (Cagnard 2011): a space X has to have a suitable fiber F over X. (fiber (E, π, X) $\pi : E \xrightarrow[\text{cont}]{\text{surjective}} X$ $\pi^{-1}(x) \simeq F$)



loop-space approach to alg. top. (~1950)

Serre defined $E_a = \{ p : I \to X \mid p(0) = a,\ p(1) = x \}$

$\pi(p) = p(1) = x$.

Recapitulation 6.2.:

- b-Ind$_=$ $\Rightarrow$ b-Transport : where b-Transport is $P : A \to \mathcal{U}$ there is $p^P_* : P(a) \to P(x)$ s.t. $(\mathrm{refl}_a)^P_* \equiv \mathrm{id}_{P(a)}$

(or an ∞?)

Proof: define $C(x,p) \equiv P(a) \to P(x)$

$C(a, \mathrm{refl}_a) \equiv P(a) \to P(a)$

Take $c \equiv \mathrm{id}_{P(a)} : C(a, \mathrm{refl}_a)$

By b-Ind$_=$ there is $F : \prod_{x:A} \prod_{p:a=x} P(a) \to P(x)$ s.t. $F(a, \mathrm{refl}_a) \equiv c$

define $p^P_* \equiv F(x,p)$.

Proposition 1.6.3. (on an Ex 2). $\flat$-Ind$_=$ $\Rightarrow$ $\flat$-Unique: i.e. $\forall F, G : \prod_{x:A} \prod_{p:a=_A x} (\dots)$ s.t.

$$\mathrm{ol} : F(a, \mathrm{refl}_a) = G(a, \mathrm{refl}_a)$$

where $C : \prod_{x:A} \prod_{p:a=_A x} \mathcal{U}$

Then $F(x,p) = G(x,p)$, for every x, p.

Proof:

$$D(x,p) \equiv F(x,p) = G(x,p)$$

$$D(a, \mathrm{refl}_a) \equiv F(a, \mathrm{refl}_a) = G(a, \mathrm{refl}_a)$$

$$\mathrm{ol} : D(a, \mathrm{refl}_a)$$

By $\flat$-Ind$_=$: $\quad \mathcal{I} : \prod_{x:A} \prod_{p:a=_A x} \left(F(x,p) = G(x,p) \right)$ s.t.

$$\mathcal{I}(a, \mathrm{refl}_a) \equiv \mathrm{ol}$$

Remark 1.6.4: One could read the above results as a form of "density" of $\{c\}$ in the set of $\{(x,p) \mid p : a =_A x\}$, since the "continuous" $F(x,p)$ are determined by $\{c\}$ (existence by $\flat$-Ind, & the extension of the constant map c, and uniqueness by the density). But this topological interpretation doesn't correspond to some actual set-theoretic translation of types as top. spaces: If $\{c\}$ is dense, then $\{c\}$ is not closed (if it was, then $\{c\}$ should intersect, as dense, its open complement) i.e. the top. on S is not T_1, and classically the intensity extension theorems (Tsirelson for comp. T_2, and Beth-Mostowski for real-comp) require the space to be T_1, so that $f : D \xrightarrow{\text{cont}} Y$ has a unique cont ext on $X \to Y$!!

I.e., we cannot justify top/by $\flat$-path-induction above. Thus only homotopical integers is needed, which is more than topological. —

Proposition 1.6.5: $\boxed{\flat\text{-Ind}_= \Rightarrow \text{Ind}_=}$ (the easy direction).

Proof: Let $C : \prod_{x,y:A} \prod_{p:x=_A y} \mathcal{U}$ and $c : \prod_{x:A} C(x,x,\mathrm{refl}_x)$ be given.

(x:A)

Let $C_x : \prod_{y:A} \prod_{p: x =_A y} \mathcal{U}$ be defined by $C_x(y, p) \equiv C(x, y, p)$

We want $c_x : C_x(x, \mathrm{refl}_x) \equiv C(x, x, \mathrm{refl}_x)$

Define $c_x \equiv c(x)$

By b-Ind$_=$ there is $F_x : \prod_{y:A} \prod_{p: x =_A y} C_x(y, p) \equiv \prod_{y:A} \prod_{p: x =_A y} C(x, y, p)$

s.t. $F_x(x, \mathrm{refl}_x) \equiv c_x \equiv c(x)$

We define $F : \prod_{x:A} \prod_{y:A} \prod_{p: x =_A y} C(x, y, p)$ by

$$F \equiv \lambda(x : A).\, F_x \quad : \prod_{x,y:A} \prod_{p: x =_A y} C(x, y, p)$$

$$F(x, x, \mathrm{refl}_x) \equiv F_x(x, \mathrm{refl}_x) \equiv c(x), \quad \text{for every } x : A.$$

□

<u>Theorem 1.6.6</u> ~~(Altenkirch + Coquand)~~, other proof by Streicher (+ K-axiom, 93) Hofmann: (without universes, see Ex 1.7 in HoTT)

Ind$_=$ ⟹ b-Ind$_=$

<u>Proof (Altenkirch and Coquand)</u>: Let $C' : \prod_{x:A} \prod_{p: a =_A x} \mathcal{U}$ and $c' : C(a, \mathrm{refl}_a)$ be given.

We define $D : \prod_{x,y:A} \prod_{p: x =_A y} \mathcal{U}$ by

$$D(x, y, p) \equiv \prod_{C : \prod_{z:A} \prod_{v: x =_A z} \mathcal{U}} \Big(C(x, \mathrm{refl}_x) \to C(y, p) \Big) \qquad (1)$$

since $C : \prod_{z:A} \prod_{r : x =_A z} \mathcal{U}$ is $C(x, \mathrm{refl}_x)$ and $C(y, p)$ are well-defined

By (1) we get

$$D(x, x, \mathrm{refl}_x) \equiv \prod_{C : \prod_{z:A} \prod_{r : x =_A z} \mathcal{U}} \left(C(x, \mathrm{refl}_x) \to C(x, \mathrm{refl}_x) \right)$$

We define

$$d : \prod_{x:A} D(x, x, \mathrm{refl}_x) \quad \text{by}$$

$$d(x) \equiv \lambda \left(C : \prod_{z:A} \prod_{r : x =_A z} \mathcal{U} \right) . \, id_{C(x, \mathrm{refl}_x)} \qquad (2)$$

$$\left(id : C \overset{d(x)}{\mapsto} id_{C(x, \mathrm{refl}_x)} \right)$$

By $\mathrm{Ind}_=$ there is $F : \prod_{x,y:A} \prod_{p : x =_A y} D(x, y, p)$ s.t.

$$F(x, x, \mathrm{refl}_x) \equiv d(x)$$

i.e. $F : \prod_{x,y:A} \prod_{p : x =_A y} \prod_{C : \prod_{z:A} \prod_{r : x =_A z} \mathcal{U}} \left(C(x, \mathrm{refl}_x) \to C(y, p) \right)$

Hence $F(a, x, p) : \prod_{C : \prod_{z:A} \prod_{r : a =_A z} \mathcal{U}} \left(C(a, \mathrm{refl}_a) \to C(x, p) \right)$

$p : a =_A x$

Hence $F(a, x, p, C', c') : C'(x, p)$

If we define $F' \equiv \lambda (x : A, p : a =_A x) . \, F(a, x, p, C', c')$ i.e.

$$F' : \prod_{x:A} \prod_{p:a=_A x} C'(x,p) \quad \text{s.t.}$$

$$\begin{aligned} F'(a, \mathrm{refl}_a) &\equiv F(a, a, \mathrm{refl}_a, C', c') \\ &\equiv F(a, a, \mathrm{refl}_a)(C', c') \\ &\equiv d(a)(C', c') \\ &\equiv d(a)(C')(c') \\ &\equiv \mathrm{id}_{C'(a,\mathrm{refl}_a)}(c'), \qquad c' : C'(a, \mathrm{refl}_a) \\ &\equiv c' \end{aligned}$$

i.e. F' is the required inhabitant for b-Ind$_=$. $\Box$

• It will be used in the determination of path-spaces, e.g., for f-contractg + in many other occasions. (See also proof of uniqueness principle for $=$).