Homotopy type theory

Simon Huber

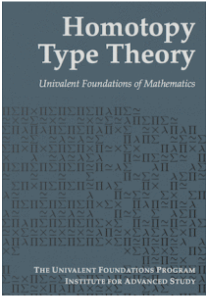
University of Gothenburg

Summer School on Types, Sets and Constructions,
Hausdorff Research Institute for Mathematics,
Bonn, 3–9 May, 2018

Overview

Part I. **Homotopy type theory**

Main reference: HoTT Book
`homotopytypetheory.org/book/`



Part II. **Cubical type theory**

Lecture I: Equality, equality, equality!

1. Intensional Martin-Löf type theory
2. Homotopy interpretation
3. H-levels
4. Univalence axiom

Some milestones

1970/80s Martin-Löf's intensional type theory

1994 Groupoid interpretation by Hofmann&Streicher

2006 Awodey/Warren: interpretation of Id in Quillen model categories

2006 Voevodsky's note on homotopy λ-calculus

2009 Lumsdaine and van den Berg/Garner: types are ∞-groupoids

2009 Voevodsky's univalent foundations

- h-level, equivalence, univalence
- model of MLTT in simplicial sets

2012/13 IAS Special Year on UF

M. Hofmann (1965–2018)

V.Voevodsky (1966–2017)

Intensional Martin-Löf type theory

- Dependently type λ-calculus with
  - dependent products Π and dependent sums Σ
  - data types N_0 (empty type), N_1 (unit type), N_2 (booleans), N (natural numbers), ...
  - intensional Martin-Löf identity type Id
  - universes $\mathsf{U}_0 : \mathsf{U}_1, \mathsf{U}_1 : \mathsf{U}_2, \mathsf{U}_2 : \mathsf{U}_3, \ldots$
- No axioms (for now), all constants are explained computationally

Two notions of "sameness" in type theory

Judgmental equality

$u \equiv v : A$ and $A \equiv B$

- judgment of type theory
- definitional equality, unfolding definitions
- In Coq/Agda: no direct access, used for *computation*

vs.

Identity types

$\mathsf{Id}_A(u, v)$

- a type
- "propositional" equality
- can appear in assumptions/context

Identity types (Martin-Löf)

Formation and introduction rule:

$$\frac{\vdash A \qquad u : A \qquad v : A}{\vdash \mathsf{Id}_A(u, v)} \qquad\qquad \frac{u : A}{\mathsf{refl}\, u : \mathsf{Id}_A(u, u)}$$

Identity induction:

$$\frac{\begin{array}{c}\vdash A \qquad x : A, y : A, z : \mathsf{Id}_A(x, y) \vdash C(x, y, z) \\ d : \Pi(x : A).C(x, x, \mathsf{refl}\, x) \\ u : A \qquad v : A \qquad p : \mathsf{Id}_A(u, v)\end{array}}{\mathsf{J}\, d\, u\, v\, p : C(u, v, p)}$$

Identity types (Martin-Löf)

Formation and introduction rule:

$$\frac{\vdash A \qquad u : A \qquad v : A}{\vdash \mathsf{Id}_A(u, v)} \qquad\qquad \frac{u : A}{\mathsf{refl}\, u : \mathsf{Id}_A(u, u)}$$

Identity induction:

$$\frac{\begin{array}{c}\vdash A \qquad x : A, y : A, z : \mathsf{Id}_A(x, y) \vdash C(x, y, z) \\ d : \Pi(x : A).C(x, x, \mathsf{refl}\, x) \\ u : A \qquad v : A \qquad p : \mathsf{Id}_A(u, v)\end{array}}{\mathsf{J}\, d\, u\, v\, p : C(u, v, p)}$$

Definitional equality: $\mathsf{J}\, d\, x\, x\, (\mathsf{refl}\, x) \equiv d\, x : C(x, x, \mathsf{refl}\, x)$

Based identity induction (Paulin-Mohring)

Fix a type A and $a : A$.

$$\frac{x : A, z : \mathsf{Id}_A(a, x) \vdash C(x, z) \\ e : C(a, \mathsf{refl}\, a) \qquad u : A \qquad p : \mathsf{Id}_A(a, u)}{\mathsf{J}'\, e\, u\, p : C(u, p)}$$

Definitional equality: $\mathsf{J}'\, e\, a\, (\mathsf{refl}\, a) \equiv e : C(a, \mathsf{refl}\, a)$

Equivalent to identity induction.

Based identity induction (Paulin-Mohring)

Fix a type A and $a : A$.

$$\frac{\begin{array}{c} x : A, z : \mathsf{Id}_A(a, x) \vdash C(x, z) \\ e : C(a, \mathsf{refl}\, a) \qquad u : A \qquad p : \mathsf{Id}_A(a, u) \end{array}}{\mathsf{J}'\, e\, u\, p : C(u, p)}$$

Definitional equality: $\mathsf{J}'\, e\, a\, (\mathsf{refl}\, a) \equiv e : C(a, \mathsf{refl}\, a)$

Equivalent to identity induction.

We also write $x =_A y$ for $\mathsf{Id}_A(x, y)$.

Special case: transport

Given $x : A \vdash B(x)$ we get

$$\mathsf{transport}^{x.B} : \Pi(x\ y : A).\, x = y \to B(x) \to B(y)$$

with

$$\mathsf{transport}\,(\mathsf{refl}\,x)\,u \equiv u : B(x).$$

Leibniz' *indiscernibility of identicals*: "if x is identical to y, then x and y have all the same properties"

The identity type is an equivalence relation:

1. $\mathsf{refl}\, x : x = x$
2. if $p : x = y$, then $p^{-1} : y = x$
3. if $p : x = y$ and $q : y = z$, then $p \cdot q : x = z$

Moreover: $(\mathsf{refl}\, x)^{-1} \equiv \mathsf{refl}\, x$ and $(\mathsf{refl}\, x) \cdot q \equiv q$

Congruence

For $f\colon A \to B$ and $p : x =_A y$ have

$$\mathsf{ap}\, f\, p : f\, x =_B f\, y$$

1. $\mathsf{ap}\, f\, (\mathsf{refl}\, x) \equiv \mathsf{refl}(f\, x)$
2. $\mathsf{ap}\, (f \circ g)\, p = \mathsf{ap}\, f\, (\mathsf{ap}\, g\, p)$
3. $\mathsf{ap}\, \mathrm{id}\, p = p$

Function extensionality?

In mathematics we often want to identify two functions whenever they are pointwise equal. In type theory this can be formulated as:

$$\Pi(A : \mathsf{U})(B : A \to \mathsf{U})(f\ g : \Pi(x : A).B\,x).$$
$$(\Pi(x : A).f\,x =_{B\,x} g\,x) \to f = g$$

A principle of modularity!

However, this is not derivable and has to be assumed as an axiom.

Voevodsky: function extensionality follows from univalence axiom!

(In **BISH** one works with setoids instead.)

Function extensionality?

Intensional MLTT without function extensionality violates the principle (Russel&Whitehead, PM 2nd ed, 1925) that

> *[..] a function can only enter into a proposition through its values.*

In MLTT

$$\vdash C(f) \text{ true} \qquad \text{and} \qquad x : A \vdash f\,x =_B g\,x \text{ true}$$

do in general **not** entail

$$\vdash C(g) \text{ true}.$$

(Take $f :\equiv \lambda x.x$, $C(z) :\equiv f =_{\mathsf{N}\to\mathsf{N}} z$, and $g :\equiv \lambda x.\,\mathsf{S}^x\,0$.)

Structure vs. property?

Using universes and Σ-types we can conveniently encode the type of types with binary operation as:

$$\mathsf{BinOp}(A) :\equiv A \times A \to A \qquad \mathsf{Bin} :\equiv \Sigma(A : \mathsf{U}).\, \mathsf{BinOp}(A)$$

For $(A, m) : \mathsf{Bin}$ we can express commutativity of m by:

$$\mathsf{Law}(A, m) :\equiv \Pi(x\ y : A).\, m(x, y) =_A m(y, x)$$

and $\mathsf{CBin} :\equiv \Sigma(A : \mathsf{U})\Sigma(m : \mathsf{BinOp}(A)).\, \mathsf{Law}(A, m)$.
Proof of commutativity now part of the data: $(A, m, p) : \mathsf{CBin}$
A priori (A, m, p) and (A, m, p') are different things!

Cure: setoids (Bishop)?

Structure of Id?

We can iterate the identity type!

$$u =_A v \qquad p =_{u=_A v} q \qquad \alpha =_{p=_{u=_A v} q} \beta \qquad \ldots$$

What is this structure and is it interesting?

Uniqueness of Identity Proofs (UIP)?

Does this hierarchy collapse? Are all $p =_{u=_A v} q$ are inhabited?

$$\mathsf{UIP} :\equiv \Pi(A : \mathsf{U})\Pi(x\ y : A)(p\ q : x =_A y).\, p = q$$

In the set-theoretic model of type theory UIP holds.

Structure of Id?

We can iterate the identity type!

$$u =_A v \qquad p =_{u =_A v} q \qquad \alpha =_{p =_{u =_A v} q} \beta \qquad \ldots$$

What is this structure and is it interesting?

Uniqueness of Identity Proofs (UIP)?

Does this hierarchy collapse? Are all $p =_{u =_A v} q$ are inhabited?

$$\mathsf{UIP} :\equiv \Pi(A : \mathsf{U})\Pi(x\ y : A)(p\ q : x =_A y).\, p = q$$

In the set-theoretic model of type theory UIP holds.

Answer here: understand this structure via *homotopy theory!*

Groupoid model (Hofmann/Streicher 1994)

Hofmann and Streicher note that Id-types satisfy the groupoid laws *up to an* Id-*equality*. Write $1_x :\equiv \mathsf{refl}\, x$ and let $p : x =_A y$.

- $p \cdot 1_y = p$ and $1_x \cdot p = p$
- $p \cdot p^{-1} = 1_x$ and $p^{-1} \cdot p = 1_y$
- $(p \cdot q) \cdot r = p \cdot (q \cdot r)$

Groupoid model

Each (closed) type A interpreted as a *groupoid* and $\mathsf{Id}_A(a, b)$ has as objects the morphisms $a \to b$ in A

$\mathbb{Z}/2\mathbb{Z}$ considered as a groupoid gives counter-example to UIP!

A predecessor of the homotopy interpretation of identity types.

Types are weak ∞-groupoids ($\sim$2009)

A generalization of the observation that types induce groupoids (up to paths).

Lumsdaine and van den Berg/Garner: the iterated Id-types give rise to the structure of ∞-groupoids!

A	$u =_A v$	$p =_{u =_A v} q$	$\alpha =_{p =_{u =_A v} q} \beta$	$\dots$
0-cells	1-cells	2-cells	3-cells	$\dots$

Classical homotopy theory

- Let $\mathbb{I} := [0,1]$ and $f, g \colon X \to Y$ be two continuous maps between topological spaces X and Y. A *homotopy* between f and g is a continuous map $H \colon X \times [0,1] \to Y$ with:

$$\begin{cases} H(x,0) = f(x) \\ H(x,1) = g(x) \end{cases}$$

  Write $f \simeq_H g$.
- $X \simeq Y$ (for spaces X and Y) if we have $f \colon X \to Y$ and $g \colon Y \to X$ such that $g \circ f \simeq_{H_1} \mathrm{id}_X$ and $f \circ g \simeq_{H_2} \mathrm{id}_Y$ for suitable homotopies H_1 and H_2.

Homotopy interpretation of type theory

Paths and higher paths in a space X give rise to its so-called *fundamental ∞-groupoid*.

$$\begin{array}{cccccc} u \in X & p\colon \mathbb{I} \to X & \alpha\colon \mathbb{I} \times \mathbb{I} \to X & \theta\colon \mathbb{I} \times \mathbb{I} \times \mathbb{I} \to X & \ldots \\ & u \simeq_p v & p \simeq_\alpha q & \alpha \simeq_\theta \beta & \\ \text{points} & \text{paths} & \text{2-paths} & \text{3-paths} & \end{array}$$

- Voevodsky (2006/2009): model of type theory in simplicial sets (combinatorial representation of spaces)
- Awodey/Warren (2006): interpretation of Id-types in model structures (abstract framework for homotopy theory)

Changes our idea what kind of objects type theory is about!

Hofmann & Streicher (1998) already were wondering about this:

> *This, however, would require "2-level groupoids" in which we have morphisms between morphisms and accordingly the identity sets are not necessarily discrete. We do not know whether such structures (or even* ***infinite-level generalisations therof****) can be sensibly organised into a model of type theory.*

Spaces as types

Types	Logic	Homotopy
A	proposition	space
$a : A$	proof	point
$x : A \vdash B(x)$	predicate of sets	fibration
$x : A \vdash b(x) : B(x)$	conditional proof	section
$\mathsf{N}_0, \mathsf{N}_1$	$\bot, \top$	$\emptyset, \{*\}$
$A + B$	$A \vee B$	coproduct
$A \times B$	$A \wedge B$	product space
$A \to B$	$A \Rightarrow B$	function space
$\Sigma(x : A)\, B(x)$	$\exists(x : A)\, B(x)$	total space
$\Pi(x : A) B(x)$	$\forall(x : A)\, B(x)$	space of sections
Id_A	equality $=$	path space $A^{\mathbb{I}}$

From the HoTT Book (p.75)

An important difference between homotopy type theory and classical homotopy theory is that homotopy type theory provides a **synthetic** *description of spaces, in the following sense. Synthetic geometry is geometry in the style of Euclid: one starts from some basic notion (points and lines), constructions (a line connecting any two points), and axioms (all right angles are equal), and deduces consequences logically. This is in contrast with analytic geometry, where notions such as points and lines are represented concretely using cartesian coordinates in* $\mathbb{R}^n$*—lines are sets of points—and the basic constructions and axioms are derived from this representation. While classical homotopy theory is analytic (spaces and paths are made of points), homotopy type theory is synthetic: points, paths, and paths between paths are basic, indivisible, primitive notions.*

Propositions

Let us first look at types which are homotopically rather boring.

We call types which are "subsingletons" *(h-)propositions*:

$$\mathsf{isProp}(A) :\equiv \Pi(x\ y : A).\, x =_A y$$

Not to be confused with "propositions" from the propositions-as-types interpretation.

Example

$\mathsf{isProp}(\mathsf{N}_0)$ and $\mathsf{isProp}(\mathsf{N}_1)$,
but $\neg\, \mathsf{isProp}(\mathsf{N})$ since $0 \neq 1$ (why?)

Sets

Slightly more interesting are *sets*:

$$\mathsf{isSet}(A) :\equiv \Pi(x\ y : A)(p\ q : x =_A y).\, p = q$$

so $\mathsf{isSet}(A)$ is $\Pi(x\ y : A).\, \mathsf{isProp}(x =_A y)$.

Uniqueness of Identity Proofs simply states that any type is a set!

How is a type a set?

1. Given $x : A$ and $f : \Pi(y : A).x = y \to x = y$, then for $y : A$ and $p : x = y$
$$p = (f\,x\,1)^{-1} \cdot (f\,y\,p)$$
by (based) identity induction and $1 = (f\,x\,1)^{-1} \cdot (f\,x\,1)$
2. For $g \colon A \to B$ define: $\mathsf{const}(g) :\equiv \Pi(x\ y : A).\, g\,x =_B g\,y$
3. If we additionally know $\Pi(y : A).\, \mathsf{const}(f\,y)$ in (1), then A is a set: for $p, q : x =_A y$:
$$p = (f\,x\,1)^{-1} \cdot (f\,y\,p) = (f\,x\,1)^{-1} \cdot (f\,y\,q) = q$$

Theorem
Any proposition is a set.

Proof.
Given $\alpha : \mathsf{isProp}(A)$ we can set $f\,y\,p :\equiv \alpha\,x\,y$ in the above. □

Hedberg's theorem

A type A is *decidable* if:

$$\mathsf{dec}(A) :\equiv A + \neg A$$

Theorem (Hedberg 1995)

If A has decidable equality, i.e., we have $\Pi(x\ y : A).\, \mathsf{dec}(x = y)$, then A is a set.

Proof.

To built $f\,y : x = y \to x = y$ by distinguishing cases $x = y + \neg(x = y)$. In case $x = y$ use this element for $f\,y\,p$. Otherwise, if $\neg(x = y)$, we get a contradiction from assuming $x = y$. In both cases we have $\mathsf{const}(f\,y)$. □

Corollary

$\mathsf{isSet}(\mathsf{N}_2)$, $\mathsf{isSet}(\mathsf{N})$.

Assuming *function extensionality* we can prove that being a property or set is itself a property:

1. isProp(isProp(A))
2. isProp(isSet(A))

We can fix our type of types with a commutative operation to:

$$\Sigma(A : \mathsf{U})(\alpha : \mathsf{isSet}(A))(m : A \times A \to A). \\ \Pi(x\ y : A).\, m(x, y) = m(y, x)$$

Any two proofs of $\Pi(x\ y : A).\, m(x, y) = m(y, x)$ are then equal!

Contractibility

A special role is played by *contractible* types:

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

Example

For $a : A$ define $\mathsf{singl}(a) \equiv \Sigma(x : A).a = x$. Then:

$$\mathsf{isContr}(\mathsf{singl}(a))$$

Prove for all $x : A$ and $p : a = x$ that $(a, 1_a) = (x, p)$ by induction on p.

From an interview with Jean-Pierre Serre, 2003:

> *[T]o apply Leray's theory I needed to construct fibre spaces which did not exist if one used the standard definition. Namely, for every space X, I needed a fibre space E with base X and with trivial homotopy (for instance, contractible). But how to get such a space?*
> *One night in 1950, on the train bringing me back from our summer vacation, I saw it in a flash: just take for E the space of paths on X (with fixed origin a), the projection $E \to X$ being the evaluation map: path $\mapsto$ extremity of the path. The fibre is then the loop space of (X, a). I had no doubt: this was it! So much so that I even woke up my wife to tell her. . . [..] It is strange that such a simple construction had so many consequences.*

Loop space $\Omega(X, a)$ in type theory: $a =_X a$

Homotopy levels

One of Voevodsky's main contributions to type theory!

h−level $n\ A$ expresses that *homotopy-level* of a type A is n:

$$\begin{array}{lcl} \text{h−level}\ 0\ A & \equiv & \text{isContr}(A) \\ \text{h−level}\ (n+1)\ A & \equiv & \Pi(x\ y : A).\ \text{h−level}\ n\ (x =_A y) \end{array}$$

h-level		
0	contractible	N_1
1	proposition	N_0, N_1
2	set	N, N_2
3	groupoids	
4	2-groupoids	
...		

Warning: HoTT book uses n-types, $n \geq -2$:

$$\text{is-}n\text{-type}\ A \equiv \text{h−level}\ (n+2)\ A$$

Homotopy levels

One of Voevodsky's main contributions to type theory!

h−level $n\ A$ expresses that *homotopy-level* of a type A is n:

$$\begin{aligned} \text{h−level}\,0\,A &\quad\equiv\quad \text{isContr}(A) \\ \text{h−level}\,(n+1)\,A &\quad\equiv\quad \Pi(x\ y : A).\,\text{h−level}\,n\,(x =_A y) \end{aligned}$$

h-level		
0	contractible	N_1
1	proposition	N_0, N_1
2	set	N, N_2
3	groupoids	
4	2-groupoids	
...		

Warning: HoTT book uses n-types, $n \geq -2$:

$$\text{is-}n\text{-type}\ A \equiv \text{h−level}\,(n+2)\,A$$

Homotopy equivalences (Voevodsky)

1. Let $f\colon A \to B$.
2. For $y : B$ define $\mathsf{fib}_f(y) :\equiv \Sigma(x : A). f\,x = y$
3. f is an *equivalence* if it has contractible fibers:

$$\mathsf{isEquiv}(f) :\equiv \Pi(y : B).\, \mathsf{isContr}(\mathsf{fib}_f(y))$$

4. $A \simeq B$ iff $\Sigma(f\colon A \to B).\, \mathsf{isEquiv}(f)$
5. $\mathsf{isEquiv}(\mathrm{id}_A)$ like $\mathsf{isContr}(\mathsf{singl}(a))$
6. Type of *quasi-inverses* of f denoted $\mathsf{qinv}(f)$:

$$\Sigma(g\colon B \to A).(\Pi(x : A).\, g(f\,x) = x) \times (\Pi(y : B).\, f(g\,y) = y)$$

7. $\mathsf{qinv}(f) \leftrightarrow \mathsf{isEquiv}(f)$
8. Assuming function extensionality: $\mathsf{isProp}(\mathsf{isEquiv}(f))$

The univalence axiom

The univalence axiom specifies what the equality for universes should be.

Define

$$\mathsf{idToEquiv}_{\mathsf{U}} : \Pi(A\ B : \mathsf{U}).A =_{\mathsf{U}} B \to A \simeq B$$

by path induction, mapping $\mathsf{refl}\, A$ to id_A proving $A \simeq A$.

Univalence axiom (Voevodsky)

$$\Pi(A\ B : \mathsf{U}).\, \mathsf{isEquiv}(\mathsf{idToEquiv}_{\mathsf{U}}\, A\, B)$$

1. The univalence axiom is a statement about a universe U

$$\mathsf{UA}_{\mathsf{U}} \qquad \Pi(A\ B : \mathsf{U}).\, \mathsf{isEquiv}(\mathsf{idToEquiv}_{\mathsf{U}}\, A\, B)$$

2. $(A =_{\mathsf{U}} B) \simeq (A \simeq B)$
3. UA implies function extensionality! (Voevodsky)
4. UA implies

$$\begin{array}{lll} \mathsf{ua} & : & \Pi(A\,B : \mathsf{U}).\, A \simeq B \to A =_{\mathsf{U}} B \qquad \text{“naive univalence”} \\ \mathsf{ua}_\beta & : & \Pi(A\,B : \mathsf{U})(f : A \simeq B)(x : A). \qquad \text{“computation” rule} \\ & & \quad \mathsf{transport}^{\lambda(X:\mathsf{U}).X}(\mathsf{ua}\, f)\, x = f\, x \end{array}$$

5. ua and ua_β also imply UA (Licata)
6. Open problem: does “naive univalence” already imply UA?

UA and UIP?

Univalence is incompatible with uniqueness of identity proofs.

1. Define $\mathsf{swap} \colon \mathsf{N}_2 \to \mathsf{N}_2$ by:

$$\mathsf{swap}(\mathsf{true}) = \mathsf{false} \qquad\qquad \mathsf{swap}(\mathsf{false}) = \mathsf{true}$$

2. swap is its own quasi-inverse, thus an equivalence;
3. by UA we get: $\mathsf{ua}\ \mathsf{swap} : \mathsf{N}_2 =_{\mathsf{U}} \mathsf{N}_2$;
4. we know: $\mathsf{transport}\,(\mathsf{ua}\ \mathsf{swap})\,\mathsf{true} = \mathsf{swap}\ \mathsf{true} \equiv \mathsf{false}$,
5. but: $\mathsf{transport}\,1_{\mathsf{N}_2}\,\mathsf{true} \equiv \mathsf{true}$,
6. so: $\mathsf{ua}\ \mathsf{swap} \neq_{\mathsf{N}_2 =_{\mathsf{U}} \mathsf{N}_2} 1_{\mathsf{N}_2}$ and hence $\neg\,\mathsf{isSet}(\mathsf{U})$.

Sharpening of $\neg\,\mathsf{isSet}(\mathsf{U})$

Theorem (Kraus/Sattler)

Given a hierarchy of univalent universes $\mathsf{U}_0, \mathsf{U}_1, \mathsf{U}_2, \ldots$

$$\mathsf{U}_0 : \mathsf{U}_1 \qquad \mathsf{U}_1 : \mathsf{U}_2 \qquad \mathsf{U}_2 : \mathsf{U}_3 \qquad \ldots$$

we have

$$\neg(\mathsf{h\text{-}level}\,(n+2)\,\mathsf{U}_n).$$

Special cases of univalence

1. If $A, B : \mathsf{U}$ are propositions (so have $\mathsf{isProp}(A)$ and $\mathsf{isProp}(B)$), then:

   $$(A \leftrightarrow B) \to A \simeq B$$

   So UA implies propositional extensionality:

   $$(A \leftrightarrow B) \to A = B$$

2. If A and B are sets, equivalence specializes to bijection/isomorphism between sets.
3. If A and B are groupoids, equivalence specializes to categorical equivalence.

Equality of structures

Recall example of type CBin of types with a commutative operation:

$$\Sigma(A : \mathsf{U})(\alpha : \mathsf{isSet}(A))(m : A \times A \to A)\Pi(x\, y : A).\, m(x, y) = m(y, x)$$

Given $A' = (A, \alpha_A, m_A, p_A)$ and $B' = (B, \alpha_B, m_B, p_B)$ there is an obvious candidate of homomorphism:

$$f\colon A \to B \text{ such that } f(m_A(x, y)) = m_B(f\, x, f\, y)$$

Univalence lifts to isomorphisms of this algebraic structure:

$$(A' \cong B') \simeq (A' = B')$$

This works for many other algebraic structures (Aczel, Coquand/Danielsson)

Independence of UA

Voevodsky gave a model of UA using Kan simplicial sets formulated in a *classical* meta-theory (ZFC plus two inaccessible cardinals).

(Various *constructive* models based on cubical sets; more later…)

(Since the set-theoretic model is not a model of UA we know that UA is independent of intensional MLTT.)

So MLTT can't distinguish equivalent types:
Given $P \colon \mathsf{U} \to \mathsf{U}$ and $A, B : \mathsf{U}$ with $A \simeq B$, we can't have $P\,A$ but $\neg(P\,B)$.

In contrast to set theory: $\{0\} \cong \{1\}$ and $0 \in \{0\}$, but $0 \notin \{1\}$.

Summary

- Intensional MLTT lacks extensionality principles
- Iterating Martin-Löf's identity type gives rise to interesting structure
- Types as spaces, spaces as types. Changes our idea what this theory is a theory of! (Not necessarily sets!)
- Types can be stratified by their h-level
- The univalence axiom is a strong extensionality principle. Type theory invariant under equivalence.

Propositional truncation

- When is $f\colon A \to B$ surjective?

$$\Pi(y : B)\Sigma(x : A).\, f\, x = y$$

- Given a type A its propositional truncation is a proposition $\|A\|$ with $\mathsf{inc}\colon A \to \|A\|$, such that for any other type B with $\mathsf{isProp}(B)$ there is a map

$$\mathsf{rec}\colon (A \to B) \to \|A\| \to B$$

  with: $\mathsf{rec}\, f\, (\mathsf{inc}\, a) \equiv f\, a : B$
- If A is a proposition, then $A \leftrightarrow \|A\|$, so $A \simeq \|A\|$, so $A = \|A\|$.
- $\|A\|$ expresses that A is inhabited, but we can't extract its witness in general

Propositional truncation

- When is $f\colon A \to B$ surjective?

$$\Pi(y : B)\Sigma(x : A).\, f\,x = y$$

  This is the space of sections!?

- Given a type A its propositional truncation is a proposition $\|A\|$ with $\mathsf{inc}\colon A \to \|A\|$, such that for any other type B with $\mathsf{isProp}(B)$ there is a map

$$\mathsf{rec}\colon (A \to B) \to \|A\| \to B$$

  with: $\mathsf{rec}\, f\, (\mathsf{inc}\, a) \equiv f\, a : B$

- If A is a proposition, then $A \leftrightarrow \|A\|$, so $A \simeq \|A\|$, so $A = \|A\|$.
- $\|A\|$ expresses that A is inhabited, but we can't extract its witness in general

Propositional truncation

- When is $f\colon A \to B$ surjective?

$$\Pi(y : B)\Sigma(x : A).\, f\,x = y$$

  This is the space of sections!?

- Given a type A its propositional truncation is a proposition $\|A\|$ with $\mathsf{inc}\colon A \to \|A\|$, such that for any other type B with $\mathsf{isProp}(B)$ there is a map

$$\mathsf{rec}\colon (A \to B) \to \|A\| \to B$$

  with: $\mathsf{rec}\, f\, (\mathsf{inc}\, a) \equiv f\, a : B$

- If A is a proposition, then $A \leftrightarrow \|A\|$, so $A \simeq \|A\|$, so $A = \|A\|$.
- $\|A\|$ expresses that A is inhabited, but we can't extract its witness in general

Propositional truncation

- When is $f\colon A \to B$ surjective?

  $$\Pi(y : B)\Sigma(x : A).\, f\, x = y$$

  This is the space of sections!?

- Given a type A its propositional truncation is a proposition $\|A\|$ with $\mathsf{inc}\colon A \to \|A\|$, such that for any other type B with $\mathsf{isProp}(B)$ there is a map

  $$\mathsf{rec}\colon (A \to B) \to \|A\| \to B$$

  with: $\mathsf{rec}\, f\, (\mathsf{inc}\, a) \equiv f\, a : B$

- If A is a proposition, then $A \leftrightarrow \|A\|$, so $A \simeq \|A\|$, so $A = \|A\|$.
- $\|A\|$ expresses that A is inhabited, but we can't extract its witness in general

The logic of h-propositions

We can define surjective as:

$$\Pi(y : B)\, \|\Sigma(x : A).\, f\, x = y\|$$

This suggest a interpretation of the logical connectives as (h-)propositions

$$\begin{array}{rcl} \top & \equiv & \mathsf{N}_1 \\ \bot & \equiv & \mathsf{N}_0 \\ A \Rightarrow B & \equiv & A \to B \\ A \wedge B & \equiv & A \times B \\ A \vee B & \equiv & \|A + B\| \\ \forall(x : A)\, B & \equiv & \Pi(x : A)\, B \\ \exists(x : A)\, B & \equiv & \|\Sigma(x : A)\, B\| \end{array}$$

This interpretation satisfies all the expected properties from logic.

Propositional truncation

What kind of type former is $\|-\|$?

$$\frac{a : A}{\mathsf{inc}\, a : \|A\|} \qquad \frac{u : \|A\| \qquad v : \|A\|}{\mathsf{squash}\, u\, v : \mathsf{Id}_{\|A\|}(u, v)}$$

Has constructors for points and paths! (From the recursor one can derive a suitable induction principle.)

Compare: inductive types specified by point constructors

$$\frac{}{0 : \mathsf{N}} \qquad \frac{n : \mathsf{N}}{\mathsf{S}\, n : \mathsf{N}}$$

Propositional truncation is a **higher inductive type** (HIT).