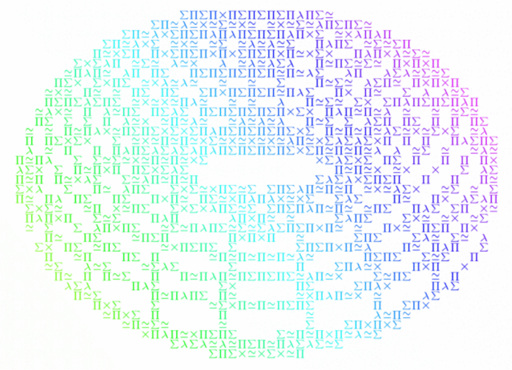
# Homotopy type theory



Ingo Blechschmidt

November 25th, 2014

# Outline

# What are foundations?

- Foundations set the logical context for doing maths.
- Their details don't matter in everyday work (mostly).
- But their main concepts do.

http://collabcubed.com/2012/10/24/high-trestle-trail-bridge-rdg/

# What are foundations?

- Foundations set the logical context for doing maths.
- Their details don't matter in everyday work (mostly).
- But their main concepts do.

- Classical foundations are *set-based* (ZF, ZFC, ...):
  **Everything is a set.**
- $0 := \emptyset, \quad 1 := \{0\}, \quad 2 := \{0,1\}, \quad \ldots$
- $(x,y) := \{\{x\},\{x,y\}\}$ (Kuratowski pairing)
- $(x,y,z) := (x,(y,z))$
- maps: $(X,Y,R)$ with $R \subseteq X \times Y$ such that ...

# What’s wrong with set-based foundations?

Set-based foundations …

- do not reflect typed mathematical practice,
- do not respect equivalence of structures,
- require complex encoding of “higher-level” subjects, complicating interactive proof environments.

# What is homotopy type theory?

- Homotopy type theory is a new foundational theory.
- Basic notions have a homotopy-theoretic flavour.
- One can start doing "real mathematics" right away, without complex encodings.
- Initiated by Voevodsky in 2005.

Some participants of the IAS special year

# What is homotopy type theory?

Homotopy type theory …

- is elegant,
- reflects mathematical practice,
- contains wondrous new concepts,
- ensures that everything respects equivalences,
- simplifies the plumbing of homotopy theory,
- allows for accessible computer formalization.

# What are values and types?

- In type theory, there are **values** and **types**.
- Every value is of exactly one type.
- Types may depend on values.


$$
\begin{aligned}
7 &: \mathbb{N} \\
(3,5) &: \mathbb{N} \times \mathbb{N} \\
\mathsf{succ} &: \mathbb{N} \to \mathbb{N} \\
\text{zero vector} &: \mathbb{R}^n \quad (n : \mathbb{N})
\end{aligned}
$$

# What are values and types?

- In type theory, there are **values** and **types**.
- Every value is of exactly one type.
- Types may depend on values.


$$
\begin{aligned}
7 &: \mathbb{N} \\
(3,5) &: \mathbb{N} \times \mathbb{N} \\
\mathsf{succ} &: \mathbb{N} \to \mathbb{N} \\
\text{zero vector} &: \mathbb{R}^n \quad (n : \mathbb{N})
\end{aligned}
$$

Let $B(x)$ be a type family depending on $x : A$.

- $\sum_{x:A} B(x) = \text{“}\{(a,b) \mid a : A, b : B(a)\}\text{”}$
- $\prod_{x:A} B(x) = \text{“}\{f : A \to \,?? \mid f(a) : B(a) \text{ for all } a : A\}\text{”}$

# What is the dependent equality type?

In set theory, for a set $X$ and elements $x, y \in X$:

- "$x = y$" is a **proposition**.
- Set theory is **layered above** predicate logic.

In type theory, for a type $X$ and values $x, y : X$:

- There is the **equality type** $\mathrm{Id}_X(x, y)$ or $(x =_X y)$.
- To verify that "$x = y$", exhibit a value of $(x = y)$.
- Have $\mathsf{refl}_x : (x = x)$.
- Identity types may contain zero or **many** values!

Intuition: $(x = y)$ is the type of **proofs** that "$x = y$".

# What is the dependent equality type?

In set theory, for a set $X$ and elements $x, y \in X$:

- "$x = y$" is a **proposition**.
- Set theory is **layered above** predicate logic.

In type theory, for a type $X$ and values $x, y : X$:

- There is the **equality type** $\mathrm{Id}_X(x, y)$ or $(x =_X y)$.
- To verify that "$x = y$", exhibit a value of $(x = y)$.
- Have $\mathsf{refl}_x : (x = x)$.
- Identity types may contain zero or **many** values!

Intuition: $(x = y)$ is the type of **proofs** that "$x = y$".

Intuition: $(x = y)$ is the type of **paths** $x \rightsquigarrow y$.

# How are types like spaces?

| homotopy theory | type theory |
|---|---|
| **space** $X$ | type $X$ |
| **point** $x \in X$ | value $x : X$ |
| **path** $x \rightsquigarrow y$ | value of $(x = y)$ |
| **(continuous) map** | value of $X \to Y$ |

- A **homotopy** between maps $f, g : X \to Y$ is a value of

$$(f \simeq g) :\equiv \prod_{x:X} (f(x) = g(x)).$$

- A space $X$ is **contractible** iff

$$\mathsf{IsContr}(X) :\equiv \sum_{x:X} \prod_{y:X} (x = y).$$

# How are types like spaces?

- "The type $X$ is **contractible**":

$$\mathsf{IsContr}(X) :\equiv \sum_{x:X} \prod_{y:X} (x = y).$$

- "The type $X$ is a **mere proposition**":

$$\mathsf{IsMereProp}(X) :\equiv \prod_{x,y:X} (x = y)$$

- "The type $X$ is a **set** or **discrete space**":

$$\mathsf{IsSet}(X) :\equiv \prod_{x,y:X} \mathsf{IsMereProp}(x = y)$$

- For instance, $\mathbb{N}$ is a set.

# How are types like spaces?

- Functions are automatically **continuous/functorial**:

$$(x = y) \longrightarrow (f(x) = f(y)).$$

- Type families $P : X \to \mathcal{U}$ automatically behave like **fibrations**, in that fibers over connected points are equivalent:

$$(x = y) \longrightarrow (P(x) \simeq P(y)).$$

# How are constructions encoded?

- The **fiber** of a map $f : X \to Y$ over a point $y : Y$ is

$$\mathsf{fib}_f(y) :\equiv \sum_{x:X} (f(x) = y).$$

- The **path space** of $X$ is

$$X^I :\equiv \sum_{x,y:X} (x = y).$$

- The **based loop space** of $X$ at $x$ is

$$\Omega^1(X, x) :\equiv (x = x).$$

- The **path fibration** of $(X, x)$ is the map

$$\mathsf{fst} : \sum_{y:X} (x = y) \to X.$$

# How are constructions encoded?

- The **fiber** of a map $f : X \to Y$ over a point $y : Y$ is

$$\mathsf{fib}_f(y) :\equiv \sum_{x:X} (f(x) = y).$$

- The **path space** of $X$ is

$$X^I :\equiv \sum_{x,y:X} (x = y).$$

- The **based loop space** of $X$ at $x$ is

$$\Omega^1(X, x) :\equiv (x = x).$$

- The **path fibration** of $(X, x)$ is the map

$$\mathsf{fst} : \sum_{y:X} (x = y) \to X.$$

# What are higher inductive definitions?

The type $\mathbb{N}$ of natural numbers is **freely generated** by

- a point $0 : \mathbb{N}$ and
- a function $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$.

This definition gives rise to an **induction principle**

$$\prod_{A:\mathbb{N}\to\mathcal{U}} \Big( A(0) \times \Big( \prod_{n:\mathbb{N}} A(n) \to A\big(\mathsf{succ}(n)\big) \Big) \longrightarrow \prod_{n:\mathbb{N}} A(n) \Big),$$

and a **recursion principle**

$$\prod_{X:\mathcal{U}} \Big( X \times \big( \mathbb{N} \to (X \to X) \big) \longrightarrow (\mathbb{N} \to X) \Big).$$

# How to present famous spaces?

The **circle** $S^1$ is generated by

- a point $\mathsf{base} : S^1$ and
- a path $\mathsf{loop} : (\mathsf{base} = \mathsf{base})$.

The **sphere** $S^2$ is generated by

- a point $\mathsf{base} : S^2$ and
- a path $\mathsf{surf} : (\mathsf{refl}_{\mathsf{base}} = \mathsf{refl}_{\mathsf{base}})$.

The **torus** $T^2$ is generated by

- a point $b : T^2$,
- a path $p : (b = b)$,
- a path $q : (b = b)$, and
- a 2-path $t : (p \cdot q = q \cdot p)$.

# How to present famous spaces?

The **suspension** $\Sigma X$ of $X$ is generated by

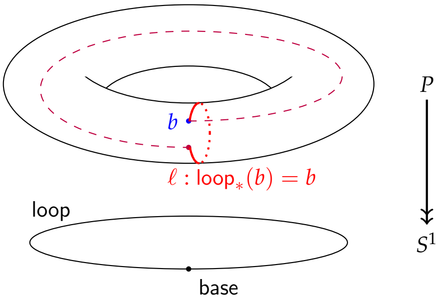
- a point $\mathsf{N} : \Sigma X$ and
- a point $\mathsf{S} : \Sigma X$ and
- a function $\mathsf{merid} : X \to (N = S)$.

The **cylinder** $\mathrm{Cyl}(X)$ of $X$ is generated by

- a function $\mathsf{bot} : X \to \mathrm{Cyl}(X)$ and
- a function $\mathsf{top} : X \to \mathrm{Cyl}(X)$ and
- a function $\mathsf{seg} : \prod_{x:X}(\mathsf{bot}(x) = \mathsf{top}(x))$.

Of course, we can show $\mathrm{Cyl}(X) \simeq X \times I \simeq X$.

# What is circle induction?




The **induction principle** of $S^1$ states: Given $P : S^1 \to \mathcal{U}$,

- a point $b : P(\mathsf{base})$, and
- a path $\ell : \mathsf{loop}_*(b) = b$

there is a function $f : \prod_{x:S^1} P(x)$ such that

- $f(\mathsf{base}) \equiv b$ and
- $f(\mathsf{loop}) = \ell$.

# What is type truncation?

Let $X$ be a type.

The **propositional truncation** $\|X\|_{-1}$ is generated by

- a function $X \to \|X\|_{-1}$ and
- for any $x, y : \|X\|_{-1}$, a path $x = y$.

The **0-truncation** $\|X\|_0$ is generated by

- a function $X \to \|X\|_0$ and
- for any $x, y : \|X\|_0$, $p, q : (x = y)$, a path $p = q$.

The **fundamental group** of $(X, x_0)$ is

$$\pi_1(X, x_0) :\equiv \|\Omega^1(X, x_0)\|_0 :\equiv \|(x_0 = x_0)\|_0.$$

# What is the univalence axiom?

An **equivalence** is a function $f : X \to Y$ such that

$$\mathsf{IsEquiv}(f) :\equiv \prod_{y:Y} \mathsf{IsContr}(\mathsf{fib}_f(y)).$$

Types $X$ and $Y$ are **equivalent** iff

$$(X \simeq Y) :\equiv \sum_{f:X\to Y} \mathsf{IsEquiv}(f).$$

The **univalence axiom** states: The canonical function

$$(X = Y) \longrightarrow (X \simeq Y)$$

is an equivalence, for all types $X$ and $Y$.

# What's the status of the axiom of choice?

- The following proposition is **just true**, but is not a faithful rendition of the axiom of choice:

$$\Big(\prod_{x:A}\sum_{y:B} R(x,y)\Big) \longrightarrow \sum_{f:A\to B}\prod_{x:A} R(x,f(x)).$$

- The real axiom of choice,

$$\Big(\prod_{x:A}\Big\|\sum_{y:B} R(x,y)\Big\|_{-1}\Big) \longrightarrow \Big\|\sum_{f:A\to B}\prod_{x:A} R(x,f(x))\Big\|_{-1},$$

  can be added as an axiom, but is rarely needed.

# What's the status of the axiom of choice?

- The following proposition is **just true**, but is not a faithful rendition of the axiom of choice:

$$\Big(\prod_{x:A}\sum_{y:B} R(x,y)\Big) \longrightarrow \sum_{f:A\to B}\prod_{x:A} R(x,f(x)).$$

- The real axiom of choice,

$$\Big(\prod_{x:A}\Big\|\sum_{y:B} R(x,y)\Big\|_{-1}\Big) \longrightarrow \Big\|\sum_{f:A\to B}\prod_{x:A} R(x,f(x))\Big\|_{-1},$$

  can be added as an axiom, but is rarely needed.
- The law of excluded middle is too rarely needed.

$$\mathsf{LEM} :\equiv \prod_{A:\mathcal{U}}\Big(\mathsf{IsMereProp}(A) \to A + \neg A\Big).$$

# What are models of HoTT?

Conjecturally, HoTT can be interpreted in any **$(\infty, 1)$-topos**. Verified models include

- $\infty$Grpd, i. e. a model in simplicial sets, and
- $(\infty, 1)$-presheaf toposes over elegant Reedy categories.

Thus, any theorem proven in HoTT holds in the context of classical homotopy theory and in more general contexts.

# References

- The textbook
  http://homotopytypetheory.org/book/
- Voevodsky on his motivations
  http://www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/2014_IAS.pdf
- Seminar slides
  http://www.math.ias.edu/~mshulman/hottseminar2012/01intro.pdf
  http://www.math.ias.edu/~mshulman/hottminicourse2012/04induction.pdf
  https://coq.inria.fr/files/coq5-slides-spitters.pdf
  https://www.andrew.cmu.edu/user/awodey/hott/CMUslides.pdf
- An application unrelated to homotopy theory
  http://www.cs.nott.ac.uk/~txa/talks/lyon14.pdf
- hott-amateurs mailing list