Homotopy Type Theory

Jeremy Avigad

Department of Philosophy and
Department of Mathematical Sciences
Carnegie Mellon University

February 2016

Homotopy Type Theory

HoTT relies on a novel homotopy-theoretic interpretation of dependent type theory.

Disclaimer #1: I am a poor substitute for Steve Awodey.

- I have borrowed slides from him and Floris van Doorn.
- Any inaccuracies are mine.

Disclaimer #2: Homotopy type theory means different things to different people.

- I'll try to survey different perspectives.
- Ultimately, I can only represent my own.

Homotopy Type Theory

HoTT as a framework for synthetic algebraic topology

- Spaces, continuous dependences, and paths are primitives.
- Provides a language for constructing topological objects.

HoTT as a novel foundation for mathematics

- With topological notions as primitive, we can view sets as particular (unstructured) spaces.
- The logical system respects isomorphism of structures.

HoTT as a framework for generic programming

- Dependent type theory has a computational interpretation.
- Paths between types are equivalences of representations.
- Dependent type theory respects these equivalences.

Homotopy Type Theory

As in any branch of logic:

- syntax: variations on / extensions of dependent type theory
- semantics: algebraic and topological interpretations

Strands of research:

- semantics of HoTT and higher inductive types
- extensions and variants of the deductive systems
- computational interpretations of HoTT
- formalization of algebraic topology
- formalization of mathematics

Homotopy Type Theory

The main ideas:

1. types as spaces, terms as continuous dependences
2. identities as paths, elimination as transport
3. proof relevance: proofs as constructions
4. sets as 0-truncated types
5. higher inductive types
6. univalence and invariance

Idea #1: Types as spaces

Dependent type theory has a computational interpretation.

- Types are datatypes.
- Terms are computable functions.

Computation has often been understood in topological terms:

- Computable functions $\mathbb{R} \to \mathbb{R}$ are continuous
- Topological models of intuitionistic logic (Rasiowa and Sikorski)
- Topological models of the lambda calculus (Scott)
- Toposes as models of intuitionistic higher-order logic

Idea #1: Types as spaces

Dependent type theory has such an interpretation.

Compare:

- Set-theoretic interpretation: types are sets, terms are functions
- Computational interpretation: types are datatypes, terms are computable functions
- Topological interpretation: types are spaces, and terms are continuous functions

Better: types are abstract spaces up to homotopy (homotopy types).

Idea #1: Types as spaces

Start with Martin-Löf type theory:

- Basic constructions:
  - $\Pi x : A,\ B$
  - $\Sigma x : A,\ B$
  - $Id_A(x, y)$
- Can add universes of types $Type_0$, $Type_1$, $Type_2$, ...
- There is no *Prop*: treat data and propositions the same
- Can add inductive types.

Idea #1: Types as spaces

Theorem (Awodey and Warren). The identity types in Martin-Löf type theory can be interpreted in any model category.

Notes:

- A Quillen model category is a category with extra structure (weak equivalences, fibrations and cofibrations).
- It provides an abstract setting for homotopy theory.
- $\Pi x \in A,\ B$ and $\Sigma x \in A,\ B$, etc. can be interpreted with additional structure.

Idea #1: Types as spaces

Theorem (Voevodsky).

- Martin-Löf type theory can be interpreted in the category of Kan simplicial sets.
- We can interpret universes of types so as to satisfy the *univalence* axiom. (More below.)

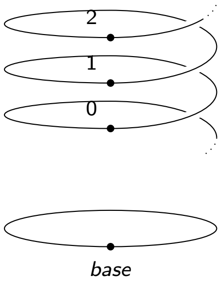
Idea #1: Types as spaces

	Set Theory	**Logic**	**Topology**
A	Set	Proposition	Space
$a : A$	Element	Proof	Point
$A \times B$	Product	Conjunction	Product
$A + B$	Disjoint Sum	Disjunction	Coproduct
$A \to B$	Functions	Implication	Mapping space
$P : A \to Type$	Family of sets	Predicate	Fibration
$\Pi x : A,\ P(x)$	Dep. Functions	U. Quantifier	Dep. product

Idea #1: Types as spaces

If $B(x) : \mathit{Type}$ depends on $x : A$, think of B as a *fibration* over A.

That is, B is a family of spaces parameterized continuously over A.



$$\begin{tikzcd}
\Sigma x : A,\ B(x) \arrow[d] \\
A
\end{tikzcd}$$

Idea #2: Identities as paths

$Id_A(x, y)$ can also be written $x =_A y$.

In logic, it says that $x : A$ and $y : A$ are equal.

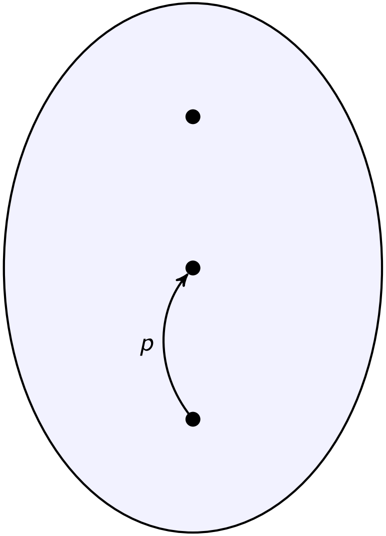
In the topological interpretation, it says there is a *path* from x to y.

Moreover, suppose $s : B$ and $t : B$ depend on x of type A.

As a function of x, $p : s =_B t$ depends continuously on x.

In other words, it is a *homotopy* between the functions.

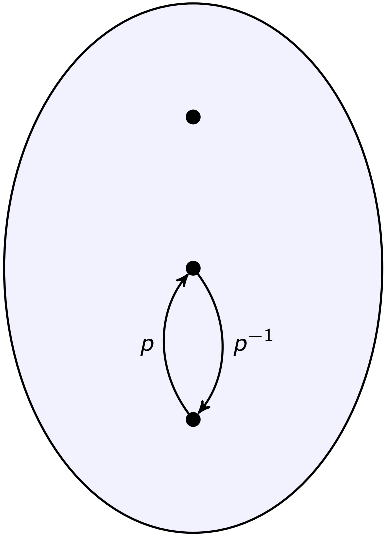
Idea #2: Identities as paths



Idea #2: Identities as paths

We can

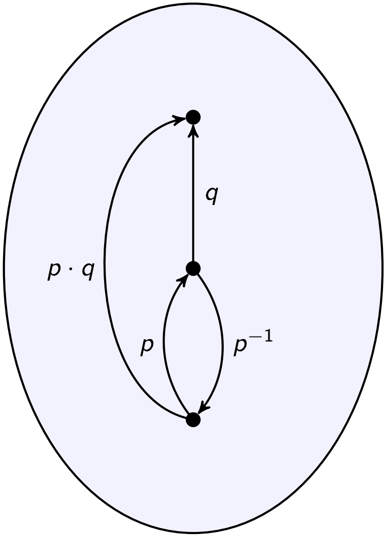
- invert paths (symmetry);



Idea #2: Identities as paths

We can

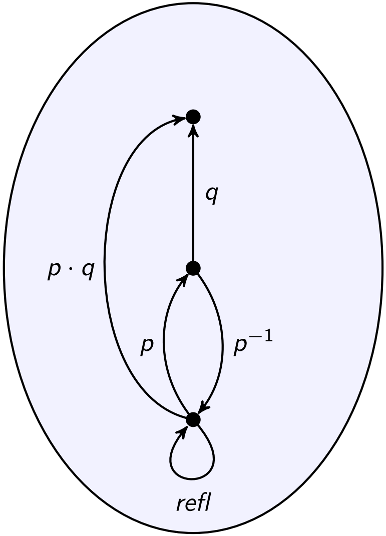
- invert paths (symmetry);
- concatenate paths (transitivity);



Idea #2: Identities as paths

We can

- invert paths (symmetry);
- concatenate paths (transitivity);
- form identity paths (reflexivity).
- and more . . .



Idea #2: Identities as paths

The familiar rule for equality:

$$\frac{p : x =_A y \qquad b : B\ x}{\mathit{subst}\ p\ b : B\ y}$$

This says that the path p in A can be *lifted* to a path in B.

Alternatively, it says that $b : B(x)$ can be *transported* along p.

The rule for equality in dependent type theory is slightly more general.

It corresponds to contracting p to the identity path.

Idea #3: Proof relevance

For any type A, we can form a new type, Id_A.

So for any x and y, we can form $Id_{Id_A(x,y)}$.

Given $x, y : A$ and $p, q : x =_A y$ we can ask whether $p =_{Id_A(x,y)} q$.

In ordinary mathematics, this is strange!

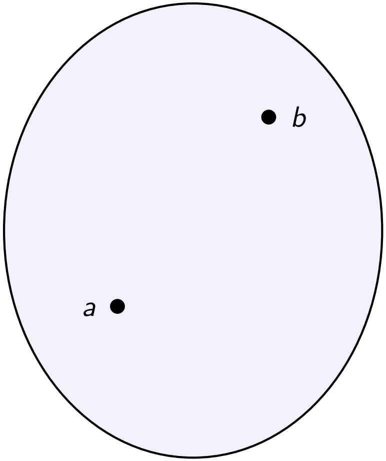
In Lean's standard library, p and q are considered the same: *proof irrelevance*.

In HoTT, these types encode important information.

Idea #3: Proof relevance

A type A can have

- points $a, b : A$

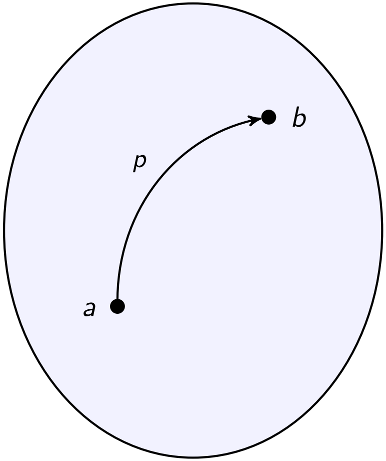


Idea #3: Proof relevance

A type A can have

- paths

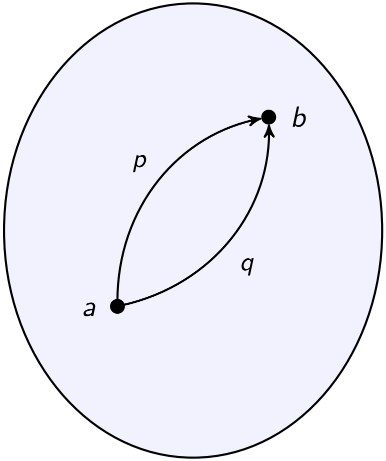
  $p : a =_A b$
- points $a, b : A$



Idea #3: Proof relevance

A type A can have

- paths
  $q, p : a =_A b$
- points $a, b : A$

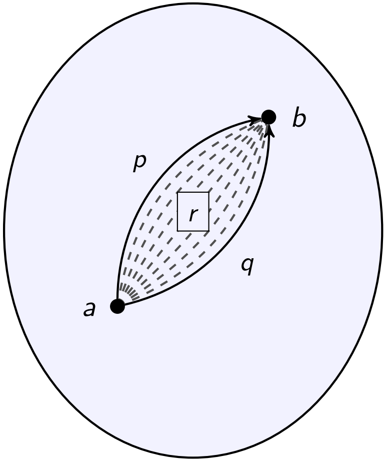


Idea #3: Proof relevance

A type A can have

$$\vdots$$

- paths between paths $r : p = q$
- paths $q, p : a =_A b$
- points $a, b : A$

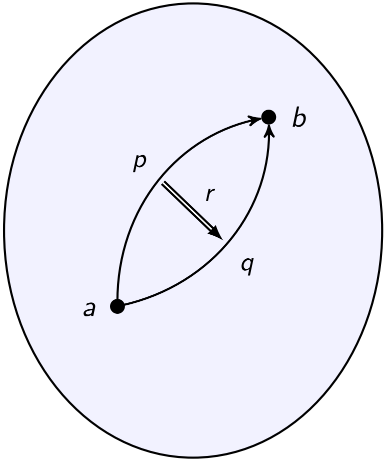


Idea #3: Proof relevance

A type A can have

$$\vdots$$

- paths between paths $r : p = q$
- paths $q, p : a =_A b$
- points $a, b : A$



Idea #3: Proof relevance

In HoTT, we don't distinguish between datatypes and propositions, or between data and proofs.

Every proof is a construction.

We often want to reason about the constructions.

Idea #4: Sets as 0-truncated types

Voevodsky observed that the universe of all types is naturally stratified by *homotopical dimension*.

A type A is called:

contractible	iff	$\sum_{x:A} \prod_{y:A} \mathtt{Id}_A(x,y)$,

A type A is a:

proposition	iff	$\prod_{x,y:A}\ \mathit{Contr}(\mathtt{Id}_A(x,y))$,
set	iff	$\prod_{x,y:A}\ \mathit{Prop}(\mathtt{Id}_A(x,y))$,
1-type	iff	$\prod_{x,y:A}\ \mathit{Set}(\mathtt{Id}_A(x,y))$,
$(n+1)$-type	iff	$\prod_{x,y:A}\ n\mathit{Type}(\mathtt{Id}_A(x,y))$,

Idea #4: Sets as 0-truncated types

The net effect is that we can view ordinary set-theoretic reasoning as reasoning about relatively simple, unstructured types.

And now we have a much bigger universe, with interesting structure to explore.

Idea #5: Higher inductive types

The natural numbers $\mathbb{N}$ are given as an (ordinary) inductive type:

$$\mathbb{N} := \begin{cases} 0 : \mathbb{N} \\ s : \mathbb{N} \to \mathbb{N} \end{cases}$$

The *recursion property* is captured by an elimination rule:

$$\frac{a : X \qquad f : X \to X}{rec(a, f) : \mathbb{N} \to X}$$

with computation rules:

$$\begin{aligned} rec(a, f)(0) &= a \\ rec(a, f)(sn) &= f(rec(a, f)(n)) \end{aligned}$$

Idea #5: Higher inductive types

In HoTT, we want to generate types with nontrivial path structure.

Lumsdaine and Shulman observed that this can be done with *higher inductive types*.

Idea #5: Higher inductive types

For example, the homotopical circle $\mathbb{S}$ can be given as an inductive type involving a higher-dimensional generator:

$$\mathbb{S} := \begin{cases} \mathit{base} : \mathbb{S} \\ \mathit{loop} : \mathtt{Id}_{\mathbb{S}}(\mathit{base}, \mathit{base}) \end{cases}$$

Here we think of $\mathit{loop} : \mathtt{Id}_{\mathbb{S}}(\mathit{base}, \mathit{base})$ as a non-trivial path,

$$\begin{tikzcd} \mathit{base} \arrow[loop above, "\mathit{loop}"] \end{tikzcd}$$

Idea #5: Higher inductive types

$$\mathbb{S} := \begin{cases} \mathit{base} : \mathbb{S} \\ \mathit{loop} : \mathtt{Id}_{\mathbb{S}}(\mathit{base}, \mathit{base}) \end{cases}$$

The *recursion principle* of $\mathbb{S}$ is given by its elimination rule:

$$\frac{a : X \qquad p : \mathtt{Id}_{\mathbb{X}}(a, a)}{\mathit{rec}(a, p) : \mathbb{S} \to X}$$

with computation rules:

$$\mathit{rec}(a, p)(\mathit{base}) = a$$
$$\mathit{rec}(a, p)(\mathit{loop}) = p$$

Idea #5: Higher inductive types

Many basic spaces and constructions can be introduced as HITs:

- higher spheres $\mathbb{S}^n$, cylinders, tori, cell complexes, . . . ,
- suspensions ΣA, mapping cylinders, homotopy pushouts, etc.,
- truncations, such as connected components $\pi_0(A)$ and "bracket" types $[A]$,
- (higher) homotopy groups π_n, Eilenberg-MacLane spaces $K(G, n)$, Postnikov systems,
- Quillen model structure,
- quotients by equivalence relations and more general quotients,
- free algebras, algebras presented by generators and relations,
- real numbers,
- cumulative hierarchy of sets.

Idea #5: Higher inductive types

We still do not have a good general theory of HITs.

We know that certain syntactic constructions make sense in certain (classes of) models.

The full theory is still under development.

Idea #6: Univalence

Suppose we add a universe *Type* to type theory.

Then for every $A, B : \mathit{Type}$, we have the type $A =_{\mathit{Type}} B$.

Equality of types will satisfy all the usual rules and properties.

We can also define the usual notion of isomorphism of types:

$$\begin{aligned} A \cong B \;\Leftrightarrow\; & \textit{there are } f : A \to B \textit{ and } g : B \to A \\ & \quad \textit{such that } g \circ f = 1_A \textit{ and } f \circ g = 1_B. \end{aligned}$$

Idea #6: Univalence

In HoTT, $A \cong B$ is not a proposition; it is a structure.

A is *weakly equivalent* to B, written $A \simeq B$, says there exists an $f : A \to B$ which has both a left and right inverse.

The univalence axiom implies that if $A \simeq B$, then $A = B$.

This is a very strong invariance principle: if we can establish a weak equivalence between A and B, we can substitute A for B in any context.

The idea extends to isomorphisms of structures.

Idea #6: Univalence

In fact, the univalence axiom is even more dramatic:

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

More precisely: the straightforward map from right to left is a weak equivalence.

Voevodsky showed that the axiom is true in the category of Kan simplicial sets, with a suitable interpretation of *Type*.

Recap

The main ideas:

1. types as spaces, terms as continuous dependences
2. identities as paths, elimination as transport
3. proof relevance: proofs as constructions
4. sets as 0-truncated types
5. higher inductive types
6. univalence and invariance

Recap

Current research interests:

- semantics of HoTT and higher inductive types
- extensions and variants of the deductive systems
- computational interpretations
- formalization of algebraic topology
- formalization of mathematics

The field is active and vibrant.

Observations

Any “foundation” for mathematics should make sense of the full range of things that mathematicians do.

As a result, different foundations tend to be inter-interpretable.

If you are looking for philosophical reassurance that mathematics is meaningful and consistent, you should pick the one you like best.

Beyond that, let’s recognize that different descriptions of mathematics are useful for different purposes.

The real scientific question:

How can we put them to good use?