An introduction to homotopy type theory

Paige Randall North

University of Pennsylvania

16 September 2020

Outline

Type theory

Homotopy type theory

Directions

Constructivism (1930 - 1970)

Moral 1: Proofs should be programs.

The proof of a statement like "For every foo, there exists a bar," should be an algorithm that constructs a bar from a foo.

(Avoid axiom of choice and the law of excluded middle.)

Moral 2: Identify propositions with underlying math objects.

For example, a theorem like "A space X is contractible," is identified with the space of pairs (x, s) where $x \in X$ and s is a section of π_x,

$$\begin{tikzcd}
\bullet \arrow[r] \arrow[d, "\pi_x"] \arrow[dr, phantom, "\lrcorner", very near start] & X^{[0,1]} \arrow[d, "ev_0 \times ev_1"] \\
* \times X \arrow[r, "x \times id_X"] \arrow[u, bend left, dashed] & X \times X
\end{tikzcd}$$

and proofs of this theorem are identified with points of the space.

Types as propositions, sets, or program specifications

Types	Propositions	Sets	Program specifications
$A \quad \text{TYPE}$	A is a proposition	A is a set	A is a program specification
$a : A$	a is a witness of A	$a \in A$	a meets A
	$A \wedge B$	$A \times B$	Do A and B
	$A \vee B$	$A + B$	Do A or B
	$A \implies B$	$A \to B$	Turn any program that does A into a program that does B
	$\top$	$*$	Do nothing
	$\bot$	$\varnothing$	Do the impossible
	$\neg A$	$A \to \varnothing$	Turn any program that does A into an impossible program

Types as propositions, sets, or program specifications

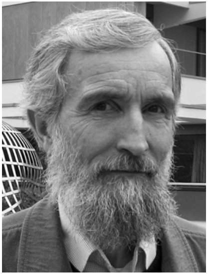
Types	Propositions	Sets	Program specifications
$A\ \ \textsc{type}$	A is a proposition	A is a set	A is a program specification
$a : A$	a is a witness of A	$a \in A$	a meets A
	$A \wedge B$	$A \times B$	Do A and B
	$A \vee B$	$A + B$	Do A or B
	$A \implies B$	$A \to B$	Turn any program that does A into a program that does B
	$\top$	$*$	Do nothing
	$\bot$	$\varnothing$	Do the impossible
	$\neg A$	$A \to \varnothing$	Turn any program that does A into an impossible program

Dependent types

- Types and terms often depend on ‘contexts’.
- $n : \mathbb{N} \vdash 2n : \mathbb{N}$
- $n : \mathbb{N} \vdash \mathrm{isEven}(2n)$
- $x : X \vdash \mathrm{Sections}(\pi_x)$

Martin-Löf Type Theory (1970's)

System of dependent type theory with type and term formers generally given *inductively* by 4 rules.



- $\mathbb{B}$-formation

$$\frac{}{\Gamma \vdash \mathbb{B}\ \text{TYPE}}$$

- $\mathbb{B}$-introduction

$$\frac{}{0 : \mathbb{B}} \quad \frac{}{1 : \mathbb{B}}$$

- $\mathbb{B}$-elimination

$$\frac{\Gamma, b : \mathbb{B} \vdash D(b) \qquad \Gamma \vdash z : D(0) \qquad \Gamma \vdash o : D(1)}{\Gamma, b : \mathbb{B} \vdash j(b) : D(b)}$$

- $\mathbb{B}$-computation

$$\frac{\Gamma, b : \mathbb{B} \vdash D(b) \qquad \Gamma \vdash z : D(0) \qquad \Gamma \vdash o : D(1)}{\Gamma \vdash j(0) \equiv z : D(0) \qquad \Gamma \vdash j(1) \equiv o : D(1)}$$

Another example: coproduct

- +-formation

$$\frac{\Gamma \vdash A \text{ TYPE} \qquad \Gamma \vdash B \text{ TYPE}}{\Gamma \vdash A + B \text{ TYPE}}$$

- +-introduction

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{inl}(a) : A + B} \quad \frac{\Gamma \vdash b : B}{\Gamma \vdash \mathsf{inr}(b) : A + B}$$

- +-elimination

$$\frac{\begin{array}{c}\Gamma, p : A + B \vdash D(p) \text{ TYPE} \\ \Gamma, a : A \vdash d(a) : D(\mathsf{inl}(a)) \qquad \Gamma, b : B \vdash e(b) : D(\mathsf{inr}(b))\end{array}}{\Gamma, p : A + B \vdash f(p) : D(p)}$$

- +-computation

$$\frac{\begin{array}{c}\Gamma, p : A + B \vdash D(p) \text{ TYPE} \\ \Gamma, a : A \vdash d(a) : D(\mathsf{inl}(a)) \qquad \Gamma, b : B \vdash e(b) : D(\mathsf{inr}(b))\end{array}}{\begin{array}{c}\Gamma, a : A \vdash d(a) \equiv f(\mathsf{inl}(a)) : D(\mathsf{inl}(a)) \\ \Gamma, b : B \vdash e(b) \equiv f(\mathsf{inr}(b)) : D(\mathsf{inr}(b))\end{array}}$$

Another example: natural numbers

- $\mathbb{N}$-formation

$$\frac{}{\Gamma \vdash \mathbb{N} \text{ TYPE}}$$

- $\mathbb{N}$-introduction

$$\frac{}{0 : \mathbb{N}} \qquad \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash s(n) : \mathbb{N}}$$

- $\mathbb{N}$-elimination

$$\frac{\Gamma \vdash z : D(0) \qquad \begin{array}{c}\Gamma, n : \mathbb{N} \vdash D(n) \\ \Gamma, n : \mathbb{N}, p : D(n) \vdash q(n, p) : D(s(n))\end{array}}{\Gamma, n : \mathbb{N} \vdash f(n) : D(n)}$$

- $\mathbb{N}$-computation

$$\frac{\Gamma \vdash z : D(0) \qquad \begin{array}{c}\Gamma, n : \mathbb{N} \vdash D(n) \\ \Gamma, n : \mathbb{N}, p : D(n) \vdash q(n, p) : D(s(n))\end{array}}{\begin{array}{c}\Gamma \vdash f(0) \equiv z : D(0) \\ \Gamma, n : \mathbb{N} \vdash f(s(n)) \equiv q(n, f(n)) : D(s(n))\end{array}}$$

The weird example: equality

- Id-formation

$$\frac{A \text{ TYPE} \qquad a : A \qquad b : A}{\mathsf{Id}_A(a, b) \text{ TYPE}}$$

- Id-introduction

$$\frac{a : A}{r_a : \mathsf{Id}_A(a, a)}$$

- Id-elimination

$$\frac{\begin{array}{c} a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash D(a, b, p) \\ a : A \vdash d(a) : D(a, a, r(a)) \end{array}}{a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash j(a, b, p) : D(a, b, p)}$$

- Id-computation

$$\frac{\begin{array}{c} a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash D(a, b, p) \\ a : A \vdash d(a) : D(a, a, r(a)) \end{array}}{a : A \vdash j(a, a, r(a)) \equiv d(a) : D(a, a, r(a))}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathrm{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{\Gamma \vdash z : D(0) \qquad \begin{array}{c}\Gamma, n : \mathbb{N} \vdash D(n) \\ \Gamma, n : \mathbb{N}, p : D(n) \vdash q(n, p) : D(sn)\end{array}}{\Gamma, n : \mathbb{N} \vdash f(n) : D(n)}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{m : \mathbb{N} \vdash z : D(0) \qquad \begin{array}{c} m : \mathbb{N}, n : \mathbb{N} \vdash D(n) \\ m : \mathbb{N}, n : \mathbb{N}, p : D(n) \vdash q(n, p) : D(sn) \end{array}}{m : \mathbb{N}, n : \mathbb{N} \vdash f(n) : D(n)}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{m : \mathbb{N} \vdash z : \mathbb{N} \qquad \begin{array}{c} m : \mathbb{N}, n : \mathbb{N} \vdash \mathbb{N} \\ m : \mathbb{N}, n : \mathbb{N}, p : \mathbb{N} \vdash q(n, p) : \mathbb{N} \end{array}}{m : \mathbb{N}, n : \mathbb{N} \vdash f(n) : \mathbb{N}}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathrm{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{m : \mathbb{N} \vdash z : \mathbb{N} \qquad \begin{array}{c} m : \mathbb{N}, n : \mathbb{N} \vdash \mathbb{N} \\ m : \mathbb{N}, n : \mathbb{N}, p : \mathbb{N} \vdash q(n, p) : \mathbb{N} \end{array}}{m : \mathbb{N}, n : \mathbb{N} \vdash \mathrm{sum}(m, n) : \mathbb{N}}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{m : \mathbb{N} \vdash m : \mathbb{N} \qquad \begin{array}{c} m : \mathbb{N}, n : \mathbb{N} \vdash \mathbb{N} \\ m : \mathbb{N}, n : \mathbb{N}, p : \mathbb{N} \vdash q(n, p) : \mathbb{N} \end{array}}{m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{\begin{array}{cc} & m : \mathbb{N}, n : \mathbb{N} \vdash \mathbb{N} \\ m : \mathbb{N} \vdash m : \mathbb{N} & m : \mathbb{N}, n : \mathbb{N}, p : \mathbb{N} \vdash sp : \mathbb{N} \end{array}}{m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathrm{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{m : \mathbb{N}, n : \mathbb{N} \vdash \mathbb{N} \qquad m : \mathbb{N} \vdash m : \mathbb{N} \qquad m : \mathbb{N}, n : \mathbb{N}, p : \mathbb{N} \vdash sp : \mathbb{N}}{m : \mathbb{N}, n : \mathbb{N} \vdash \mathrm{sum}(m, n) : \mathbb{N}}$$

We can show $n : \mathbb{N} \vdash \mathrm{Id}_{\mathbb{N}}(0 + n, n)$ or $0 + n = n$.

$$\frac{\Gamma, n : \mathbb{N} \vdash D(n) \qquad \Gamma \vdash z : D(0) \qquad \Gamma, n : \mathbb{N}, p : D(n) \vdash q(n, p) : D(sn)}{\Gamma, n : \mathbb{N} \vdash f(n) : D(n)}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{m : \mathbb{N}, n : \mathbb{N} \vdash \mathbb{N} \qquad m : \mathbb{N} \vdash m : \mathbb{N} \qquad m : \mathbb{N}, n : \mathbb{N}, p : \mathbb{N} \vdash sp : \mathbb{N}}{m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}}$$

We can show $n : \mathbb{N} \vdash \mathsf{Id}_{\mathbb{N}}(0 + n, n)$ or $0 + n = n$.

$$\frac{n : \mathbb{N} \vdash D(n) \qquad \vdash z : D(0) \qquad n : \mathbb{N}, p : D(n) \vdash q(n, p) : D(sn)}{n : \mathbb{N} \vdash f(n) : D(n)}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{m : \mathbb{N} \vdash m : \mathbb{N} \qquad \begin{array}{c} m : \mathbb{N}, n : \mathbb{N} \vdash \mathbb{N} \\ m : \mathbb{N}, n : \mathbb{N}, p : \mathbb{N} \vdash sp : \mathbb{N} \end{array}}{m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}}$$

We can show $n : \mathbb{N} \vdash \mathsf{Id}_{\mathbb{N}}(0 + n, n)$ or $0 + n = n$.

$$\frac{\vdash z : 0 + 0 = 0 \qquad \begin{array}{c} n : \mathbb{N} \vdash 0 + n = n \\ n : \mathbb{N}, p : 0 + n = n \vdash q(n, p) : 0 + sn = sn \end{array}}{n : \mathbb{N} \vdash f(n) : 0 + n = n}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathrm{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{\begin{array}{c} m : \mathbb{N}, n : \mathbb{N} \vdash \mathbb{N} \\ m : \mathbb{N} \vdash m : \mathbb{N} \qquad m : \mathbb{N}, n : \mathbb{N}, p : \mathbb{N} \vdash sp : \mathbb{N} \end{array}}{m : \mathbb{N}, n : \mathbb{N} \vdash \mathrm{sum}(m, n) : \mathbb{N}}$$

We can show $n : \mathbb{N} \vdash \mathrm{Id}_{\mathbb{N}}(0 + n, n)$ or $0 + n = n$.

$$\frac{\begin{array}{c} n : \mathbb{N} \vdash 0 + n = n \\ \vdash r_0 : 0 + 0 = 0 \qquad n : \mathbb{N}, p : 0 + n = n \vdash q(n, p) : 0 + sn = sn \end{array}}{n : \mathbb{N} \vdash f(n) : 0 + n = n}$$

Calculation: addition

We can define addition: $m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}$ using $\mathbb{N}$-elimination.

$$\frac{m : \mathbb{N} \vdash m : \mathbb{N} \qquad \begin{array}{c} m : \mathbb{N}, n : \mathbb{N} \vdash \mathbb{N} \\ m : \mathbb{N}, n : \mathbb{N}, p : \mathbb{N} \vdash sp : \mathbb{N} \end{array}}{m : \mathbb{N}, n : \mathbb{N} \vdash \mathsf{sum}(m, n) : \mathbb{N}}$$

We can show $n : \mathbb{N} \vdash \mathsf{Id}_{\mathbb{N}}(0 + n, n)$ or $0 + n = n$.

$$\frac{\vdash r_0 : 0 + 0 = 0 \qquad \begin{array}{c} n : \mathbb{N} \vdash 0 + n = n \\ n : \mathbb{N}, p : 0 + n = n \vdash sp : 0 + sn = sn \end{array}}{n : \mathbb{N} \vdash f(n) : 0 + n = n}$$

Calculation: Id is an equivalence relation

We can show that Id is symmetric:
$a : A, b : A, p : \mathsf{Id}_A(a,b) \vdash p^{-1} : \mathsf{Id}_A(b,a)$.

$$\frac{\begin{array}{c} a : A, b : A, p : \mathsf{Id}_A(a,b) \vdash D(a,b,p) \\ a : A \vdash d(a) : D(a,a,r_a) \end{array}}{a : A, b : A, p : \mathsf{Id}_A(a,b) \vdash j(a,b,p) : D(a,b,p)}$$

Calculation: Id is an equivalence relation

We can show that Id is symmetric:
$a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash p^{-1} : \mathsf{Id}_A(b, a)$.

$$\frac{\begin{array}{c} a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash \mathsf{Id}_A(b, a) \\ a : A \vdash d(a) : \mathsf{Id}_A(a, a) \end{array}}{a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash j(a, b, p) : \mathsf{Id}_A(b, a)}$$

Calculation: Id is an equivalence relation

We can show that Id is symmetric:
$a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash p^{-1} : \mathsf{Id}_A(b, a)$.

$$\frac{\begin{array}{c} a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash \mathsf{Id}_A(b, a) \\ a : A \vdash r_a : \mathsf{Id}_A(a, a) \end{array}}{a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash j(a, b, p) : \mathsf{Id}_A(b, a)}$$

Calculation: Id is an equivalence relation

We can show that Id is symmetric:
$a : A, b : A, p : \mathrm{Id}_A(a, b) \vdash p^{-1} : \mathrm{Id}_A(b, a)$.

$$\frac{\begin{array}{c} a : A, b : A, p : \mathrm{Id}_A(a, b) \vdash \mathrm{Id}_A(b, a) \\ a : A \vdash r_a : \mathrm{Id}_A(a, a) \end{array}}{a : A, b : A, p : \mathrm{Id}_A(a, b) \vdash p^{-1} : \mathrm{Id}_A(b, a)}$$

Calculation: Id is an equivalence relation

We can show that Id is symmetric:
$a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash p^{-1} : \mathsf{Id}_A(b, a)$.

$$\frac{\begin{array}{c} a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash \mathsf{Id}_A(b, a) \\ a : A \vdash r_a : \mathsf{Id}_A(a, a) \end{array}}{a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash p^{-1} : \mathsf{Id}_A(b, a)}$$

We can show that Id is transitive:
$a : A, b : A, c : A, p : \mathsf{Id}_A(a, b), q : \mathsf{Id}_A(b, c) \vdash p * q : \mathsf{Id}_A(a, c)$.

$$\frac{\begin{array}{c} a : A, b : A, c : A, p : \mathsf{Id}_A(a, b), q : \mathsf{Id}_A(b, c) \vdash \mathsf{Id}_A(a, c) \\ b : A \vdash p : \mathsf{Id}_A(a, b) \end{array}}{a : A, b : A, p : \mathsf{Id}_A(a, b) \vdash p * q : \mathsf{Id}_A(a, c)}$$

Outline

Type theory

Homotopy type theory

Directions

From sets to homotopy types

Axiom: Uniqueness of Identity Proofs

For any $p, q : a = b$, we have $p = q$.

Set model (folk)

Sets obey the rules of Martin-Löf type theory. Depedent types $a : A \vdash B(a)$ TYPE correspond to indexed families $\{B(a)\}_{a \in A}$ or functions $B \to A$.

$$\mathsf{Id}_X(x, y) = \begin{cases} * & \text{if } x = y \\ \varnothing & \text{otherwise} \end{cases}$$

Groupoid model (Hofmann & Streicher)

Groupoids obey the rules of Maritn-Löf type theory. Depedent types $a : A \vdash B(a)$ TYPE correspond to functors $A \to \mathcal{G}rp$ or isofibrations $B \to A$.

$$\mathsf{Id}_{\mathcal{G}}(X, Y) = \mathsf{hom}_{\mathbb{G}}(X, Y)$$

Voevodsky's Univalent Foundations

Univalence axiom

There are universe types, and
$(X =_U Y) \to (X \simeq Y)$ is an equivalence.

Simplicial set model

Kan complexes obey the rules of Martin-Löf type theory. Depedent types $a : A \vdash B(a)$ TYPE correspond to Kan fibrations $B \to A$.

$\mathrm{Id}_K(x, y)$ is the Kan complex of paths from x to y

Dependent sums and products

We can take the dependent sum of $a : A \vdash B(a)$ to get $\vdash \Sigma_{a:A} B(a)$. It has a canonical element for each pair $a : A$, $b : B(a)$.

We can take the dependent product of $a : A \vdash B(a)$ to get $\vdash \Pi_{a:A} B(a)$. It has a canonical element for each $a : A \vdash b(a) : B(a)$.

Types	Propositions	Sets	Program specifications
$\Sigma_{a:A} B(a)$	$\exists_{a:A} B(a)$	$\bigcup_{a:A} B(a)$	Do $B(a)$ for some a
$\Pi_{a:A} B(a)$	$\forall_{a:A} B(a)$	$\Pi_{a:A} B(a)$	Do $B(a)$ for all a

Connection with model category theory

- Id-elimination

$$\frac{\begin{array}{c} a : A, b : A, p : \mathrm{Id}_A(a,b) \vdash D(a,b,p) \\ a : A \vdash d(a) : D(a,a,r(a)) \end{array}}{a : A, b : A, p : \mathrm{Id}_A(a,b) \vdash j(a,b,p) : D(a,b,p)}$$

$$\begin{tikzcd} A \arrow[r, "d"] \arrow[d, "r"] & D \arrow[d, "\pi"] \\ \mathrm{Id}(A) \arrow[r, equal] \arrow[ur, dashed] & \mathrm{Id}(A) \end{tikzcd}$$

Theorem (Gambino-Garner)

Any model of type theory comes with a weak factorization system $(\mathcal{D}^{\boxslash}, {}^{\boxslash}(\mathcal{D}^{\boxslash}))$.

Connection with model category theory

- Id-elimination

$$\frac{a : A, b : A, p : \mathrm{Id}_A(a, b) \vdash D(a, b, p) \qquad a : A \vdash d(a) : D(a, a, r(a))}{a : A, b : A, p : \mathrm{Id}_A(a, b) \vdash j(a, b, p) : D(a, b, p)}$$

$$\begin{tikzcd}
A \arrow[r, "d"] \arrow[d, "r"] & D \arrow[d, "\pi"] \\
\mathrm{Id}(A) \arrow[r] \arrow[ur, dashed] & B
\end{tikzcd}$$

Theorem (Gambino-Garner)

Any model of type theory comes with a weak factorization system $(\mathcal{D}^{\boxslash}, {}^{\boxslash}(\mathcal{D}^{\boxslash}))$.

Outline

Type theory

Homotopy type theory

Directions

Synthetic homotopy theory

- Constructivize and computer-check classical homotopy theory.
  - Homotopy groups of spheres
  - Blakers-Massey theorem
  - Localization (Christensen)
  - Symmetry Book
- Invent homotopical mathematics
  - Higher groups (Buchholtz & Rijke)

Homotopy levels

Homotopy levels

A type is of *h-level* 0 if it is contractible and of *h*-level $n + 1$ if its identity types are of *h*-level n. (So empty or contractible types are level 1, 'sets' are level 2, 'groupoids' are level 3...)

Groups

A *group* is a set S with a specified element, unary and binary operations, and witnesses for the axioms.

$$Grp := \Sigma_{S:U, w:ishlevel(2,U), e:S, m:S\times S\to S, i:S\to S} \Pi_{r,s,t:S} m(s,e) = s$$

$$\times\ m(e,s) = s\ \times\ m(s,i(s)) = s\ \times m(i(s),s) = e\ \times\ \dots$$

The type of groups is a 3-type.

Homotopical higher category theory

We could define a *category* as having a set of objects and a set of morphisms.

Univalent category

A *univalent category* consists of a type of objects O and a set $\mathrm{hom}(X, Y)$ for every $X, Y : O$

- satisfying the axioms of a category, and
- such that the $(X = Y) \to (X \cong Y)$ is an equivalence.

Under the simplicial set interpretation, these correspond to complete Segal spaces.

Proposotion

For two univalent categories $\mathcal{C}, \mathcal{D}$, the morphism $(\mathcal{C} = \mathcal{D}) \to \mathcal{C} \simeq \mathcal{D}$ is an equivalence.

Flavours of type theory

- Cubical type theory
  - Semantics in cubical sets
  - The Univalence Axiom is a theorem (more constructive).
- Type theory for $(\infty, 1)$-categories
- Directed homotopy type theory
- Two-level type theory
- Propositional type theory
- Modal type theories

Initiality conjecture

Conjecture

A type theory T is the initial object of its models.

- Done for individual basic type theories
- Define models for different type theories by hand
- Problem is largely to define the terms in the conjecture.