Emily Riehl

Johns Hopkins University

A synthetic theory of ∞-categories in homotopy type theory

joint with Michael Shulman

Association for Symbolic Logic, Joint Mathematics Meetings

Motivation I

CAHIERS DE TOPOLOGIE
ET GÉOMÉTRIE DIFFÉRENTIELLE
CATÉGORIQUES

VOL. XXXII-1 (1991)

∞-GROUPOIDS AND HOMOTOPY TYPES

by M.M. KAPRANOV and V.A. VOEVODSKY

RÉSUMÉ. Nous présentons une description de la categorie homotopique des CW-complexes en termes des ∞-groupoïdes. La possibilité d'une telle description a été suggérée par A. Grothendieck dans son mémoire "A la poursuite des champs".

It is well-known [GZ] that CW-complexes X such that $\pi_i(X,x) = 0$ for all $i \geq 2$, $x \in X$, are described, at the homotopy level, by groupoids. A. Grothendieck suggested, in his unpublished memoir [Gr], that this connection should have a higher-dimensional generalisation involving polycategories. viz. polycategorical analogues of groupoids. It is the purpose of this paper to establish such a generalisation.

- 15 statements =
  4 theorems
  + 9 propositions
  + 1 lemma
  + 1 corollary
- 5 short "obvious" proofs + 3 proofs

- Carlos Simpson's "Homotopy types of strict 3-groupoids" (1998) shows that the 3-type of S^2 can't be realized by a strict 3-groupoid — contradicting the last corollary
- But no explicit mistake was found. Voevodsky: "I was sure that we were right until the fall of 2013 (!!)"

A sociological problem

MATHEMATICS

The Origins and Motivations of Univalent Foundations

A Personal Mission to Develop Computer Proof Verification to Avoid Mathematical Mistakes

By Vladimir Voevodsky • Published 2014

"A technical argument by a trusted author, which is hard to check and looks similar to arguments known to be correct, is hardly ever checked in detail."

Motivation II

The Yoneda lemma. An object of a category is determined up to canonical isomorphism by the network of relationships that the object has with all the other objects in the category.

Corollary. All theorems in category theory.

The Yoneda lemma for ordinary 1-categories is proven on:

- page 61/314 of *Categories for the Working Mathematician*
- page 57/240 of *Category Theory in Context*

The Yoneda lemma for ∞-categories is proven on:

- page 269/416 in a series of papers by Riehl–Verity
- page 47/78 of Riehl–Shulman, A type theory for synthetic ∞-categories, Higher Structures 1(1):116–193, 2017.

Motivation III

Why do I study category theory?

— I find category theoretic arguments to be aesthetically appealing.

What draws me to homotopy type theory?

— I find homotopy type theoretic arguments to be aesthetically appealing.

Plan

1. The right way to think about equality
2. Homotopy type theory
3. A type theory for ∞-categories
4. Segal, Rezk, and discrete types
5. The synthetic theory of ∞-categories

Main takeaway:

- path induction (substitution for equality): the identity type family is freely generated by the reflexivity term
- arrow induction (Yoneda lemma): the hom type family is freely generated by the identity arrow

1

The right way to think about equality

What is the correct way to think about equality?

> "The heart and soul of much mathematics consists of the fact that the 'same' object can be presented to us in different ways."
> — Barry Mazur "When is one thing equal to some other thing?"

Not what it is but what it does.

Principle of substitution. To prove that every x, y with $x = y$ have property P, it suffices to:

- Prove that every pair x, x (for which $x = x$) has property P.

Example.

- To prove symmetry — for all x, y if $x = y$ then $y = x$ — apply the principle of substitution to the property $P(x, y) := y = x$. Now it's enough to show that for all x then $x = x$, which is true by reflexivity.
- Transitivity — for all x, y, z if $x = y$ and $y = z$ then $x = z$ — can be deduced similarly from the principle of substitution.

2

Homotopy type theory

Homotopy type theory

Homotopy type theory is:

- a formal system for mathematical constructions and proofs
- in which the basic objects, types, may be regarded as "spaces" or ∞-groupoids
- and all constructions are automatically "continuous" or equivalence-invariant.

$$\text{Homotopy type theory is } \begin{cases} \text{homotopy (type theory)} \\ \text{(homotopy type) theory} \end{cases}$$

Types A can be regarded simultaneously as both mathematical constructions and mathematical assertions; accordingly, a term $a : A$ can be regarded as a proof of the proposition A. But it's somewhat misleading to think of propositions as types, because types may have non-trivial higher dimensional structure.

Types, terms, and type constructors

Homotopy type theory has:

- types A, B
- terms $x : A$, $y : B$
- dependent types $x : A \vdash B(x)$, $x, y : A \vdash B(x, y)$ including in particular identity types $x, y : A \vdash x =_A y$.

Type constructors build new types and terms from given ones:

- products $A \times B$, coproducts $A + B$, function types $A \to B$,
- dependent sums $\sum_{x:A} B(x)$, dependent products $\prod_{x:A} B(x)$.

Each type constructor comes with rules:

(i) formation: a way to construct new types

(ii) introduction: ways to construct terms of these types

(iii) elimination: ways to use them to construct other terms

(iv) computation: what happens when we follow (ii) by (iii)

The Curry-Howard-Voevodsky correspondence

type theory	set theory	logic	homotopy theory
A	set	proposition	space
$x : A$	element	proof	point
$\emptyset, 1$	$\emptyset, \{\emptyset\}$	$\bot, \top$	$\emptyset, *$
$A \times B$	set of pairs	A and B	product space
$A + B$	disjoint union	A or B	coproduct
$A \to B$	set of functions	A implies B	function space
$x : A \vdash B(x)$	family of sets	predicate	fibration
$x : A \vdash b : B(x)$	fam. of elements	conditional proof	section
$\prod_{x:A} B(x)$	product	$\forall x.B(x)$	space of sections
$\sum_{x:A} B(x)$	disjoint sum	$\exists x.B(x)$	total space
$p : x =_A y$	$x = y$	proof of equality	path from x to y
$\sum_{x,y:A} x =_A y$	diagonal	equality relation	path space for A

Identity types

Formation and introduction rules for identity types

$$\frac{x, y : A}{x =_A y \text{ type}} \qquad\qquad \frac{x : A}{\mathsf{refl}_x : x =_A x}$$

Semantics
$$\left\{ \begin{array}{ccc} & & \sum_{x,y:A} x =_A y \\ & \overset{\lambda x.\mathsf{refl}_x}{\nearrow} & \downarrow \\ A & \xrightarrow[\Delta]{} & A \times A \end{array} \right.$$

Hence $\sum_{x,y:A} x =_A y$ is interpreted as the path space of A and a term $\mathsf{p} : x =_A y$ may be thought of as a path from x to y in A.

Path induction

The identity type family is freely generated by the terms $\mathsf{refl}_x : x =_A x$.

Path induction. If $B(x, y, p)$ is a type family dependent on $x, y : A$ and $p : x =_A y$, then to prove $B(x, y, p)$ it suffices to assume y is x and p is refl_x. I.e., there is a function

$$\mathsf{path\text{-}ind} : \left(\prod_{x:A} B(x, x, \mathsf{refl}_x) \right) \to \left(\prod_{x,y:A} \prod_{p:x=_A y} B(x, y, p) \right).$$

Path induction expresses the elimination rule for Per Martin-Löf's identity type — an enhanced version of:

Principle of substitution. To prove that every x, y with $x = y$ have property P, it suffices to:

- Prove that every pair x, x (for which $x = x$) has property P.

The ∞-groupoid of paths

Theorem (Lumsdaine, Garner–van den Berg). The terms belonging to the iterated identity types of any type A form an ∞-groupoid.

The ∞-groupoid structure of A has

- terms $x : A$ as objects
- paths $p : x =_A y$ as 1-morphisms
- paths of paths $h : p =_{x =_A y} q$ as 2-morphisms, ...

The required structures are proven from the path induction principle:

- constant paths (reflexivity) $\mathsf{refl}_x : x = x$
- reversal (symmetry) $p : x = y$ yields $p^{-1} : y = x$
- concatenation (transitivity) $p : x = y$ and $q : y = z$ yield $q * p : x = z$

and furthermore

- concatenation is associative
- the associators are coherent, ...

3

A type theory for ∞-categories

The intended model

$\mathcal{S}\mathrm{et}^{\Delta^{op}\times\Delta^{op}}$	$\supset$	$\mathcal{R}\mathrm{eedy}$	$\supset$	$\mathcal{S}\mathrm{egal}$	$\supset$	$\mathcal{R}\mathrm{ezk}$
$\parallel$		$\parallel$		$\parallel$		$\parallel$
bisimplicial sets		types		types with composition		types with composition & univalence

Theorem (Shulman). Homotopy type theory is modeled by the category of Reedy fibrant bisimplicial sets.

Theorem (Rezk). ∞-categories are modeled by Rezk spaces aka complete Segal spaces.

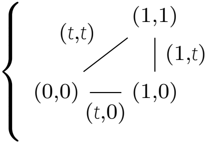
Shapes in the theory of the directed interval

Our types may depend on other types and also on shapes $\Phi \subset \mathbb{2}^n$, polytopes embedded in a directed cube, defined in a language

$$\top, \bot, \wedge, \vee, \equiv \qquad \text{and} \qquad 0, 1, \leq$$

satisfying intuitionistic logic and strict interval axioms.

$$\Delta^n := \{(t_1, \ldots, t_n) : \mathbb{2}^n \mid t_n \leq \cdots \leq t_1\} \qquad \text{e.g.} \qquad \Delta^1 := \mathbb{2}$$

$\Delta^2 :=$ 

$$\partial\Delta^2 := \{(t_1, t_2) : \mathbb{2}^2 \mid (t_2 \leq t_1) \wedge ((0 = t_2) \vee (t_2 = t_1) \vee (t_1 = 1))\}$$
$$\Lambda^2_1 := \{(t_1, t_2) : \mathbb{2}^2 \mid (t_2 \leq t_1) \wedge ((0 = t_2) \vee (t_1 = 1))\}$$

Because $\phi \wedge \psi$ implies ϕ, there are shape inclusions $\Lambda^2_1 \subset \partial\Delta^2 \subset \Delta^2$.

Extension types

shape inclusion: $\Phi := \{t \in 2^n \mid \phi\}$ and $\Psi = \{t \in 2^n \mid \psi\}$ so that ϕ implies ψ, i.e., so that $\Phi \subset \Psi$.

Formation rule for extension types

$$\frac{\Phi \subset \Psi \text{ shape} \qquad A \text{ type} \qquad a : \Phi \to A}{\left\langle \begin{array}{ccc} \Phi & \xrightarrow{a} & A \\ \downarrow & \nearrow & \\ \Psi & & \end{array} \right\rangle \text{ type}}$$

A term $f : \left\langle \begin{array}{ccc} \Phi & \xrightarrow{a} & A \\ \downarrow & \nearrow & \\ \Psi & & \end{array} \right\rangle$ defines

$$f : \Psi \to A \text{ so that } f(t) \equiv a(t) \text{ for } t : \Phi.$$

4

Segal, Rezk, and discrete types

Hom types

The hom type for A depends on two terms in A:

$$x, y : A \vdash \hom_A(x, y)$$

$$\frac{\partial\Delta^1 \subset \Delta^1 \text{ shape} \qquad A \text{ type} \qquad [x, y] : \partial\Delta^1 \to A}{\hom_A(x, y) := \left\langle \begin{array}{ccc} \partial\Delta^1 & \xrightarrow{[x,y]} & A \\ \downarrow & \nearrow & \\ \Delta^1 & & \end{array} \right\rangle \text{ type}}$$

A term $f : \hom_A(x, y)$ defines an arrow in A from x to y.

Segal types have unique binary composites

A type A is Segal iff every composable pair of arrows has a unique composite, i.e., for every $f : \hom_A(x,y)$ and $g : \hom_A(y,z)$ the type

$$\left\langle \begin{array}{ccc} \Lambda^2_1 & \xrightarrow{[f,g]} & A \\ \downarrow & \nearrow & \\ \Delta^2 & & \end{array} \right\rangle \quad \text{is contractible.}$$

Notation. Let $\text{comp}_{g,f} : \left\langle \begin{array}{ccc} \Lambda^2_1 & \xrightarrow{[f,g]} & A \\ \downarrow & \nearrow & \\ \Delta^2 & & \end{array} \right\rangle$ denote the unique inhabitant and write $g \circ f : \hom_A(x,z)$ for its inner face, *the* composite of f and g.

Identity arrows

For any $x : A$, the constant function defines a term

$$\mathrm{id}_x :\equiv \lambda t.x : \hom_A(x,x) :\equiv \left\langle \begin{array}{ccc} \partial\Delta^1 & \xrightarrow{[x,x]} & A \\ \downarrow & \nearrow & \\ \Delta^1 & & \end{array} \right\rangle,$$

which we denote by id_x and call the identity arrow.

For any $f : \hom_A(x,y)$ in a Segal type A, the term

$$\lambda(s,t).f(t) : \left\langle \begin{array}{ccc} \Lambda^2_1 & \xrightarrow{[\mathrm{id}_x, f]} & A \\ \downarrow & \nearrow & \\ \Delta^2 & & \end{array} \right\rangle$$

witnesses the unit axiom $f = f \circ \mathrm{id}_x$.

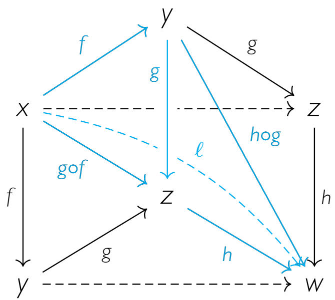
Associativity of composition

Let A be a Segal type with arrows

$$f : \hom_A(x, y), \quad g : \hom_A(y, z), \quad h : \hom_A(z, w).$$

Prop. $$h \circ (g \circ f) = (h \circ g) \circ f.$$

Proof: Consider the composable arrows in the Segal type $\Delta^1 \to A$:



Composing defines a term in the type $\Delta^2 \to (\Delta^1 \to A)$ which yields a term $\ell : \hom_A(x, w)$ so that $\ell = h \circ (g \circ f)$ and $\ell = (h \circ g) \circ f$.

Isomorphisms

An arrow $f : \hom_A(x,y)$ in a Segal type is an isomorphism if it has a two-sided inverse $g \colon \hom_A(y,x)$. However, the type

$$\sum_{g \colon \hom_A(y,x)} (g \circ f = \mathrm{id}_x) \times (f \circ g = \mathrm{id}_y)$$

has higher-dimensional structure and is *not* a proposition. Instead define

$$\mathrm{isiso}(f) := \left(\sum_{g \colon \hom_A(y,x)} g \circ f = \mathrm{id}_x \right) \times \left(\sum_{h \colon \hom_A(y,x)} f \circ h = \mathrm{id}_y \right).$$

For $x, y : A$, the type of isomorphisms from x to y is:

$$x \cong_A y := \sum_{f : \hom_A(x,y)} \mathrm{isiso}(f).$$

Rezk types

By path induction, to define a map

$$\text{path-to-iso} : (x =_A y) \to (x \cong_A y)$$

for all $x, y : A$ it suffices to define

$$\text{path-to-iso}(\text{refl}_x) :\equiv \text{id}_x.$$

A Segal type A is Rezk if every isomorphism is an identity, i.e., if the map

$$\text{path-to-iso} : \prod_{x,y:A} (x =_A y) \to (x \cong_A y)$$

is an equivalence.

Discrete types

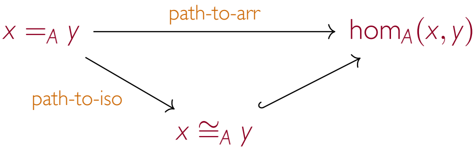
Similarly by path induction define

$$\text{path-to-arr} \colon \prod_{x,y:A} (x =_A y) \to \hom_A(x,y) \quad \text{by} \quad \text{path-to-arr}(\text{refl}_x) :\equiv \text{id}_x.$$

A type A is discrete if path-to-arr is an equivalence.

Prop. A type is discrete if and only if it is Rezk and all of its arrows are isomorphisms. Thus, if the Rezk types are ∞-categories, then the discrete types are ∞-groupoids.

Proof:



5

The synthetic theory of ∞-categories

Covariant type families

A type family $x : A \vdash B(x)$ over a Segal type A is covariant if for every $f : \hom_A(x, y)$ and $u : B(x)$ there is a unique lift of f with domain u.

Notation. The codomain of the unique lift defines a term $f_* u : B(y)$.

Prop. For $u : B(x)$, $f : \hom_A(x, y)$, and $g : \hom_A(y, z)$,

$$g_*(f_* u) = (g \circ f)_* u \qquad \text{and} \qquad (\mathrm{id}_x)_* u = u.$$

Prop. If $x : A \vdash B(x)$ is covariant then for each $x : A$ the fiber $B(x)$ is discrete. Thus covariant type families are fibered in ∞-groupoids.

Prop. Fix $a : A$. The type family $x : A \vdash \hom_A(a, x)$ is covariant.

The Yoneda lemma

Let $x : A \vdash B(x)$ be a covariant family over a Segal type and fix $a : A$.

Yoneda lemma. The maps

$$\text{ev-id} := \lambda\phi.\phi(a, \mathrm{id}_a) : \left(\prod_{x:A} \hom_A(a, x) \to B(x)\right) \to B(a)$$

and

$$\text{yon} := \lambda u.\lambda x.\lambda f.f_* u : B(a) \to \left(\prod_{x:A} \hom_A(a, x) \to B(x)\right)$$

are inverse equivalences.

Corollary. A natural *isomorphism* $\phi : \prod_{x:A} \hom_A(a, x) \cong \hom_A(b, x)$ induces an *identity* $\text{ev-id}(\phi) : b =_A a$ if the type A is Rezk.

The dependent Yoneda lemma

Yoneda lemma. If A is a Segal type and $B(x)$ is a covariant family dependent on $x : A$, then evaluation at (a, id_a) defines an equivalence

$$\text{ev-id} : \left(\prod_{x:A} \hom_A(a, x) \to B(x) \right) \to B(a)$$

The Yoneda lemma is a "directed" version of the "transport" operation for identity types, suggesting a dependently-typed generalization analogous to the full induction principle for identity types.

Dependent Yoneda lemma. If A is a Segal type and $B(x, y, f)$ is a covariant family dependent on $x, y : A$ and $f : \hom_A(x, y)$, then evaluation at (x, x, id_x) defines an equivalence

$$\text{ev-id} : \left(\prod_{x,y:A} \prod_{f:\hom_A(x,y)} B(x, y, f) \right) \to \prod_{x:A} B(x, x, \mathrm{id}_x)$$

Dependent Yoneda is directed path induction

Takeaway: the dependent Yoneda lemma is directed path induction.

Path induction. If $B(x, y, p)$ is a type family dependent on $x, y : A$ and $p : x =_A y$, then to prove $B(x, y, p)$ it suffices to assume y is x and p is refl_x. I.e., there is a function

$$\text{path-ind} : \left(\prod_{x:A} B(x, x, \mathsf{refl}_x) \right) \to \left(\prod_{x,y:A} \prod_{p:x=_A y} B(x, y, p) \right).$$

Arrow induction. If $B(x, y, f)$ is a covariant family dependent on $x, y : A$ and $f : \hom_A(x, y)$ and A is Segal, then to prove $B(x, y, f)$ it suffices to assume y is x and f is id_x. I.e., there is a function

$$\text{id-ind} : \left(\prod_{x:A} B(x, x, \mathrm{id}_x) \right) \to \left(\prod_{x,y:A} \prod_{f:\hom_A(x,y)} B(x, y, f) \right).$$

References

For considerably more, see:

Emily Riehl and Michael Shulman, A type theory for synthetic ∞-categories, Higher Structures 1(1):116–193, 2017. arXiv:1705.07442

To explore homotopy type theory:

Homotopy Type Theory: Univalent Foundations of Mathematics, https://homotopytypetheory.org/book/

Michael Shulman, Homotopy type theory: the logic of space, arXiv:1703.03007

Thank you!