Emily Riehl

Johns Hopkins University

∞-category theory for undergraduates

Theories of Everything
with Curt Jaimungal

1

What is category theory for?

Abstraction and understanding

- Shortly before his untimely 1832 death, Évariste Galois solved the problem of characterizing polynomial equations whose solutions may be expressed by a formula involving only integers, roots, and the four basic arithmetic operations.
- His paper was initially rejected and remained unpublished until 1846 when Joseph Liouville published it with some further explanations.
- But even Liouville missed the point of what is now called Galois theory: that the solutions of a polynomial equation define a group.
- Galois theory—widely misunderstood by the top mathematicians of the day—is now regularly taught to undergraduates around the world.

Moral: New abstractions can make complicated mathematics easier to understand.

What is category theory for?

Corollary. For any function $f\colon A \to B$, the inverse image function $f^{-1}\colon \mathfrak{P}(B) \to \mathfrak{P}(A)$ between the powersets of A and B preserves both unions and intersections, while the direct image function $f_*\colon \mathfrak{P}(A) \to \mathfrak{P}(B)$ preserves only unions.

Corollary. For any vector spaces U, V, and W, $U \otimes (V \oplus W) \cong (U \otimes V) \oplus (U \otimes W)$.

Corollary. The free group on the set $X \sqcup Y$ is the free product of the free group on X and the free group on Y.

All of these results have the same proof and are true for the same reason:

Theorem. Left adjoint functors preserve coproducts,
while right adjoint functors preserve products.

A categorical proof of a result in linear algebra

Corollary. For any vector spaces U, V, and W, $U \otimes (V \oplus W) \cong (U \otimes V) \oplus (U \otimes W)$.

The proof uses the Yoneda lemma: to show $U \otimes (V \oplus W) \cong (U \otimes V) \oplus (U \otimes W)$ it suffices to give natural correspondences between linear maps to another vector space X:

$$U \otimes (V \oplus W) \to X \qquad \leftrightsquigarrow \qquad (U \otimes V) \oplus (U \otimes W) \to X.$$

Letting X equal $U \otimes (V \oplus W)$ or $(U \otimes V) \oplus (U \otimes W)$, the identity functions define linear maps in each direction between these vector spaces, which are inverses (by naturality).

Proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{U \otimes (V \oplus W) \to X}{V \oplus W \to \mathsf{Lin}(U, X)}\ \text{BY CURRYING}}{V \to \mathsf{Lin}(U, X) \quad \text{and} \quad W \to \mathsf{Lin}(U, X)}\ \text{BY CASES}}{U \otimes V \to X \quad \text{and} \quad U \otimes W \to X}\ \text{BY UNCURRYING}}{(U \otimes V) \oplus (U \otimes W) \to X}\ \text{BY CASES}$$

Categories and isomorphisms

We now isolate the abstract core of the proof we just gave:

Vector spaces and linear maps define a category, which is given by:

- a collection of objects U, V, W, …,
- a collection of arrows $f\colon U \to V$, $g\colon V \to W$ with source and target objects, so that
- each object V has a specified identity arrow $\mathrm{id}_V\colon V \to V$, and
- any composable arrows $f\colon U \to V$ and $g\colon V \to W$ have a composite $g \circ f\colon U \to W$,
- so that composition is associative and unital.

Two objects V and W in a category are isomorphic, denoted $V \cong W$, if there exist:

- arrows $f\colon V \to W$ and $g\colon W \to V$ so that
- the composites equal the appropriate identities: $g \circ f = \mathrm{id}_V$ and $f \circ g = \mathrm{id}_W$.

The Yoneda lemma

The Yoneda lemma. Two objects A and B in a category are isomorphic if and only if

- for all objects X, the sets of arrows

$$\mathrm{Hom}(A, X) := \{f \colon A \to X\} \quad \text{and} \quad \mathrm{Hom}(B, X) := \{g \colon B \to X\}$$

  are isomorphic and moreover
- the isomorphisms $\mathrm{Hom}(A, X) \cong \mathrm{Hom}(B, X)$ are "natural" in the sense of commuting with composition with any arrow $h \colon X \to Y$.

Proof ($\Leftarrow$): Suppose $\mathrm{Hom}(A, X) \cong \mathrm{Hom}(B, X)$ for all X. Taking $X = A$ and $X = B$, we use the bijections and identities to define arrows $g \colon B \to A$ and $f \colon A \to B$:

$$\begin{array}{ccc} \mathrm{Hom}(A, A) & \cong & \mathrm{Hom}(B, A) \\ \cup & & \cup \\ \mathrm{id}_A & \longmapsto & g \end{array} \qquad \begin{array}{ccc} \mathrm{Hom}(A, B) & \cong & \mathrm{Hom}(B, B) \\ \cup & & \cup \\ f & \longmapsto & \mathrm{id}_B \end{array}$$

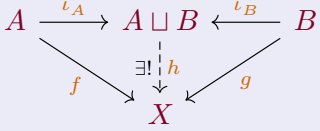
By naturality—left as an exercise—they're inverses: $g \circ f = \mathrm{id}_A$ and $f \circ g = \mathrm{id}_B$. □

Coproducts

Two objects A and B in a category admit a coproduct if there is another object and maps as displayed:

$$A \xrightarrow{\iota_A} A \sqcup B \xleftarrow{\iota_B} B$$

so that any other diagram of this shape factors uniquely through the coproduct:



In other words, arrows $h\colon A \sqcup B \to X$ are uniquely defined by cases from two arrows $f\colon A \to X$ and $g\colon B \to X$, via the isomorphism defined by composing with ι_A and ι_B.

$$\begin{array}{ccc} \mathsf{Hom}(A \sqcup B, X) & \cong & \mathsf{Hom}(A, X) \times \mathsf{Hom}(B, X) \\ \cup & & \cup \\ h & \longmapsto & (h \circ \iota_A, h \circ \iota_B) \end{array}$$

Example. For vector spaces V and W, the direct sum $V \oplus W$ defines their coproduct.

Adjunctions

A functor is an arrow between categories $F\colon \mathsf{C} \to \mathsf{D}$, sending objects and arrows in C to objects and arrows in D preserving source, target, identities, and composition.

A pair of functors $F\colon \mathsf{C} \to \mathsf{D}$ and $U\colon \mathsf{D} \to \mathsf{C}$ define an adjunction if there is a natural bijection between arrows for all C in C and all D in D:

$$FC \to D \qquad \leftrightsquigarrow \qquad C \to UD$$

defining a natural isomorphism

$$\mathsf{Hom}_{\mathsf{D}}(FC, D) \cong \mathsf{Hom}_{\mathsf{C}}(C, UD).$$

Example. For a vector space U, there is an adjunction defined by the tensor product with U and the vector space of linear maps out of U:

$$U \otimes V \to W \qquad \leftrightsquigarrow \qquad V \to \mathsf{Lin}(U, W)$$

A categorical proof of a result in category theory

Theorem. Left adjoints $F\colon \mathsf{C} \to \mathsf{D}$ preserve coproducts $A \sqcup B$: $F(A \sqcup B) \cong FA \sqcup FB$.

Proof: By the Yoneda lemma, to show $F(A \sqcup B) \cong FA \sqcup FB$ it suffices to give a natural isomorphism for any object X of D:

$$\mathsf{Hom}_{\mathsf{D}}(F(A \sqcup B), X) \cong \mathsf{Hom}_{\mathsf{D}}(FA \sqcup FB, X).$$

$$\dfrac{\dfrac{\dfrac{F(A \sqcup B) \to X}{A \sqcup B \to UX}\ \text{BY ADJUNCTION}}{A \to UX \quad \text{and} \quad B \to UX}\ \text{BY CASES}}{\dfrac{FA \to X \quad \text{and} \quad FB \to X}{FA \sqcup FB \to X}\ \text{BY CASES}}\ \text{BY ADJUNCTION}$$

defining a natural isomorphism

$$\begin{aligned}\mathsf{Hom}_{\mathsf{D}}(F(A \sqcup B), X) &\cong \mathsf{Hom}_{\mathsf{C}}(A \sqcup B, UX) \cong \mathsf{Hom}_{\mathsf{C}}(A, UX) \times \mathsf{Hom}_{\mathsf{C}}(B, UX) \\ &\cong \mathsf{Hom}_{\mathsf{D}}(FA, X) \times \mathsf{Hom}_{\mathsf{D}}(FB, X) \cong \mathsf{Hom}_{\mathsf{D}}(FA \sqcup FB, X). \qquad \square\end{aligned}$$

Abstraction and understanding

Moral: New abstractions can make complicated mathematics easier to understand. New abstractions can also make complicated mathematics easier to communicate.

Abstractions from category theory are used to:

- clarify arguments (removing inessential details)
- generalize constructions and proofs to other settings
- upload more complicated mathematical ideas into one's working memory

Category theory defines the language of much of 20th century mathematics. Modern algebraic geometry, algebraic topology, representation theory, and certain aspects of mathematical physics are all expressed in the language of category theory.

Without categorical language, we couldn't (easily) define spectral sequences, simplicial sets, cohomology theories, schemes, topological quantum field theories, ...

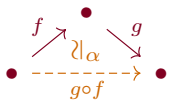
2

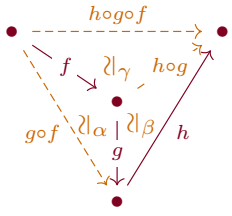
What is ∞-category theory for?

What are ∞-categories?

> *[A category] frames a possible template for any mathematical theory: the theory should have nouns and verbs, i.e., objects, and morphisms, and there should be an explicit notion of composition related to the morphisms.*
>
> *—Barry Mazur, "When is one thing equal to some other thing?"*

An ∞-category frames a template with nouns, verbs, adjectives, adverbs, pronouns, prepositions, conjunctions, interjections,... which has:

- objects $\bullet$ and 1-morphisms between them $\bullet \longrightarrow \bullet$
- composition witnessed by invertible 2-morphisms 

- associativity 

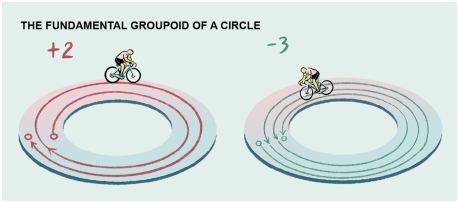
witnessed by invertible 3-morphisms

with these witnesses coherent up to invertible morphisms all the way up.

Example: homotopy types as (higher) groupoids

The fundamental groupoid of a topological space is the category whose

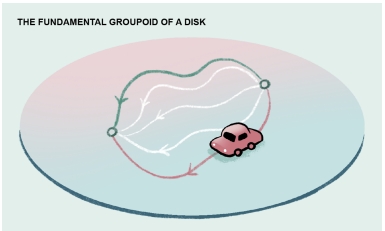
- objects are points in the space
- arrows are paths in the space up to based homotopy



Images by Matteo Farinella

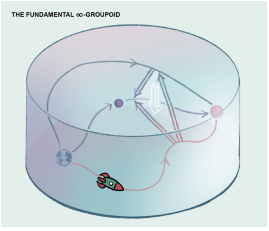
Problem: the fundamental groupoid, while easy to define, does not see "higher structure":

for all $n > 1$, the fundamental groupoid of S^n is trivial!



Example: homotopy types as ∞-groupoids

The full homotopy type of a topological space is captured by its fundamental ∞-groupoid whose



- objects are points, 1-morphisms are paths,
- 2-morphisms are homotopies between paths,
- 3-morphisms are homotopies between homotopies, ...

The quotient homotopy category recovers the fundamental groupoid of points and based homotopy classes of paths.

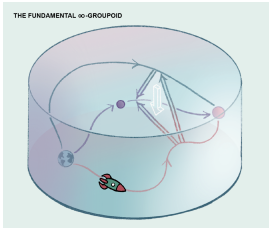
Grothendieck's Homotopy Hypothesis:
The fundamental ∞-groupoid defines an equivalence between spaces (up to weak homotopy equivalence) and ∞-groupoids (up to equivalence).

Scholze: an ∞-groupoid is an anima, the "soul" of a space.

∞-categories and their quotient 1-categories

The fundamental ∞-groupoid of a topological space is the ∞-category whose

- objects are points, 1-morphisms are paths,
- 2-morphisms are homotopies between paths,
- 3-morphisms are homotopies between homotopies, ...



The quotient homotopy category recovers the fundamental groupoid of points and based homotopy classes of paths. Similarly:

- The derived category of a ring is the homotopy category of the ∞-category of chain complexes.
- The category of closed n-manifolds and diffeomorphism classes of cobordisms is the homotopy category of the ∞-category of closed n-manifolds and cobordisms.

Here "∞-category" is a nickname for the $n = 1$ special case of an (∞, n)-category, a weak infinite dimensional category in which all morphisms above dimension n are invertible (for fixed $0 \leq n \leq \infty$).

What is ∞-category theory for?

A powerful idea in mathematics is to turn geometry into algebra, producing perhaps a cochain complex of abelian groups:

$$C^\bullet := \; C^0 \xrightarrow{d^0} C^1 \xrightarrow{d^1} C^2 \xrightarrow{d^2} C^3 \xrightarrow{d^3} \cdots$$

The fundamental geometric information is captured in the cohomology groups $H^n C^\bullet$, which motivates the definition of the derived category whose:

- objects are cochain complexes and
- arrows are maps of cochain complex, with the quasi-isomorphisms—those maps that induce isomorphisms on all cohomology groups—regarded as isomorphisms.

Problem: Important constructions, such as the construction of the mapping cone of an arrow, do not define functors on the derived category!

Solution: Replace the derived category by the ∞-category of cochain complexes, where the mapping cone is functorial and indeed easily definable via a colimit.

Rebuilding the pragmatic foundations for higher structures

I am pretty strongly convinced that there is an ongoing reversal in the collective consciousness of mathematicians: the homotopical picture of the world becomes the basic intuition, and if you want to get a discrete set, then you pass to the set of connected components of a space defined only up to homotopy ... Cantor's problems of the infinite recede to the background: from the very start, our images are so infinite that if you want to make something finite out of them, you must divide them by another infinity.

— *Yuri Manin "We do not choose mathematics as our profession, it chooses us: Interview with Yuri Manin" by Mikhail Gelfand*

∞-categories in set theory

Essentially, ∞-categories are 1-categories in which all the sets have been replaced by ∞-groupoids aka homotopy types aka anima:

$$\text{sets} :: \infty\text{-groupoids}$$
$$\text{categories} :: \infty\text{-categories}$$

Where

- categories have sets of objects, ∞-categories have ∞-groupoids of objects, and
- categories have hom-sets, ∞-categories have ∞-groupoidal mapping spaces.

While the axioms that turn a directed graph into a category are expressed in the language of set theory — a category has a composition function satisfying axioms expressed in first-order logic with equality — composition in an ∞-category, as a morphism between ∞-groupoids, isn't a "function" in the traditional sense (since homotopy types do not have underlying sets of points).

This is why ∞-categories are so difficult to model within set theory.

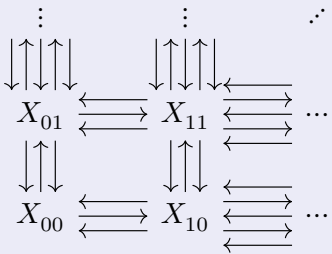
Definitions of ∞-categories

∞-categories are 1-categories in which all the sets have been replaced by ∞-groupoids.

An ∞-category is presented by a quasi-category, which is a simplicial set

in which every inner horn has a filler.

An ∞-category is presented by a complete Segal spaces, which is a bisimplicial set



that is Reedy fibrant and in which the Segal and completeness maps are equivalences.

Challenge: everything takes too long to make technically precise

Problem: ∞-category theory is a prerequisite for understanding the literature in a variety of cutting-edge areas of mathematics:

- Blumberg, Gepner, and Tabuada "A universal characterization of higher algebraic K-theory" 2013
- Gaitsgory and Rozenblyum "A study in derived algebraic geometry" 2017
- Nikolaus and Scholze "On topological cyclic homotopy" 2018
- Nadler and Tanaka "A stable ∞-category of Lagrangian cobordisms" 2020
- Bauer, Burke, and Ching "Tangent ∞-categories and Goodwillie calculus" 2023
- Fargues and Scholze "Geometrization of the local Langlands correspondence" 2024

Where will researchers in these areas find the time to learn ∞-category theory?

How will we teach ∞-category theory to undergraduates?

Dream from the future

∞-category theory would be easier to explain if the foundations of mathematics — set theory and logic — weren't so far away.

Thesis statement: If the foundations of mathematics were something more like homotopy type theory, we could teach ∞-category theory to undergraduates.

Plan:

§1 What is category theory for?

§2 What is ∞-category theory for?

§3 An informal introduction to homotopy type theory

§4 ∞-category theory for undergraduates

§5 ∞-category theory for computers

3

An informal introduction to 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.

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

Types A are used to describe mathematical structures but are also used to encode mathematical assertions. In the former case, a term $a : A$ constructs a particular instance of that structure, while in the latter case a term $a : A$ may be regarded as a proof of the proposition A.

This latter point of view is called propositions as types though that idea is somewhat misleading, because types may have non-trivial higher dimensional structure.

Types, terms, and type constructors

Homotopy type theory has:

- types A, B; e.g., $\mathbb{N}$, $\mathbb{R}$, Group
- terms $x : A$, $y : B$; e.g., $17 : \mathbb{N}$, $\sqrt{2} : \mathbb{R}$, $K_4 : \text{Group}$
- dependent types $x : A \vdash B(x)$, $x : A, y : B(x) \vdash C(x, y)$;
  e.g., $n : \mathbb{N} \vdash \mathbb{R}^n$, $n : \mathbb{N} \vdash \text{Field}_{\text{char}(n)}$, $G : \text{Group}, k : \text{Field} \vdash \text{Rep}_k(G)$
- dependent terms $x : A \vdash b_x : B(x)$, $x : A, y : B(x) \vdash c_{x,y} : C(x, y)$;
  e.g., $n : \mathbb{N} \vdash \vec{0}^n : \mathbb{R}^n$, $n : \mathbb{N} \vdash \mathbb{Z}/n : \text{Ring}_{\text{char}(n)}$, $G : \text{Group}, k : \text{Field} \vdash k[G] : \text{Rep}_k(G)$

Type constructors build new types and terms from given ones:

- products $A \times B$, coproducts $A + B$, function types $A \to B$, identity types $x, y : A \vdash x =_A y$, 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)

Product types and function types

Product types are governed by the rules

- $\times$-form: given types A and B there is a type $A \times B$
- $\times$-intro: given terms $a : A$ and $b : B$ there is a term $(a, b) : A \times B$
- $\times$-elim: given $p : A \times B$ there are terms $\mathsf{pr}_1 p : A$ and $\mathsf{pr}_2 p : B$

plus computation rules that relate pairings and projections.

Function types are governed by the rules

- $\to$-form: given types A and B there is a type $A \to B$
- $\to$-intro: given a dependent term $x : A \vdash b_x : B$, there is a term $\lambda x.b_x : A \to B$
- $\to$-elim: given terms $f : A \to B$ and $a : A$ there is a term $f(a) : B$

plus computation rules that relate λ-abstractions and evaluations.

Mathematics in dependent type theory

$\times$-form: A, $B \rightsquigarrow A \times B$	$\to$-form: A and $B \rightsquigarrow A \to B$
$\times$-intro: $a : A$, $b : B \rightsquigarrow (a, b) : A \times B$	$\to$-intro: $x : A \vdash b_x : B \rightsquigarrow \lambda x.b_x : A \to B$
$\times$-elim: $p : A \times B \rightsquigarrow \mathsf{pr}_1 p : A$, $\mathsf{pr}_2 p : B$	$\to$-elim: $f : A \to B$, $a : A \rightsquigarrow f(a) : B$

To prove a mathematical proposition in dependent type theory, one constructs a term in the type that encodes its statement.

Proposition. For any types A and B, $(A \times (A \to B)) \to B$.

Proof: By $\to$-intro, it suffices to assume given a term $p : (A \times (A \to B))$ and define a term of type B. By $\times$-elim, p provides terms $\mathsf{pr}_1 p : A$ and $\mathsf{pr}_2 p : A \to B$. By $\to$-elim, these combine to give a term $\mathsf{pr}_2 p(\mathsf{pr}_1 p) : B$. Thus we have

$$\lambda p.\mathsf{pr}_2 p(\mathsf{pr}_1 p) : (A \times (A \to B)) \to B. \quad \square$$

This gives a constructive proof of a logical tautology, modus ponens, while simultaneously constructing an explicit function, the evaluation function.

The homotopy type theoretic Rosetta stone

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

Identity types and path induction

Martin-Löf's rules for identity types are govenrned by the rules:

$=$-form: given a type A and terms $x, y : A$, there is a type $x =_A y$

$=$-intro: given a type A and term $x : A$ there is a term $\mathsf{refl}_x : x =_A x$

The elimination rule for the identity type defines an induction principle analogous to recursion over the natural numbers: it provides sufficient conditions for which to define a dependent function out of the identity type family.

Since $p : x =_A y$ are interpreted paths, we re-brand $=$-elim as:

Path induction: For any type family $P(x, y, p)$ over $x, y : A, p : x =_A y$, to prove $P(x, y, p)$ for all x, y, p it suffices to assume y is x and p is refl_x. That is

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

A computation rule establishes that the proof of $P(x, x, \mathsf{refl}_x)$ is the given one.

The ∞-groupoid of paths

Identity types can be iterated: given $x, y : A$ and $p, q : x =_A y$ there is a type $p =_{x=_A y} q$.

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 $p * q : x = z$

and furthermore concatenation is associative and unital, the associators are coherent ...

Contractible types

The homotopical perspective on type theory suggests new definitions:

A type A is contractible if it comes with a term of type

$$\text{is-contr}(A) := \sum_{a:A} \prod_{x:A} a =_A x$$

By $^{\Sigma}$-elim a proof of contractibility provides:

- a term $c : A$ called the center of contraction and
- a dependent function $h : \prod_{x:A} c =_A x$ called the contracting homotopy, which can be thought of as a continuous choice of paths $h(x) : c =_A x$ for each $x : A$.

Voevodsky's hierarchy of types

Contractible types, those types A for which the type $\text{is-contr}(A) := \sum_{a:A} \prod_{x:A} a =_A x$ has a term, form the bottom level of Voevodsky's hierarchy of types.

A type A is

- a proposition if $\text{is-prop}(A) := \prod_{x,y:A} \text{is-contr}(x =_A y)$
- a set or 0-type if $\text{is-set}(A) := \prod_{x,y:A} \text{is-prop}(x =_A y)$
- a $n+1$-type for $n : \mathbb{N}$ if $\text{is-}n+1\text{-type}(A) := \prod_{x,y:A} \text{is-}n\text{-type}(x =_A y)$

- The unit type $*$ is a contractible type or -2-type.
- The empty type $\varnothing$ is a proposition or -1-type.
- The natural numbers type $\mathbb{N}$ is a set or 0-type.
- The circle S^1 is a 1-type.
- The complex projective space $\mathbb{CP}^\infty$ is a 2-type.
- Higher spheres S^2, S^3 are not truncated, i.e., are not n-types for any n.

Equivalences

Similarly, homotopy theory suggests definitions of when two types A and B are equivalent or when a function $f : A \to B$ is an equivalence:

An equivalence between types A and B is a term of type:

$$A \simeq B :\equiv \sum_{f:A\to B} \left(\sum_{g:B\to A} \prod_{a:A} g(f(a)) =_A a \right) \times \left(\sum_{h:B\to A} \prod_{b:B} f(h(b)) =_B b \right)$$

A term of type $A \simeq B$ provides:

- functions $f : A \to B$ and $g, h : B \to A$ and
- homotopies α and β relating $g \circ f$ and $f \circ h$ to the identity functions.

Using this data, one can define a homotopy from g to h.

So why not say $f : A \to B$ is an equivalence just when:

$$\sum_{g:B\to A} \left(\prod_{a:A} g(f(a)) =_A a \right) \times \left(\prod_{b:B} f(g(b)) =_B b \right) ?$$

This type is not a proposition and may have non-trivial higher structure.

The univalence axiom

Another notion of sameness between types is provided by the universe $\mathcal{U}$ of types, which has (small) types A, B as its terms $\rightsquigarrow$ $A, B : \mathcal{U}$.

Q: How do the types $A =_{\mathcal{U}} B$ and $A \simeq B$ compare?

By path induction, there is a canonical function

$$\text{id-to-equiv} : (A =_{\mathcal{U}} B) \to (A \simeq B)$$

defined by sending refl_A to the identity equivalence id_A.

Univalence Axiom: The function id-to-equiv $: (A =_{\mathcal{U}} B) \to (A \simeq B)$ is an equivalence.

"Identity is equivalent to equivalence."

$$(A =_{\mathcal{U}} B) \simeq (A \simeq B)$$

4

∞-category theory for undergraduates

∞-category theory for undergraduates

∞-category theory would be easier to explain if the foundations of mathematics — set theory and logic — weren't so far away.

Thesis statement: If the foundations of mathematics were something more like homotopy type theory, we could teach ∞-category theory to undergraduates.

Higher Structures 1(1):116–193, 2017.

HIGHER STRUCTURES

A type theory for synthetic ∞-categories

Emily Riehl[a] **and Michael Shulman**[b]

[a] *Dept. of Mathematics, Johns Hopkins U., 3400 N Charles St., Baltimore, MD 21218*
[b] *Dept. of Mathematics, University of San Diego, 5998 Alcalá Park, San Diego, CA 92110*

Abstract

We propose foundations for a synthetic theory of $(\infty, 1)$-categories within homotopy type theory.

Everything that follows is made rigorous in a formal framework, called simplicial type theory, which extends homotopy type theory with simplicial shapes and extension types.

Undergraduates do not need to understand the full details of this formal framework — that can be left to the experts. They just need to learn how to construct definitions and proofs.

Simplicial homotopy type theory

In simplicial type theory, types may depend on other types and also on shapes, which are polytopes $\Phi := \{\vec{t} : 2^n \mid \phi(\vec{t})\}$ cut out of a directed cube by a formula $\phi(\vec{t})$ called a tope.

- Shapes and their defining topes are described syntactically in a language using the symbols $\top, \bot, \wedge, \vee, \equiv$ and $0, 1, \leq$ satisfying intuitionistic logic and strict interval axioms: e.g., $\Delta^n := \{(t_1, \dots, t_n) : 2^n \mid t_n \leq \cdots \leq t_1\}$.
- The shape defined by $\phi \vee \psi$ is the strict pushout of the shapes defined by ϕ and ψ over $\phi \wedge \psi$: e.g., $\partial\Delta^1 := \{t : 2 \mid (t \equiv 0) \vee (t \equiv 1)\}$ is the coproduct of two points.
- Shape inclusions $\Phi \subset \Psi$ arise from impliciations in intuitionistic logic: e.g., the topes

$$\begin{aligned}
\Delta^2 &:= \{(t_1, t_2) : 2^2 \mid t_2 \leq t_1\} \\
\partial\Delta^2 &:= \{(t_1, t_2) : 2^2 \mid (t_2 \leq t_1) \wedge ((0 \equiv t_2) \vee (t_2 \equiv t_1) \vee (t_1 \equiv 1))\} \\
\Lambda^2_1 &:= \{(t_1, t_2) : 2^2 \mid (t_2 \leq t_1) \wedge ((0 \equiv t_2) \vee (t_1 \equiv 1))\}
\end{aligned}$$

define shape inclusions $\Lambda^2_1 \subset \partial\Delta^2 \subset \Delta^2$.

Extension types

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

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

The simplicial type theory allows us to *prove* equivalences between extension types along composites or products of shape inclusions.

Hom types

In the simplicial type theory, any type A has a family of hom types depending on two terms in $x, y : A$:

$$\mathsf{Hom}_A(x,y) :\equiv \left\langle \begin{array}{ccc} \partial\Delta^1 & \xrightarrow{[x,y]} & A \\ \downarrow & \nearrow & \\ \Delta^1 & & \end{array} \right\rangle \text{ type}$$

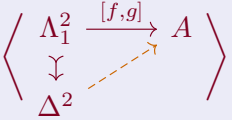
A term $f : \mathsf{Hom}_A(x,y)$ defines an arrow in A from x to y.

We think of the type $\mathsf{Hom}_A(x,y)$ as the mapping space in A from x to y.

A type A also has a family of identity types or path spaces $x = y$ depending on two terms in $x, y : A$, which we will connect to the hom-types momentarily.

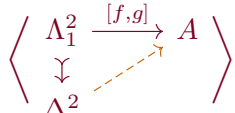
Pre-∞-categories

defn (Riehl–Shulman after Segal). A type A is a pre-∞-category if every composable pair of arrows $f : \mathrm{Hom}_A(x, y)$ and $g : \mathrm{Hom}_A(y, z)$ has a unique composite, i.e.,



is contractible.[a]

[a] A type C is contractible just when $\sum_{c:C} \prod_{x:C} c = x$.

By contractibility,  has a unique inhabitant $\mathrm{comp}_{f,g} : \Delta^2 \to A$.

Write $g \circ f : \mathrm{Hom}_A(x, z)$ for its inner face, *the* composite of f and g.

Identity arrows

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

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

which we denote by id_x and call the identity arrow.

For any $f : \mathsf{Hom}_A(x,y)$ in a pre-∞-category A, the term in the contractible type

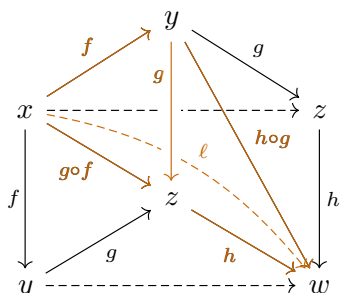
$$\lambda(s,t).f(t) : \left\langle \begin{array}{ccc} \Lambda^2_1 & \xrightarrow{[\mathsf{id}_x, f]} & A \\ \downarrow & \nearrow & \\ \Delta^2 & & \end{array} \right\rangle$$

witnesses the unit axiom $f = f \circ \mathsf{id}_x$. The axiom $f = \mathsf{id}_y \circ f$ is proven similarly.

Associativity of composition

Proposition. In a pre-∞-category A, composition is associative: for any arrows $f : \mathrm{Hom}_A(x, y)$, $g : \mathrm{Hom}_A(y, z)$, and $h : \mathrm{Hom}_A(z, w)$, we have $h \circ (g \circ f) = (h \circ g) \circ f$.

Proof: Consider the composable arrows in the pre-∞-category $\Delta^1 \to A$:



Composing defines a term in the type $\Delta^2 \to (\Delta^1 \to A)$ which defines an arrow $\ell : \mathrm{Hom}_A(x, w)$ so that $\ell = h \circ (g \circ f)$ and $\ell = (h \circ g) \circ f$.

Isomorphisms

An arrow $f : \mathsf{Hom}_A(x, y)$ in a pre-∞-category is an isomorphism if it has a two-sided inverse $g : \mathsf{Hom}_A(y, x)$. However, the type

$$\sum_{g:\mathsf{Hom}_A(y,x)} (g \circ f = \mathsf{id}_x) \times (f \circ g = \mathsf{id}_y)$$

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

$$\text{is-iso}(f) := \left(\sum_{g:\mathsf{Hom}_A(y,x)} g \circ f = \mathsf{id}_x \right) \times \left(\sum_{h:\mathsf{Hom}_A(y,x)} f \circ h = \mathsf{id}_y \right).$$

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

$$x \cong_A y := \sum_{f:\mathsf{Hom}_A(x,y)} \text{is-iso}(f).$$

∞-categories

By path induction, to define a map

$$\text{iso-eq}\colon (x =_A y) \to (x \cong_A y)$$

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

$$\text{is-iso}(\text{refl}_x) := \text{id}_x.$$

Definition (Riehl–Shulman after Rezk). A pre-∞-category A is ∞-category iff every isomorphism is an identity, i.e., iff the map

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

is an equivalence.

Coproducts and adjunctions in ∞-categories

Two objects $a, a' : A$ in an ∞-category admit a coproduct if there is another object $a \sqcup a' : A$ and arrows $\iota_a : \mathsf{Hom}_A(a, a \sqcup a')$ and $\iota_{a'} : \mathsf{Hom}_A(a', a \sqcup a')$ so that the function defined by composing with ι_a and $\iota_{a'}$ defines an equivalence

$$\lambda x.\lambda h.(h \circ \iota_a, h \circ \iota_{a'}) : \prod_{x:A} \mathsf{Hom}_A(a \sqcup a', x) \simeq \mathsf{Hom}_A(a, x) \times \mathsf{Hom}_A(a', x)$$

A pair of functions $f\colon A \to B$ and $u\colon B \to A$ between ∞-categories define an adjunction if there is a family of equivalences

$$\prod_{a:A} \prod_{b:B} \mathsf{Hom}_B(fa, b) \simeq \mathsf{Hom}_A(a, ub).$$

Note both definitions are arguably simplier than for ordinary categories: the "naturality" condition is dropped because here it is automatically true.

An ∞-categorical proof of a result in ∞-category theory

Theorem. Left adjoints $f\colon A \to B$ between between ∞-categories preserve coproducts: for $a, a' : A$ admitting a coproduct $a \sqcup a'$ in A, $f(a \sqcup a') \cong fa \sqcup fa'$ in B.

Proof: By the ∞-categorical Yoneda lemma, to show $f(a \sqcup a') \cong fa \sqcup fa'$ it suffices to define a family of equivalences $\prod_{x:B} \mathsf{Hom}_B(f(a \sqcup a'), x) \simeq \mathsf{Hom}_B(fa \sqcup fa', x)$. The equivalences

$$\prod_{x:A} \mathsf{Hom}_A(a \sqcup a', x) \simeq \mathsf{Hom}_A(a, x) \times \mathsf{Hom}_A(a', x) \quad \text{and}$$
$$\prod_{a:A} \prod_{b:B} \mathsf{Hom}_B(fa, b) \simeq \mathsf{Hom}_A(a, ub)$$

defined by the coproduct and the adjunction compose to define the required equivalence

$$\prod_{x:B} \mathsf{Hom}_B(f(a \sqcup a'), x) \simeq \mathsf{Hom}_A(a \sqcup a', ux) \simeq \mathsf{Hom}_A(a, ux) \times \mathsf{Hom}_A(a', ux)$$
$$\simeq \mathsf{Hom}_B(fa, x) \times \mathsf{Hom}_B(fa', x) \simeq \mathsf{Hom}_B(fa \sqcup fa', x). \quad \square$$

5

∞-category theory for computers

Could ∞-category theory be taught to undergraduates?

As far as we know, there are no existing formalizations of ∞-category theory in any proof assistant library such as Lean-mathlib, Agda-UniMath, Coq-HoTT,...

Why not?

Could ∞-Category Theory Be Taught to Undergraduates?

Emily Riehl

1. The Algebra of Paths

It is natural to probe a suitably nice topological space X by means of its paths, the continuous functions from the standard unit interval $I = [0, 1] \subset \mathbb{R}$ to X. But what structure do the paths in X form?

To start, the paths form the edges of a directed graph whose vertices are the points of X: a path $p\colon I \to X$ defines an arrow from the point $p(0)$ to the point $p(1)$. Moreover, this graph is reflexive, with the constant path refl_x at each point $x \in X$ defining a distinguished endoarrow.

Can this reflexive directed graph be given the structure of a category? To do so, it is natural to define the composite of a path p from x to y and a path q from y to z by gluing together these continuous maps—i.e., by concatenating the paths—and then by reparametrizing via the homeomorphism $I \cong I \cup_{1\sim 0} I$ that traverses each path at double speed:

$$I \xrightarrow{\cong} I \cup_{1\sim 0} I \xrightarrow{p*q} X \qquad (1.1)$$

But the composition operation $*$ fails to be associative or unital. In general, given a path r from z to w, the

Emily Riehl is a professor of mathematics at Johns Hopkins University. Her email address is eriehl@jhu.edu.

Communicated by Notices Associate Editor Steven Sam.

For permission to reprint this article, please contact: reprint-permission@ams.org.

DOI: https://doi.org/10.1090/noti2692

The traditional foundations of mathematics are not really suitable for "higher mathematics" such as ∞-category theory, where the basic objects are built out of higher-dimensional types instead of mere sets. However, there are proposals for new foundations for mathematics that are closer to mathematician's core intuitions, based on Martin-Löf's dependent type theory such as

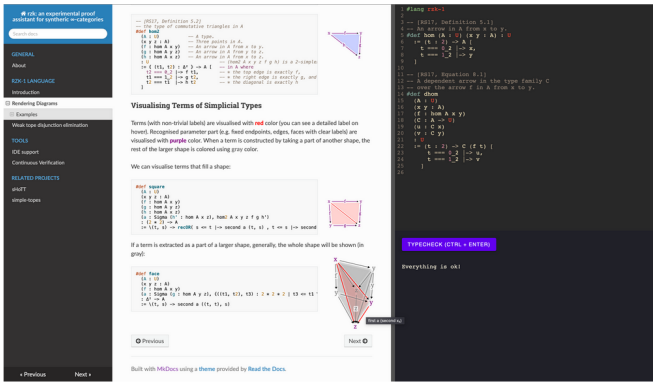
- homotopy type theory,
- higher observational type theory, and the
- simplicial type theory, that we use here.

An experimental proof assistant Rzk for ∞-category theory

rzk

MkDocs documentation | Haddock documentation | Build with GHCJS and Deploy to GitHub Pages passing

An experimental proof assistant for synthetic ∞-categories.



The proof assistant Rzk was written by Nikolai Kudasov:

About this project

This project has started with the idea of bringing Riehl and Shulman's 2017 paper [1] to "life" by implementing a proof assistant based on their type theory with shapes. Currently an early prototype with an online playground is available. The current implementation is capable of checking various formalisations. Perhaps, the largest formalisations are available in two related projects: https://github.com/fizruk/sHoTT and https://github.com/emilyriehl/yoneda. `sHoTT` project (originally a fork of the yoneda project) aims to cover more formalisations in simplicial HoTT and ∞-categories, while `yoneda` project aims to compare different formalisations of the Yoneda lemma.

Internally, `rzk` uses a version of second-order abstract syntax allowing relatively straightforward handling of binders (such as lambda abstraction). In the future, `rzk` aims to support dependent type inference relying on E-unification for second-order abstract syntax [2]. Using such representation is motivated by automatic handling of binders and easily automated boilerplate code. The idea is that this should keep the implementation of `rzk` relatively small and less error-prone than some of the existing approaches to implementation of dependent type checkers.

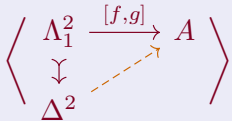
An important part of `rzk` is a tope layer solver, which is essentially a theorem prover for a part of the type theory. A related project, dedicated just to that part is available at https://github.com/fizruk/simple-topes. `simple-topes` supports used-defined cubes, topes, and tope layer axioms. Once stable, `simple-topes` will be merged into `rzk`, expanding the proof assistant to the type theory with shapes, allowing formalisations for (variants of) cubical, globular, and other geometric versions of HoTT.

rzk-lang.github.io/rzk

All of the results mentioned above and much more besides have been formally verified!

Pre-∞-categories, formally

defn (Riehl–Shulman after Segal). A type A is a pre-∞-category if every pair of arrows $f : \mathrm{Hom}_A(x, y)$ and $g : \mathrm{Hom}_A(y, z)$ has a unique composite, i.e.,



is contractible.

A type is **a pre-∞-category** if every composable pair of arrows has a unique composite, meaning that the type of composites is contractible.

Note this is a considerable simplification of the usual Segal condition, which also requires homotopical uniqueness of higher-order composites. Here this higher-order uniqueness is a consequence of the uniqueness of binary composition.

RS17, Definition 5.3

```
#def Is-pre-∞-category
  ( A : U)
  : U
  :=
    ( x : A) → (y : A) → (z : A)
  → ( f : Hom A x y) → (g : Hom A y z)
  → is-contr (Σ (h : Hom A x z) , (Hom2 A x y z f g h))
```

Composition in a pre-∞-category, formally

By contractibility, $\left\langle \begin{array}{ccc} \Lambda^2_1 & \xrightarrow{[f,g]} & A \\ \downarrow & \nearrow & \\ \Delta^2 & & \end{array} \right\rangle$ has a unique inhabitant $\mathrm{comp}_{f,g} : \Delta^2 \to A$.

Write $g \circ f : \mathrm{Hom}_A(x, z)$ for its inner face, *the* composite of f and g.

Pre-∞-categories have a composition functor and witnesses to the composition relation. Composition is written in diagrammatic order to match the order of arguments in `is-pre-∞-category`.

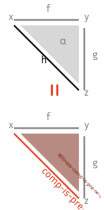
```
#def Comp-is-pre-∞-category
  ( A : U)
  ( is-pre-∞-category-A : Is-pre-∞-category A)
  ( x y z : A)
  ( f : Hom A x y)
  ( g : Hom A y z)
  : Hom A x z
  := first (first (is-pre-∞-category-A x y z f g))

#def Witness-comp-is-pre-∞-category
  ( A : U)
  ( is-pre-∞-category-A : Is-pre-∞-category A)
  ( x y z : A)
  ( f : Hom A x y)
  ( g : Hom A y z)
  : Hom2 A x y z f g (Comp-is-pre-∞-category A is-pre-∞-category-A x y z f g)
  := second (first (is-pre-∞-category-A x y z f g))
```

Uniqueness of composition in a pre-∞-category, formally

Composition in a pre-∞-category is unique in the following sense. If there is a witness that an arrow `h` is a composite of `f` and `g`, then the specified composite equals `h`.

```
#def Uniqueness-comp-is-pre-∞-category
  ( A : U)
  ( is-pre-∞-category-A : Is-pre-∞-category A)
  ( x y z : A)
  ( f : Hom A x y)
  ( g : Hom A y z)
  ( h : Hom A x z)
  ( alpha : Hom2 A x y z f g h)
  : ( Comp-is-pre-∞-category A is-pre-∞-category-A x y z f g) = h
  :=
    first-path-Σ
      ( Hom A x z)
      ( Hom2 A x y z f g)
      ( Comp-is-pre-∞-category A is-pre-∞-category-A x y z f g
      , Witness-comp-is-pre-∞-category A is-pre-∞-category-A x y z f g)
      ( h , alpha)
      ( homotopy-contraction
        ( Σ ( k : Hom A x z) , (Hom2 A x y z f g k))
        ( is-pre-∞-category-A x y z f g)
        ( h , alpha))
```



See emilyriehl.github.io/yoneda and rzk-lang.github.io/sHoTT for more.

Conclusions and future work

Observations:

- ∞-category theory is significantly easier to formalize in a foundation system based on homotopy type theory.
- By moving much of the complexity of "higher structures" into the background foundation system, the gap between ∞-category theory and 1-category narrows substantially.
- A computer proof assistant is a fantastic tool for learning to write proofs in new foundations — indeed, through formalization in RZK we caught an error of circular reasoning in the Riehl–Shulman paper!

Future work:

- We would love help formalizing more results from ∞-category theory in RZK.
- But the initial version of the simplicial type theory is not sufficiently powerful to prove all results about ∞-categories, so further extensions of this synthetic framework are needed.

References

- Emily Riehl, Could ∞-category theory be taught to undergraduates?, Notices of the AMS 70(5):727–736, May 2023; arXiv:2302.07855
- Nikolai Kudasov, Emily Riehl, Jonathan Weinberger, Formalizing the ∞-categorical Yoneda lemma, CPP 2024: 274–290; arXiv:2309.08340

- Emily Riehl and Michael Shulman, A type theory for synthetic ∞-categories, Higher Structures 1(1):116–193, 2017; arXiv:1705.07442
- César Bardomiano Martínez, Limits and colimits of synthetic ∞-categories, arXiv:2202.12386
- Ulrik Buchholtz, Jonathan Weinberger, Synthetic fibered $(\infty, 1)$-category theory, Higher Structures 7(1): 74–165, 2023; arXiv:2105.01724
- Daniel Gratzer, Jonathan Weinberger, Ulrik Buchholtz, Directed univalence in simplicial homotopy type theory; arXiv:2407.09146
- Daniel Gratzer, Jonathan Weinberger, Ulrik Buchholtz, The Yoneda embedding in simplicial type theory; arXiv:2501.13229