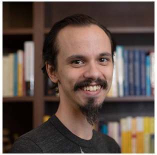
Emily Riehl

Johns Hopkins University

Prospects for Computer Formalization of Infinite-Dimensional Category Theory

joint with Mario Carneiro, Nikolai Kudasov, Dominic Verity, and Jonathan Weinberger




CPP 2025

In Honor of Martin Luther King Jr. Day

I accept this award today with an abiding faith in America and an audacious faith in the future of mankind. I refuse to accept despair as the final response to the ambiguities of history. I refuse to accept the idea that the "isness" of man's present nature makes him morally incapable of reaching up for the eternal "oughtness" that forever confronts him. I refuse to accept the idea that man is mere flotsam and jetsam in the river of life, unable to influence the unfolding events which surround him. I refuse to accept the view that mankind is so tragically bound to the starless midnight of racism and war that the bright daybreak of peace and brotherhood can never become a reality. … I have the audacity to believe that peoples everywhere can have three meals a day for their bodies, education and culture for their minds, and dignity, equality and freedom for their spirits. I believe that what self-centered men have torn down men other-centered can build up. … I still believe that we shall overcome! … This faith can give us courage to face the uncertainties of the future. It will give our tired feet new strength as we continue our forward stride toward the city of freedom. When our days become dreary with low-hovering clouds and our nights become darker than a thousand midnights, we will know that we are living in the creative turmoil of a genuine civilization struggling to be born.

— 10 December 1964, Oslo

Recent achievements in computer formalized mathematics

> *Formalized mathematics, in tandem with other forms of computerized mathematics[1], provides better management of mathematical knowledge, an opportunity to carry out ever more complex and larger projects, and hitherto unseen levels of precision.*
>
> — *Andrej Bauer, "The dawn of formalized mathematics," delivered at the 8th European Congress of Mathematics*

Recent successes include:

- the Feit-Thompson Odd Order Theorem, a foundational result in the classification of finite simple groups, 2006–2012, Coq
- the Kepler conjecture, resolving a 1611 conjecture, 2003–2014, HOL Light
- the liquid tensor experiment, formalizing condensed mathematics, 2020–2022, Lean
- the Brunerie number, computing $\pi_4 S^3 \cong \mathbb{Z}/2\mathbb{Z}$, 2015–2022, Cubical agda

[1] Jacques Carette, William M. Farmer, Michael Kohlhase, and Florian Rabe. Big math and the one-brain barrier — the tetrapod model of mathematical knowledge. Mathematical Intelligencer, 43(1):78–87, 2021.

Plan

1. Prospects for formalizing the ∞-categories literature

2. Formalizing axiomatic ∞-category theory via ∞-cosmoi in `Lean`

3. Formalizing synthetic ∞-category theory in simplicial HoTT in `Rzk`

1

Prospects for formalizing the ∞-categories literature

Formalizing post-rigorous mathematics?

From Terry Tao's blog post "There's more to mathematics than rigour and proofs":

One can roughly divide mathematical education into three stages:

1. *The "pre-rigorous" stage, in which mathematics is taught in an informal, intuitive manner, based on examples, fuzzy notions, and hand-waving. ...The emphasis is more on computation than on theory. This stage generally lasts until the early undergraduate years.*
2. *The "rigorous" stage, in which one is now taught that in order to do maths "properly", one needs to work and think in a much more precise and formal manner ...The emphasis is now primarily on theory; and one is expected to be able to comfortably manipulate abstract mathematical objects without focusing too much on what such objects actually "mean". This stage usually occupies the later undergraduate and early graduate years.*
3. *The "post-rigorous"stage, in which one has grown comfortable with all the rigorous foundations of one's chosen field, and is now ready to revisit and refine one's pre-rigorous intuition on the subject, but this time with the intuition solidly buttressed by rigorous theory. ... The emphasis is now on applications, intuition, and the "big picture". This stage usually occupies the late graduate years and beyond.*

... The ideal state to reach is when every heuristic argument naturally suggests its rigorous counterpart, and vice versa.

Case studies from the literature

The literature developing the theory of ∞-categories is arguably "post-rigorous":

- Arguments are not always explained in full detail.
- Some claims made as part of the argument may not quite be true as stated.
- Nevertheless, proofs with gaps or errors are often "morally correct."

For instance, proofs in the literature may rely on

- incomplete definitions,
- sketched arguments, or
- explicit unproven conjectures.

Avoiding a precise definition of ∞-categories

The precursor to Jacob Lurie's *Higher Topos Theory* is a 2003 preprint On ∞-Topoi, which avoids using a precise definition of ∞-categories[2]:

> *We will begin in §1 with an informal review of the theory of ∞-categories. There are many approaches to the foundation of this subject, each having its own particular merits and demerits. Rather than single out one of those foundations here, we shall attempt to explain the ideas involved and how to work with them. The hope is that this will render this paper readable to a wider audience, while experts will be able to fill in the details missing from our exposition in whatever framework they happen to prefer.*

Perlocutions of this form are quite common in the field — however the book *Higher Topos Theory* does not proceed in this manner, instead proving theorems for a concrete model of ∞-categories.

---

[2] Very roughly, an ∞-category is a weak infinite-dimensional category. In the parlance of the field, selecting a set-theoretic definition of ∞-categories is referred to as "choosing a model."

A proof(?) of the cobordism hypothesis

The cobordism hypothesis classifies (fully-extended) topological quantum field theories, which are functors indexed by a suitably-defined higher category of cobordisms between framed n-manifolds with corners. In a celebrated expository article on the subject, Dan Freed writes:

> *The cobordism hypothesis was conjectured by Baez-Dolan in the mid 1990s. It has now been proved by Hopkins-Lurie in dimension two and by Lurie in higher dimensions. There are many complicated foundational issues which lie behind the definitions and the proof, and only a detailed sketch has appeared so far.*[1]

The footnote elaborates:

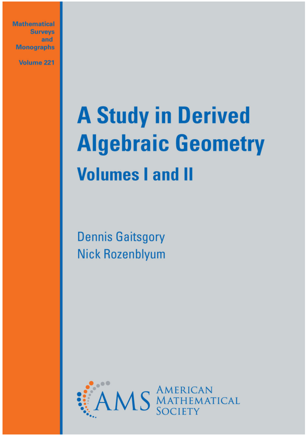
> [1] *Nonetheless, we use "theorem" and its synonyms in this manuscript. The foundations are rapidly being filled in and alternative proofs have also been carried out, though none has yet appeared in print.*

There seems to be no clear consensus on this point of view: a mathOVERFLOW question "What is the status of the cobordism hypothesis?" asked in 2023 remains open.

A conjectural(?) study in derived algebraic geometry

A two-volume study in derived algebraic geometry runs to nearly 1000 pages. Much of the first volume is devoted to developing necessary preliminary results in $(\infty, 1)$-category theory and $(\infty, 2)$-category theory, and includes the following disclaimer:

> *Unfortunately, the existing literature on* $(\infty, 2)$*-categories does not contain the proofs of all the statements that we need. We decided to leave some of the statements unproved, and supply the corresponding proofs elsewhere (including the proofs here would have altered the order of the exposition, and would have come at the expense of clarity).*



This is followed by a list of seven unproved statements.

A contradiction with no obvious error

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 memoire "A la poursuite des champs".

It is well-known [GZ] that CW-complexes X such that $\pi_1(X,x) = 0$ 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."

Obstructions to formalization?

How might this literature read differently in a future where mathematicians are expected to work interactively with a computer proof assistant?

- If it is undesirable to give a precise construction of a mathematical notion (e.g., of the category of ∞-categories), one could instead axiomatize the necessary properties (and hope that the theory is not vacuous).
- Sketch proofs will be harder to implement, as a proof assistant will require clearer definitions and scaffolding. But a formalized sketch, will make it much clearer what gaps remain in the proof.
- A proof modulo unproven conjectures should be formalizable, provided those conjectures and clearly stated in exactly the way they are used.
- An incorrect proof should not be formalizable — which is of course a good thing. And perhaps the process of formalization would help identify the error by calling attention to a subtle obstacle to be overcome.

The fundamental problem: how do we formalize proofs, in an area like ∞-category theory, where arguments tend to be long and involve complexity at nearly every step?

Prospects for formalization?

I can imagine three strategies for formalizing the theory of ∞-categories.

Strategy I. Give precise "analytic" definitions of ∞-categorical notions in some model (e.g., using quasi-categories). Prove theorems using the combinatorics of that model.

Strategy II. Axiomatize the category of ∞-categories (e.g., using the notion of ∞-cosmos or something similar). State and prove theorems about ∞-categories in this axiomatic language. To show that this theory is non-vacuous, prove that some model satisfies the axioms and formalize other examples, as desired.

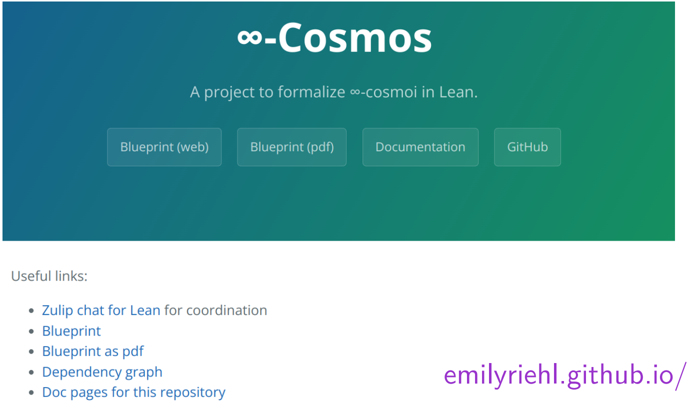
Strategy III. Avoid the technicalities of set-based models by developing the theory of ∞-categories "synthetically," in a domain-specific type theory. Formalization then requires a bespoke proof assistant (e.g., `Rzk`).

2

Formalizing axiomatic ∞-category theory via ∞-cosmoi in `Lean`

An axiomatic theory of ∞-categories in Lean

The ∞-cosmos project — co-led Mario Carneiro, Dominic Verity, and myself — aims to formalize a particular axiomatic theory approach to ∞-category theory `Lean`'s mathematics library `Mathlib`. Pietro Monticone and others helped us set up a blueprint, website, github repository, and Zulip channel to organize the workflow.



emilyriehl.github.io/infinity-cosmos

The idea of an ∞-category

Lean defines an ordinary 1-category as follows:

```
class Quiver (V : Type u) where
  /-- The type of edges/arrows/morphisms between a given source and target. -/
  Hom : V → V → Sort v
```

```
class CategoryStruct (obj : Type u) extends Quiver.{v + 1} obj : Type max u (v + 1) where
  /-- The identity morphism on an object. -/
  id : ∀ X : obj, Hom X X
  /-- Composition of morphisms in a category, written `f ≫ g`. -/
  comp : ∀ {X Y Z : obj}, (X ⟶ Y) → (Y ⟶ Z) → (X ⟶ Z)
```

```
class Category (obj : Type u) extends CategoryStruct.{v} obj : Type max u (v + 1) where
  /-- Identity morphisms are left identities for composition. -/
  id_comp : ∀ {X Y : obj} (f : X ⟶ Y), 𝟙 X ≫ f = f := by aesop_cat
  /-- Identity morphisms are right identities for composition. -/
  comp_id : ∀ {X Y : obj} (f : X ⟶ Y), f ≫ 𝟙 Y = f := by aesop_cat
  /-- Composition in a category is associative. -/
  assoc : ∀ {W X Y Z : obj} (f : W ⟶ X) (g : X ⟶ Y) (h : Y ⟶ Z), (f ≫ g) ≫ h = f ≫ g ≫ h := by
    aesop_cat
```

The idea of an ∞-category is just to

- replace all the types by ∞-groupoids aka homotopy types aka anima, i.e., the information of a topological space encoded by its homotopy groups
- and suitably weaken all the structures and axioms.

“Analytic” ∞-categories in Lean

An elegant “coordinatization” of these ideas encodes an ∞-category as a quasi-category, which Johan Commelin contributed to `Mathlib`:

```
/-- A simplicial set `S` is a *quasicategory* if it satisfies the following horn-filling condition:
for every `n : ℕ` and `0 < i < n`,
every map of simplicial sets `σ₀ : Λ[n, i] → S` can be extended to a map `σ : Δ[n] → S`.

[Kerodon, 003A] -/
class Quasicategory (S : SSet) : Prop where
  hornFilling' : ∀ ⦃n : ℕ⦄ ⦃i : Fin (n+3)⦄ (σ₀ : Λ[n+2, i] ⟶ S)
    (_h0 : 0 < i) (_hn : i < Fin.last (n+2)),
      ∃ σ : Δ[n+2] ⟶ S, σ₀ = hornInclusion (n+2) i ≫ σ
```

where ∞-groupoids can be similarly “coordinatized” as Kan complexes:

```
/-- A simplicial set `S` is a *Kan complex* if it satisfies the following horn-filling condition:
for every nonzero `n : ℕ` and `0 ≤ i ≤ n`,
every map of simplicial sets `σ₀ : Λ[n, i] → S` can be extended to a map `σ : Δ[n] → S`. -/
class KanComplex (S : SSet.{u}) : Prop where
  hornFilling : ∀ ⦃n : ℕ⦄ ⦃i : Fin (n + 2)⦄ (σ₀ : Λ[n + 1, i] ⟶ S),
    ∃ σ : Δ[n + 1] ⟶ S, σ₀ = hornInclusion (n + 1) i ≫ σ
```

But very few results have been formalized with these technical definitions. Indeed, only last week, Joël Riou discovered that the definition of Kan complexes was wrong!

The idea of the ∞-cosmos project

The aim of the ∞-cosmos project is to leverage the existing 1-category theory, 2-category theory, and enriched category theory libraries in `Lean` to formalize basic ∞-category theory.

This is achieved by developing the theory of ∞-categories more abstractly, using the axiomatic notion of an ∞-cosmos, which is an enriched category whose objects are ∞-categories.

From this we can extract a 2-category whose objects are ∞-categories, whose morphisms are ∞-functors, and whose 2-cells are ∞-natural transformations. The formal theory of ∞-categories (adjunctions, co/limits, Kan extensions) can be defined using this 2-category and some of these notions are in the `Mathlib` already!

Proving that quasi-categories define an ∞-cosmos will be hard, but this tedious verifying of homotopy coherences will only need to be done once rather than in every proof.

Progress: a formalized definition of an ∞-cosmos

The ∞-cosmos project was launched in September 2024. After adding some background material on enriched category theory, we have formalized the main definition and made numerous supporting contributions to `Mathlib`.

```
variable (K : Type u) [Category.{v} K] [SimplicialCategory K]

/-- A `PreInfinityCosmos` is a simplicially enriched category whose hom-spaces are quasi-categories
and whose morphisms come equipped with a special class of isofibrations.-/
class PreInfinityCosmos extends SimplicialCategory K where
  [has_qcat_homs : ∀ {X Y : K}, SSet.Quasicategory (EnrichedCategory.Hom X Y)]
  IsIsofibration : MorphismProperty K
```

```
variable (K : Type u) [Category.{v} K][PreInfinityCosmos.{v} K]

/-- An `InfinityCosmos` extends a `PreInfinityCosmos` with limit and isofibration axioms..-/
class InfinityCosmos extends PreInfinityCosmos K where
  comp_isIsofibration {A B C : K} (f : A ↠ B) (g : B ↠ C) : IsIsofibration (f.1 ≫ g.1)
  iso_isIsofibration {X Y : K} (e : X ⟶ Y) [IsIso e] : IsIsofibration e
  all_objects_fibrant {X Y : K} (hY : IsConicalTerminal Y) (f : X ⟶ Y) : IsIsofibration f
  [has_products : HasConicalProducts K]
  prod_map_fibrant {γ : Type w} {A B : γ → K} (f : ∀ i, A i ↠ B i) :
    IsIsofibration (Limits.Pi.map (λ i ↦ (f i).1))
  [has_isoFibration_pullbacks {E B A : K} (p : E ↠ B) (f : A ⟶ B)  : HasConicalPullback p.1 f]
  pullback_is_isoFibration {E B A P : K} (p : E ↠ B) (f : A ⟶ B)
    (fst : P ⟶ E) (snd : P ⟶ A) (h : IsPullback fst snd p.1 f) : IsIsofibration snd
  [has_limits_of_towers (F : ℕᵒᵖ ⥤ K) :
    (∀ n : ℕ, IsIsofibration (F.map (homOfLE (Nat.le_succ n)).op)) → HasConicalLimit F]
  has_limits_of_towers_isIsofibration (F : ℕᵒᵖ ⥤ K) (hf) :
    haveI := has_limits_of_towers F hf
    IsIsofibration (limit.π F (.op 0))
  [has_cotensors : HasCotensors K]
  leibniz_cotensor  {U V : SSet} (i : U ⟶ V) [Mono i] {A B : K} (f : A ↠ B) {P : K}
    (fst : P ⟶ U ⋔ A) (snd : P ⟶ V ⋔ B)
    (h : IsPullback fst snd (cotensorCovMap U f.1) (cotensorContraMap i B)) :
    IsIsofibration (h.isLimit.lift <|
      PullbackCone.mk (cotensorContraMap i A) (cotensorCovMap V f.1)
        (cotensor_bifunctoriality i f.1)) --TODO : Prove that these pullbacks exist.
  local_isoFibration {X A B : K} (f : A ↠ B) : Isofibration (toFunMap X f.1)
```

Challenge: Lean's difficulty with the 1-category of categories

In formalizing the free category and underlying reflexive quiver adjunction:

```
left_triangle := by
  ext V
  apply Cat.FreeRefl.lift_unique'
  simp only [id_obj, Cat.free_obj, comp_obj, Cat.freeRefl_obj_α, NatTrans.comp_app,
    forget_obj, whiskerRight_app, associator_hom_app, whiskerLeft_app, id_comp,
    NatTrans.id_app']
  rw [Cat.id_eq_id, Cat.comp_eq_comp]
  simp only [Cat.freeRefl_obj_α, Functor.comp_id]
  rw [← Functor.assoc, ← Cat.freeRefl_naturality, Functor.assoc]
  dsimp [Cat.freeRefl]
  rw [adj.counit.component_eq' (Cat.FreeRefl V)]
  conv =>
    enter [1, 1, 2]
    apply (Quiv.comp_eq_comp (X := Quiv.of _) (Y := Quiv.of _) (Z := Quiv.of _) ..).symm
  rw [Cat.free.map_comp]
  show (_ ⋙ ((Quiv.forget ⋙ Cat.free).map (X := Cat.of _) (Y := Cat.of _)
    (Cat.FreeRefl.quotientFunctor V))) ⋙ _ = _
  rw [Functor.assoc, ← Cat.comp_eq_comp]
  conv => enter [1, 2]; apply Quiv.adj.counit.naturality
  rw [Cat.comp_eq_comp, ← Functor.assoc, ← Cat.comp_eq_comp]
  conv => enter [1, 1]; apply Quiv.adj.left_triangle_components V.toQuiv
  exact Functor.id_comp _
```

Lean was confused by

- what category we're in when objects are type classes
- competing notations for functors
- whiskered commutative diagrams

Challenge: dependent equalities between the 2-cells in a 2-category

On paper, 2-cells in a 2-category compose by pasting:

$$
\begin{tikzcd}
& A \arrow[r, "G_1"] \arrow[d, "L_1"] \arrow[dr, phantom, "\Downarrow\alpha"] & C \arrow[r, equal] \arrow[d, "L_2"] \arrow[r, phantom, shift right=3, "\Downarrow\eta_2"] & C \arrow[r, "G_2"] \arrow[d, "L_2"] \arrow[dr, phantom, "\Downarrow\beta"] & E \arrow[r, equal] \arrow[d, "L_3"] \arrow[r, phantom, shift right=3, "\Downarrow\eta_3"] & E \\
B \arrow[r, equal] \arrow[ur, "R_1"] \arrow[r, phantom, shift left=3, "\Downarrow\epsilon_1"] & B \arrow[r, "H_1"'] & D \arrow[r, equal] \arrow[ur, "R_2"] \arrow[r, phantom, shift left=3, "\Downarrow\epsilon_2"] & D \arrow[r, "H_2"'] & F \arrow[ur, "R_3"'] &
\end{tikzcd}
$$

In `Mathlib`, the 2-cells displayed here belong to dependent types (over their boundary 1-cells and objects) and depending on how the whiskerings are encoded are not obviously composable at all:

e.g., is $R_3H_2L_2\eta_2G_1R_1$ composable with $R_3H_2\epsilon_2L_2G_1R_1$?

Challenge: dependent equalities between the 2-cells in a 2-category

```
/-- The mates equivalence commutes with vertical composition. -/
theorem mateEquiv_vcomp
    (α : G₁ ⋙ L₂ ⟶ L₁ ⋙ H₁) (β : G₂ ⋙ L₃ ⟶ L₂ ⋙ H₂) :
    (mateEquiv (G := G₁ ⋙ G₂) (H := H₁ ⋙ H₂) adj₁ adj₃) (leftAdjointSquare.vcomp α β) =
      rightAdjointSquare.vcomp (mateEquiv adj₁ adj₂ α) (mateEquiv adj₂ adj₃ β) := by
  unfold leftAdjointSquare.vcomp rightAdjointSquare.vcomp mateEquiv
  ext b
  simp only [comp_obj, Equiv.coe_fn_mk, whiskerLeft_comp, whiskerLeft_twice, whiskerRight_comp,
    assoc, comp_app, whiskerLeft_app, whiskerRight_app, id_obj, Functor.comp_map,
    whiskerRight_twice]
  slice_rhs 1 4 => rw [← assoc, ← assoc, ← unit_naturality (adj₃)]
  rw [L₃.map_comp, R₃.map_comp]
  slice_rhs 2 4 =>
    rw [← R₃.map_comp, ← R₃.map_comp, ← assoc, ← L₃.map_comp, ← G₂.map_comp, ← G₂.map_comp]
    rw [← Functor.comp_map G₂ L₃, β.naturality]
  rw [(L₂ ⋙ H₂).map_comp, R₃.map_comp, R₃.map_comp]
  slice_rhs 4 5 =>
    rw [← R₃.map_comp, Functor.comp_map L₂ _, ← Functor.comp_map _ L₂, ← H₂.map_comp]
    rw [adj₂.counit.naturality]
  simp only [comp_obj, Functor.comp_map, map_comp, id_obj, Functor.id_map, assoc]
  slice_rhs 4 5 =>
    rw [← R₃.map_comp, ← H₂.map_comp, ← Functor.comp_map _ L₂, adj₂.counit.naturality]
  simp only [comp_obj, id_obj, Functor.id_map, map_comp, assoc]
  slice_rhs 3 4 =>
    rw [← R₃.map_comp, ← H₂.map_comp, left_triangle_components]
  simp only [map_id, id_comp]
```

In the 2-category Cat, I formalized a proof that the unit η_2 and counit ϵ_2 cancel, but not via a 2-categorical pasting argument. As a result, `Mathlib` does not know this is true in any 2-category.

Contributors to the ∞-cosmos project

So far formalizations (and preliminary mathematical work) have been contributed by:

Dagur Asgeirsson, Alvaro Belmonte, Mario Carneiro, Daniel Carranza, Johan Commelin, Jack McKoen, Pietro Monticone, Matej Penciak, Nima Rasekh, Emily Riehl, Joël Riou, Joseph Tooby-Smith, Adam Topaz, Dominic Verity, Nick Ward, and Zeyi Zhao.

Anyone is welcome to join us!

emilyriehl.github.io/infinity-cosmos

3

Formalizing synthetic ∞-category theory in simplicial HoTT in `Rzk`

Could ∞-category theory be taught to undergraduates?

Recall ∞-categories are like categories where all the sets are replaced by ∞-groupoids:

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

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 $I \coloneqq [0,1] \subset \mathbb{R}$ 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 $p : I \to X$ 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 $I \cong I \cup_{1=0} I$ that traverses each path at double speed:

$$\begin{tikzcd} I \arrow[r, "\cong"] \arrow[rr, bend right, "p*q"'] & I \cup_{1=0} I \arrow[r, "p\cup q"] & X \end{tikzcd} \tag{1.1}$$

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 based on Martin-Löf's dependent type theory where the primative types have "higher structure" such as

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

∞-categories in simplicial homotopy type theory

The identity type family gives each type the structure of an ∞-groupoid: each type A has a family of identity types over $x, y : A$ whose terms $p : x =_A y$ are called paths.

In a "directed" extension of homotopy type theory introduced in

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

each type A also has a family of hom types $\text{Hom}_A(x, y)$ over $x, y : A$ whose terms $f : \text{Hom}_A(x, y)$ are called arrows.

defn (Riehl–Shulman after Joyal and Rezk). A type A is an ∞-category if:

- Every pair of arrows $f : \text{Hom}_A(x, y)$ and $g : \text{Hom}_A(y, z)$ has a unique composite, defining a term $g \circ f : \text{Hom}_A(x, z)$.
- Paths in A are equivalent to isomorphisms in A.

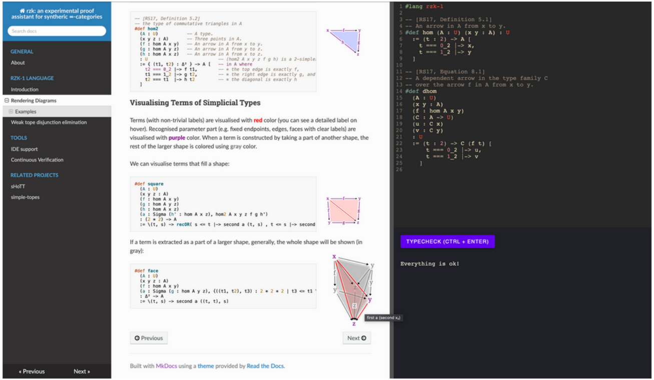
With more of the work being done by the foundation system, perhaps someday ∞-category theory will be easy enough to teach to undergraduates?

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

A formalized proof of the ∞-categorical Yoneda lemma

Nikolai Kudasov, Jonathan Weinberger, and I formalized the ∞-Yoneda lemma:

For any pre-∞-category `A` terms `a b : A`, the contravariant Yoneda lemma provides an equivalence between the type `(z : A) → Hom A z a → Hom A z b` of natural transformations and the type `Hom A a b`.

One of the maps in this equivalence is evaluation at the identity. The inverse map makes use of the contravariant transport operation.

The following map, `contra-evid` evaluates a natural transformation out of a representable functor at the identity arrow.

```
#def Contra-evid
  ( A : U)
  ( a b : A)
  : ( ( z : A) → Hom A z a → Hom A z b) → Hom A a b
  := \ ϕ → ϕ a (Id-hom A a)
```

The inverse map only exists for pre-∞-categories.

```
#def Contra-yon
  ( A : U)
  ( is-pre-∞-category-A : Is-pre-∞-category A)
  ( a b : A)
  : Hom A a b → ((z : A) → Hom A z a → Hom A z b)
  := \ v z f → Comp-is-pre-∞-category A is-pre-∞-category-A z a b f v
```

emilyriehl.github.io/yoneda/

Contributors to the simplicial HoTT library

So far formalizations to the broader project of formalizing synthetic ∞-category theory (and work on the proof assistant `Rzk`) have been contributed by:

Abdelrahman Aly Abounegm, Fredrik Bakke, César Bardomiano Martínez, Jonathan Campbell, Robin Carlier, Theofanis Chatzidiamantis-Christoforidis, Aras Ergus, Matthias Hutzler, Nikolai Kudasov, Kenji Maillard, David Martínez Carpena, Stiéphen Pradal, Nima Rasekh, Emily Riehl, Florrie Verity, Tashi Walde, and Jonathan Weinberger.

Anyone is welcome to join us!

rzk-lang.github.io/sHoTT

Questions for the future

- It is very painful to ellaborate higher categorical proofs all the way down to the foundations. Are enough contributors willing to do this wearisome technical work?
- `Lean` is very powerful and will only become moreso. But will the tactics introduced to spead up formalization make proofs too hard to understand?
- Proofs in `Rzk` of theorems that are way beyond the current capacity of `Lean` are conceptual and short. But the formal system is unfamiliar and so far incomplete. Is this too much of a hurdle for non-expert users?
- Theorems formalized in `Rzk` are useless to users of `Mathlib`. Will we be able to integrate them into `Lean`?
- A healthy ecosystem for mathematical formalization will involve lots of domain specific formal systems. Will AI-powered co-pilots every be able to support formalization in experimental proof assistants?
- Many of us expect an increasing degree of automation in the production of formalized mathematics. How do we ensure that computer formalized mathematics remains understandable by humans?

Thank you!