Emily Riehl

Johns Hopkins University

Formalizing ∞-category theory in the Rzk proof assistant

joint with Nikolai Kudasov and Jonathan Weinberger

Interactions of Proof Assistants and Mathematics

Plan

1. Computer formalization of mathematics
2. In search of foundations for higher structures
3. Simplicial type theory and the Rzk proof assistant
4. Synthetic ∞-category theory
5. A formalized proof of the ∞-categorical Yoneda lemma

1

Computer formalization of mathematics

Motivation

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 A. Grothendieck suggested, in his unpublished memoir [Gr], that this connection should have a higher-dimensional generalisation involving polycategories.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

Motivation

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 A. Grothendieck suggested, in his unpublished memoir [Gr], that this connection should have a higher-dimensional generalisation involving polycategories.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.

Motivation

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."

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*

[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.

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 Kepler conjecture, resolving a 1611 conjecture, 2003–2014, ISABELLE
- the Feit-Thompson Odd Order Theorem, a foundational result in the classification of finite simple groups, 2006–2012, COQ
- 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.

2

In search of foundations for higher structures

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:

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

∞-categories in set theory

Essentially, ∞-categories are 1-categories in which all the sets have been replaced by ∞-groupoids aka homotopy types:

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

Where

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

∞-categories in set theory

Essentially, ∞-categories are 1-categories in which all the sets have been replaced by ∞-groupoids aka homotopy types:

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

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

∞-categories in set theory

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

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

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).

∞-categories in set theory

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

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

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.

Composing paths

In the total singular complex aka the fundamental ∞-groupoid aka the anima or "soul" of a space X, composites of paths are witnessed by higher paths:

$$\begin{tikzcd} & y \arrow[dr, squiggly, "g"] & \\ x \arrow[ur, squiggly, "f"] & & z \end{tikzcd}$$

Composing paths

In the total singular complex aka the fundamental ∞-groupoid aka the anima or "soul" of a space X, composites of paths are witnessed by higher paths:

$$\begin{tikzcd} & y \arrow[dr, squiggly, "g"] \arrow[d, phantom, "\Downarrow\alpha"] & \\ x \arrow[ur, squiggly, "f"] \arrow[rr, squiggly, "k"'] & {} & z \end{tikzcd}$$

Composing paths

In the total singular complex aka the fundamental ∞-groupoid aka the anima or "soul" of a space X, composites of paths are witnessed by higher paths:

$$\begin{tikzcd} & y \arrow[dr, squiggly, "g"] \arrow[d, phantom, "\Downarrow\alpha"] & \\ x \arrow[ur, squiggly, "f"] \arrow[rr, squiggly, "k"'] & {} & z \end{tikzcd}$$

Theorem. The space of composites of two paths f and g in X is contractible.

Composing paths

In the total singular complex aka the fundamental ∞-groupoid aka the anima or "soul" of a space X, composites of paths are witnessed by higher paths:

$$\begin{tikzcd} & y \arrow[dr, squiggly, "g"] \arrow[d, phantom, "\Downarrow\alpha"] & \\ x \arrow[ur, squiggly, "f"] \arrow[rr, squiggly, "k"'] & {} & z \end{tikzcd}$$

Theorem. The space of composites of two paths f and g in X is contractible.

Proof: The space of composites of paths f and g in X is defined by the pullback:

$$\begin{tikzcd} \mathrm{Comp}(f,g) \arrow[r, hook] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Map}(\Delta, X) \arrow[d, two heads, "\text{restrict}"] \\ * \arrow[r, "f \wedge g"'] & \mathrm{Map}(\Lambda, X) \end{tikzcd}$$

Composing paths

In the total singular complex aka the fundamental ∞-groupoid aka the anima or "soul" of a space X, composites of paths are witnessed by higher paths:

$$\begin{tikzcd} & y \arrow[dr, squiggly, "g"] & \\ x \arrow[ur, squiggly, "f"] \arrow[rr, squiggly, "k"'] & {} \arrow[d, phantom, "\Downarrow\alpha"] & z \\ & {} & \end{tikzcd}$$

Theorem. The space of composites of two paths f and g in X is contractible.

Proof: The space of composites of paths f and g in X is defined by the pullback:

$$\begin{tikzcd} S^{n-1} \arrow[r] \arrow[d, hook] & \mathsf{Comp}(f,g) \arrow[r, hook] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathsf{Map}(\Delta, X) \arrow[d, two heads, "\mathsf{restrict}"] \\ D^n \arrow[r] \arrow[ur, dashed] & * \arrow[r, "f \wedge g"'] & \mathsf{Map}(\Lambda, X) \end{tikzcd}$$

A space is contractible just when any sphere S^{n-1} can be filled to a disk D^n for $n \geq 0$.

Composing paths

In the total singular complex aka the fundamental ∞-groupoid aka the anima or "soul" of a space X, composites of paths are witnessed by higher paths:

$$\begin{tikzcd} & y \arrow[dr, squiggly, "g"] & \\ x \arrow[ur, squiggly, "f"] \arrow[rr, squiggly, "k"'] \arrow[rr, phantom, shift left=3, "\Downarrow\alpha"] & & z \end{tikzcd}$$

Theorem. The space of composites of two paths f and g in X is contractible.

Proof: The space of composites of paths f and g in X is defined by the pullback:

$$\begin{tikzcd} S^{n-1} \arrow[r] \arrow[d, hook] & \mathsf{Comp}(f,g) \arrow[r, hook] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathsf{Map}(\Delta, X) \arrow[d, two heads, "\text{restrict}"] \\ D^n \arrow[r] \arrow[ur, dashed] \arrow[urr, dashed, bend right=10] & * \arrow[r, "f \wedge g"'] & \mathsf{Map}(\Lambda, X) \end{tikzcd}$$

A space is contractible just when any sphere S^{n-1} can be filled to a disk D^n for $n \geq 0$.

Composing paths

In the total singular complex aka the fundamental ∞-groupoid aka the anima or "soul" of a space X, composites of paths are witnessed by higher paths:

$$\begin{tikzcd} & y \arrow[dr, squiggly, "g"] & \\ x \arrow[ur, squiggly, "f"] \arrow[rr, squiggly, "k"'] & \arrow[u, phantom, "\Downarrow\alpha"] & z \end{tikzcd}$$

Theorem. The space of composites of two paths f and g in X is contractible.

Proof: The space of composites of paths f and g in X is defined by the pullback:

$$\begin{tikzcd} S^{n-1} \arrow[r] \arrow[d, hook] & \mathsf{Comp}(f,g) \arrow[r, hook] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathsf{Map}(\Delta, X) \arrow[d, two heads, "\mathsf{restrict}"] \\ D^n \arrow[r] \arrow[ur, dashed] \arrow[urr, dashed, bend right=10] & * \arrow[r, "f \wedge g"'] & \mathsf{Map}(\Lambda, X) \end{tikzcd} \quad \leftrightsquigarrow \quad \begin{tikzcd} S^{n-1} \times \Delta \cup_{S^{n-1} \times \Lambda} D^n \times \Lambda \arrow[r] \arrow[d, hook] & X \\ D^n \times \Delta \arrow[ur, dashed] & \end{tikzcd}$$

A space is contractible just when any sphere S^{n-1} can be filled to a disk D^n for $n \geq 0$.

Composing paths

In the total singular complex aka the fundamental ∞-groupoid aka the anima or "soul" of a space X, composites of paths are witnessed by higher paths:

$$\begin{tikzcd}
 & y \arrow[dr, squiggly, "g"] & \\
x \arrow[ur, squiggly, "f"] \arrow[rr, squiggly, "k"'] \arrow[rr, phantom, bend left=20, "\Downarrow\alpha"] & & z
\end{tikzcd}$$

Theorem. The space of composites of two paths f and g in X is contractible.

Proof: The space of composites of paths f and g in X is defined by the pullback:

$$\begin{tikzcd}
S^{n-1} \arrow[r] \arrow[d, hook] & \mathsf{Comp}(f,g) \arrow[r, hook] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathsf{Map}(\Delta, X) \arrow[d, two heads, "\text{restrict}"] \\
D^n \arrow[r] \arrow[ur, dashed] \arrow[urr, dashed, bend right=10] & * \arrow[r, "f\wedge g"'] & \mathsf{Map}(\Lambda, X)
\end{tikzcd}
\quad \leftrightsquigarrow \quad
\begin{tikzcd}
S^{n-1}\times\Delta \cup_{S^{n-1}\times\Lambda} D^n\times\Lambda \arrow[r] \arrow[d, hook] & X \\
D^n\times\Delta \arrow[u, dotted, bend right] \arrow[ur, dashed] &
\end{tikzcd}$$

A space is contractible just when any sphere S^{n-1} can be filled to a disk D^n for $n \geq 0$. The extension exists since the inclusion admits a continuous deformation retract. $\square$

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,...

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?

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 $I \coloneqq [0,1] \subset \mathbb{R}$ 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 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.

∞-categories in 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.

∞-categories in 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 $\mathsf{Hom}_A(x, y)$ over $x, y : A$ whose terms $f : \mathsf{Hom}_A(x, y)$ are called arrows.

∞-categories in 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 $\mathrm{Hom}_A(x, y)$ over $x, y : A$ whose terms $f : \mathrm{Hom}_A(x, y)$ are called arrows.

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

∞-categories in 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 $\mathrm{Hom}_A(x, y)$ over $x, y : A$ whose terms $f : \mathrm{Hom}_A(x, y)$ are called arrows.

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

- Every pair of arrows $f : \mathrm{Hom}_A(x, y)$ and $g : \mathrm{Hom}_A(y, z)$ has a unique composite, defining a term $g \circ f : \mathrm{Hom}_A(x, z)$.

∞-categories in 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 $\mathrm{Hom}_A(x, y)$ over $x, y : A$ whose terms $f : \mathrm{Hom}_A(x, y)$ are called arrows.

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

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

∞-categories in 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?

3

Simplicial type theory and the Rzk proof assistant

Shapes in the theory of the directed interval

Our types may depend on other types and also on shapes $\Phi \subset \mathbb{2}^n$, which are 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.

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

Shapes in the theory of the directed interval

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

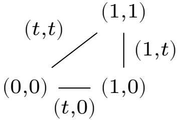
$$\top, \bot, \wedge, \vee, \equiv \qquad \text{and} \qquad 0, 1, \leq$$

satisfying intuitionistic logic and strict interval axioms.

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

$$\Delta^1 := 2$$

$$\Delta^2 := \{(t_1, t_2) : 2^2 \mid t_2 \leq t_1\} := \left\{ \right.$$



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

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

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

Extension types

Formation rule for extension types

$$\frac{\Phi \subset \Psi \text{ shape} \qquad A \text{ type} \qquad a : \Phi \to A}{\left\langle \begin{tikzcd} \Phi \arrow[r, "a"] \arrow[d, tail] & A \\ \Psi \arrow[ur, dashed] & \end{tikzcd} \right\rangle \text{ type}}$$

Extension types

Formation rule for extension types

$$\frac{\Phi \subset \Psi \ \textsf{shape} \qquad A \ \textsf{type} \qquad a : \Phi \to A}{\left\langle \begin{tikzcd} \Phi \arrow[r, "a"] \arrow[d, tail] & A \\ \Psi \arrow[ur, dashed] & \end{tikzcd} \right\rangle \ \textsf{type}}$$

A term $f : \left\langle \begin{tikzcd} \Phi \arrow[r, "a"] \arrow[d, tail] & A \\ \Psi \arrow[ur, dashed] & \end{tikzcd} \right\rangle$ defines

Extension types

Formation rule for extension types

$$\frac{\Phi \subset \Psi \text{ shape} \qquad A \text{ type} \qquad a : \Phi \to A}{\left\langle \begin{tikzcd} \Phi \arrow[r, "a"] \arrow[d, tail] & A \\ \Psi \arrow[ur, dashed] & \end{tikzcd} \right\rangle \text{ type}}$$

A term $f : \left\langle \begin{tikzcd} \Phi \arrow[r, "a"] \arrow[d, tail] & A \\ \Psi \arrow[ur, dashed] & \end{tikzcd} \right\rangle$ defines

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

Extension types

Formation rule for extension types

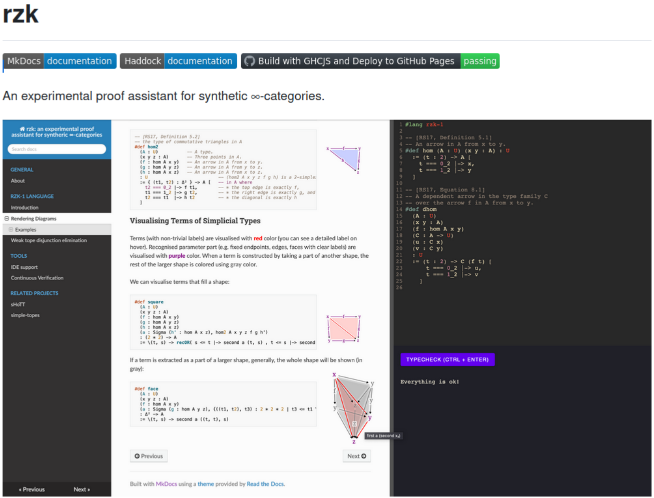
$$\frac{\Phi \subset \Psi \text{ shape} \qquad A \text{ type} \qquad a : \Phi \to A}{\left\langle \begin{tikzcd} \Phi \arrow[r, "a"] \arrow[d, tail] & A \\ \Psi \arrow[ur, dashed] & \end{tikzcd} \right\rangle \text{ type}}$$

A term $f : \left\langle \begin{tikzcd} \Phi \arrow[r, "a"] \arrow[d, tail] & A \\ \Psi \arrow[ur, dashed] & \end{tikzcd} \right\rangle$ 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.

An experimental proof assistant Rzk for ∞-category theory



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

Our initial aim was to write a formalized proof of the ∞-categorical Yoneda lemma.

github.com/emilyriehl/yoneda or emilyriehl.github.io/yoneda/

- proof from Emily Riehl & Mike Shulman, A type theory for synthetic ∞-categories, Higher Structures 2017.
- formalizations written by Nikolai Kudasov, Emily Riehl, Jonathan Weinberger.
- completed March 12 – April 17, 2023

A formalized proof of the ∞-categorical Yoneda lemma

Our initial aim was to write a formalized proof of the ∞-categorical Yoneda lemma.

github.com/emilyriehl/yoneda or emilyriehl.github.io/yoneda/

- proof from Emily Riehl & Mike Shulman, A type theory for synthetic ∞-categories, Higher Structures 2017.
- formalizations written by Nikolai Kudasov, Emily Riehl, Jonathan Weinberger.
- completed March 12 – April 17, 2023

Our ultimate aim is to compare ∞-category theory in simplicial type theory with ordinary category theory in traditional foundations. Thus,

- We've included a formalization of the 1-categorical Yoneda lemma in `Lean` by Sina Hazratpour as part of an Introduction to Proofs course at Johns Hopkins.
- We wrote a first version of `yoneda-lemma-precategories.lagda.md`.

More recently, we've professionalized our library, implementing a style guide suggested by Fredrik Bakke, who also contributed some formalizations.

Future work: formalize synthetic ∞-category theory

Help us formalize other results from synthetic ∞-category in Rzk!

We have suggested some formalization goals at

github.com/rzk-lang/sHoTT or rzk-lang.github.io/sHoTT

that should be achievable if you have prior familiarity with homotopy type theory.

It is also possible to formalize standard (book) homotopy type theory in Rzk, and these background results are used as prerequisites for our main project.

4

Synthetic ∞-category theory

Hom types

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

$$\mathrm{Hom}_A(x,y) := \left\langle \begin{tikzcd} \partial\Delta^1 \arrow[r, "{[x,y]}"] \arrow[d, hook] & A \\ \Delta^1 \arrow[ur, dashed] & \end{tikzcd} \right\rangle \text{ type}$$

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

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

Hom types

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

$$\mathsf{Hom}_A(x,y) := \left\langle \begin{tikzcd} \partial\Delta^1 \arrow[r, "{[x,y]}"] \arrow[d, hook] & A \\ \Delta^1 \arrow[ur, dashed] & \end{tikzcd} \right\rangle \text{ type}$$

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 Joyal). 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.,

$$\left\langle \begin{tikzcd} \Lambda^2_1 \arrow[r, "{[f,g]}"] \arrow[d, hook] & A \\ \Delta^2 \arrow[ur, dashed] & \end{tikzcd} \right\rangle \quad \text{is contractible.}^a$$

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

Pre-∞-categories

defn (Riehl–Shulman after Joyal). 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.,

$$\left\langle \begin{tikzcd} \Lambda^2_1 \arrow[r, "{[f,g]}"] \arrow[d, tail] & A \\ \Delta^2 \arrow[ur, dashed] & \end{tikzcd} \right\rangle \quad \text{is contractible.}^{a}$$

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

By contractibility, $\left\langle \begin{tikzcd} \Lambda^2_1 \arrow[r, "{[f,g]}"] \arrow[d, tail] & A \\ \Delta^2 \arrow[ur, dashed] & \end{tikzcd} \right\rangle$ 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.

Identity arrows

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

$$\mathsf{id}_x := \lambda t.x : \mathsf{Hom}_A(x,x) := \left\langle \begin{tikzcd} \partial\Delta^1 \arrow[r, "{[x,x]}"] \arrow[d, hook] & A \\ \Delta^1 \arrow[ur, dashed] & \end{tikzcd} \right\rangle,$$

which we denote by id_x and call the identity arrow.

Identity arrows

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

$$\mathsf{id}_x := \lambda t.x : \mathsf{Hom}_A(x,x) := \left\langle \begin{tikzcd} \partial\Delta^1 \arrow[r, "{[x,x]}"] \arrow[d, tail] & A \\ \Delta^1 \arrow[ur, dashed] & \end{tikzcd} \right\rangle,$$

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

$$\lambda(s,t).f(t) : \left\langle \begin{tikzcd} \Lambda^2_1 \arrow[r, "{[\mathsf{id}_x, f]}"] \arrow[d, tail] & A \\ \Delta^2 \arrow[ur, dashed] & \end{tikzcd} \right\rangle$$

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

Associativity of composition

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

Associativity of composition

Prop. 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 $\Delta^1 \to A$:

$$
\begin{tikzcd}
 & & y \arrow[dd, "g"'] \arrow[drr, "g"] \arrow[dddrr, "h\circ g"] & & \\
x \arrow[urr, "f"] \arrow[drr, "g\circ f"'] \arrow[dd, "f"'] & & & & z \arrow[dd, "h"] \\
 & & z \arrow[drr, "h"'] & & \\
y \arrow[urr, "g"'] & & & & w
\end{tikzcd}
$$

Associativity of composition

Prop. 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 $\Delta^1 \to A$:

$$\begin{tikzcd}
& & y \arrow[drr, "g"] \arrow[dd, "g"] \arrow[dddrr, "h\circ g"] & & \\
x \arrow[urr, "f"] \arrow[rrrr, dashed] \arrow[drr, "g\circ f"] \arrow[dd, "f"'] & & & & z \arrow[dd, "h"] \\
& & z \arrow[drr, "h"] & & \\
y \arrow[urr, "g"] \arrow[rrrr, dashed] & & & & w
\end{tikzcd}$$

Composing defines a term in the type $\Delta^2 \to (\Delta^1 \to A)$

Associativity of composition

Prop. 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 $\Delta^1 \to A$:

$$
\begin{tikzcd}
 & y \arrow[dr, "g"] \arrow[dd, "g"'] \arrow[dddr, "h\circ g"] & \\
x \arrow[ur, "f"] \arrow[rr, dashed] \arrow[dr, "g\circ f"'] \arrow[ddr, dashed, bend left, "\ell"] \arrow[dd, "f"'] & & z \arrow[dd, "h"] \\
 & z \arrow[dr, "h"'] & \\
y \arrow[ur, "g"'] \arrow[rr, dashed] & & w
\end{tikzcd}
$$

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

Isomorphisms

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

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

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

Isomorphisms

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

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

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

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

Isomorphisms

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

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

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

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

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

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

∞-categories

By path induction, to define a map

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

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

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

∞-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

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

defn (Riehl–Shulman after Rezk). A pre-∞-category A is ∞-category iff every isomorphism is an identity

∞-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

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

defn (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.

∞-groupoids

Similarly by path induction define

$$\text{arr-eq} \colon (x =_A y) \to \text{Hom}_A(x, y)$$

for all $x, y : A$ by $\text{arr-eq}(\text{refl}_x) :\equiv \text{id}_x$.

∞-groupoids

Similarly by path induction define

$$\mathsf{arr\text{-}eq} \colon (x =_A y) \to \mathsf{Hom}_A(x, y)$$

for all $x, y : A$ by $\mathsf{arr\text{-}eq}(\mathsf{refl}_x) := \mathsf{id}_x$.

A type A is an ∞-groupoid iff every arrow is an identity, i.e., iff arr-eq is an equivalence.

∞-groupoids

Similarly by path induction define

$$\text{arr-eq} : (x =_A y) \to \text{Hom}_A(x, y)$$

for all $x, y : A$ by $\text{arr-eq}(\text{refl}_x) := \text{id}_x$.

A type A is an ∞-groupoid iff every arrow is an identity, i.e., iff arr-eq is an equivalence.

Prop. A type is an ∞-groupoid if and only if it is an ∞-category and all of its arrows are isomorphisms.

Proof:

$$\begin{tikzcd}
x =_A y \arrow[rr, "\text{arr-eq}"] \arrow[dr, "\text{iso-eq}"'] & & \text{Hom}_A(x, y) \\
& x \cong_A y \arrow[ur, hook] &
\end{tikzcd}$$

defn. An ∞-groupoid is a type in which arrows are equivalent to identities:

$$\mathsf{arr\text{-}eq} \colon (x =_A y) \to \mathsf{Hom}_A(x, y) \text{ is an equivalence.}$$

∞-categories for undergraduates

defn. An ∞-groupoid is a type in which arrows are equivalent to identities:

$$\text{arr-eq} \colon (x =_A y) \to \mathrm{Hom}_A(x, y) \text{ is an equivalence.}$$

defn. An ∞-category is a type

- which has unique binary composites of arrows:

$$\left\langle \begin{tikzcd} \Lambda^2_1 \arrow[r, "{[f,g]}"] \arrow[d, tail] & A \\ \Delta^2 \arrow[ur, dashed] & \end{tikzcd} \right\rangle \quad \text{is contractible}$$

∞-categories for undergraduates

defn. An ∞-groupoid is a type in which arrows are equivalent to identities:

$$\mathsf{arr\text{-}eq} \colon (x =_A y) \to \mathrm{Hom}_A(x, y) \text{ is an equivalence.}$$

defn. An ∞-category is a type

- which has unique binary composites of arrows:

$$\left\langle \begin{tikzcd} \Lambda^2_1 \arrow[r, "{[f,g]}"] \arrow[d, hook] & A \\ \Delta^2 \arrow[ur, dashed] & \end{tikzcd} \right\rangle \quad \text{is contractible}$$

- and in which isomorphisms are equivalent to identities:

$$\mathsf{iso\text{-}eq} \colon (x =_A y) \to (x \cong_A y) \text{ is an equivalence.}$$

5

A formalized proof of the ∞-categorical Yoneda lemma

Covariant type families

defn (Riehl–Shulman after Joyal). A type family $B(x)$ over $x : A$ is covariant if for every $f : \mathrm{Hom}_A(x, y)$ and $u : B(x)$ there is a unique lift of f with domain u.

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

Covariant type families

defn (Riehl–Shulman after Joyal). A type family $B(x)$ over $x : A$ is covariant if for every $f : \mathrm{Hom}_A(x, y)$ and $u : B(x)$ there is a unique lift of f with domain u

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

Prop. Fix $a : A$. The type family $\mathrm{Hom}_A(a, x)$ over $x : A$ is covariant if and only if A is a pre-∞-category.

Covariant type families

defn (Riehl–Shulman after Joyal). A type family $B(x)$ over $x : A$ is covariant if for every $f : \mathrm{Hom}_A(x, y)$ and $u : B(x)$ there is a unique lift of f with domain u

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

Prop. Fix $a : A$. The type family $\mathrm{Hom}_A(a, x)$ over $x : A$ is covariant if and only if A is a pre-∞-category.

Prop. When A is a pre-∞-category, for any $u : B(x)$, $f : \mathrm{Hom}_A(x, y)$, and $g : \mathrm{Hom}_A(y, z)$, then $g_*(f_* u) = (g \circ f)_* u$ and $(\mathrm{id}_x)_* u = u$.

Covariant type families

defn (Riehl–Shulman after Joyal). A type family $B(x)$ over $x : A$ is *covariant* if for every $f : \mathrm{Hom}_A(x, y)$ and $u : B(x)$ there is a unique lift of f with domain u

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

Prop. Fix $a : A$. The type family $\mathrm{Hom}_A(a, x)$ over $x : A$ is covariant if and only if A is a pre-∞-category.

Prop. When A is a pre-∞-category, for any $u : B(x)$, $f : \mathrm{Hom}_A(x, y)$, and $g : \mathrm{Hom}_A(y, z)$, then $g_*(f_* u) = (g \circ f)_* u$ and $(\mathrm{id}_x)_* u = u$.

Prop. For any covariant families $B(x)$ and $C(x)$ over $x : A$, a pre-∞-category, any family of maps $\phi : \prod_{x:A} B(x) \to C(x)$ is natural.

Covariant type families

defn (Riehl–Shulman after Joyal). A type family $B(x)$ over $x : A$ is covariant if for every $f : \mathrm{Hom}_A(x, y)$ and $u : B(x)$ there is a unique lift of f with domain u

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

Prop. Fix $a : A$. The type family $\mathrm{Hom}_A(a, x)$ over $x : A$ is covariant if and only if A is a pre-∞-category.

Prop. When A is a pre-∞-category, for any $u : B(x)$, $f : \mathrm{Hom}_A(x, y)$, and $g : \mathrm{Hom}_A(y, z)$, then $g_*(f_* u) = (g \circ f)_* u$ and $(\mathrm{id}_x)_* u = u$.

Prop. For any covariant families $B(x)$ and $C(x)$ over $x : A$, a pre-∞-category, any family of maps $\phi : \prod_{x:A} B(x) \to C(x)$ is natural.

Prop. If $B(x)$ is covariant over $x : A$, a pre-∞-category, then each fiber $B(x)$ is an ∞-groupoid.

The Yoneda lemma

Let $B(x)$ be a covariant family over $x : A$, a pre-∞-category, and fix $a : A$.

The Yoneda lemma

Let $B(x)$ be a covariant family over $x : A$, a pre-∞-category, and fix $a : A$.

Yoneda lemma. The maps

$$\mathsf{evid} :\equiv \lambda\phi.\phi(a, \mathsf{id}_a) : \left(\prod_{x:A} \mathsf{Hom}_A(a,x) \to B(x)\right) \to B(a) \quad \text{and}$$

$$\mathsf{yon} :\equiv \lambda u.\lambda x.\lambda f.f_* u : B(a) \to \left(\prod_{x:A} \mathsf{Hom}_A(a,x) \to B(x)\right)$$

are inverse equivalences.

The Yoneda lemma

Let $B(x)$ be a covariant family over $x : A$, a pre-∞-category, and fix $a : A$.

Yoneda lemma. The maps

$$\mathsf{evid} :\equiv \lambda\phi.\phi(a, \mathsf{id}_a) : \left(\prod_{x:A} \mathsf{Hom}_A(a, x) \to B(x)\right) \to B(a) \quad \text{and}$$

$$\mathsf{yon} :\equiv \lambda u.\lambda x.\lambda f.f_* u : B(a) \to \left(\prod_{x:A} \mathsf{Hom}_A(a, x) \to B(x)\right)$$

are inverse equivalences.

Proof: By definition, $\mathsf{evid} \circ \mathsf{yon}(u) :\equiv (\mathsf{id}_a)_* u$. By functoriality $(\mathsf{id}_a)_* u = u$, so yon is a section of evid.

The Yoneda lemma

Let $B(x)$ be a covariant family over $x : A$, a pre-∞-category, and fix $a : A$.

Yoneda lemma. The maps

$$\mathsf{evid} :\equiv \lambda\phi.\phi(a, \mathsf{id}_a) : \left(\prod_{x:A} \mathsf{Hom}_A(a,x) \to B(x)\right) \to B(a) \quad \text{and}$$

$$\mathsf{yon} :\equiv \lambda u.\lambda x.\lambda f. f_* u : B(a) \to \left(\prod_{x:A} \mathsf{Hom}_A(a,x) \to B(x)\right)$$

are inverse equivalences.

Proof: By definition, $\mathsf{evid} \circ \mathsf{yon}(u) :\equiv (\mathsf{id}_a)_* u$. By functoriality $(\mathsf{id}_a)_* u = u$, so yon is a section of evid. To see that yon is a retraction of evid, start from the definition $\mathsf{yon} \circ \mathsf{evid}(\phi)(x, f) :\equiv f_*\phi(a, \mathsf{id}_a)$. By naturality of ϕ and the identity law for pre-∞-categories $f_*\phi(a, \mathsf{id}_a) = \phi(x, f \circ \mathsf{id}_a) = \phi(x, f)$. $\square$

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 and Michael Shulman, A type theory for synthetic ∞-categories, Higher Structures 1(1):116–193, 2017; arXiv:1705.07442
- Emily Riehl, Could ∞-category theory be taught to undergraduates?, Notices of the AMS 70(5):727–736, May 2023; arXiv:2302.07855
- 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
- Nikolai Kudasov, Emily Riehl, Jonathan Weinberger, Formalizing the ∞-categorical Yoneda lemma, 1–13; arXiv:2309.08340

Danke!