



Emily Riehl

Johns Hopkins University

Thoughts on the future of type theory in mathematics

TYPES 2026

A(n a)typical personal history with type theory

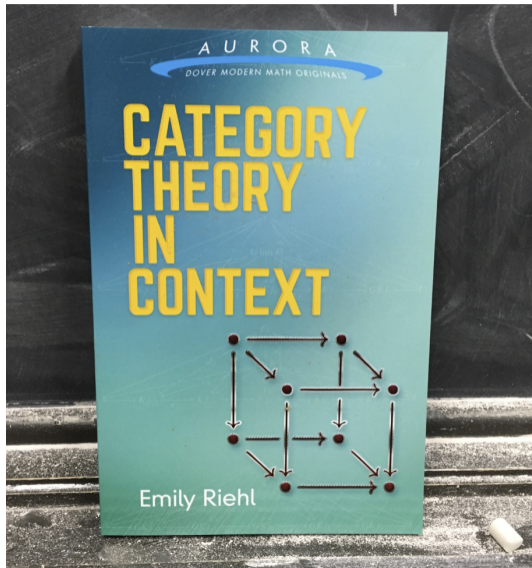


My personal history with type theory progressed along the following lines:

- **ignorance:** Like a typical mathematician, I could not state the axioms of set theory/first-order logic when I earned my PhD nor had much interest in foundations.
- **exposure:** However, I knew some homotopy type theorists and started to hear about new developments in seminars and at conferences.
- **experimentation:** After my interest was piqued — and in order to have something to say at the conferences I was getting invited to — I started work on a few projects.
- **understanding:** There were certain details in proofs I did not understand well until I taught a topics course in homotopy type theory and wrote problem sets in [Agda](#).
- **expertise:** While I often find myself evangelizing for type theory to other mathematicians, I do not consider myself an expert.

These experiences have changed how I think about mathematics
and made me a better mathematician.

Category Theory in Context v2



I wrote a category theory book in 2015, aimed at advanced mathematics students.

This Spring, I am working on a revised second edition.

While I've learned a few new theorems in category theory since then, more significant changes have been catalyzed by my exposure to type theory.

Improved mathematical communication



Compare the original

PROPOSITION 4.2.6. *Given a pair of functors $F: \mathbf{C} \rightleftarrows \mathbf{D}: G$, there exists a natural isomorphism $\mathbf{D}(Fc, d) \cong \mathbf{C}(c, Gd)$ if and only if there exists a pair of natural transformations $\eta: 1_{\mathbf{C}} \Rightarrow GF$ and $\epsilon: FG \Rightarrow 1_{\mathbf{D}}$ satisfying the triangle identities.*

with the revision

THEOREM 4.2.7. *Given a pair of functors $F: \mathbf{C} \rightleftarrows \mathbf{D}: G$ the data of a **fully-specified adjunction**, with $F \dashv G$, is equivalently encoded by any of the following:*

- (i) *a natural family of isomorphisms $\mathbf{D}(Fc, d) \cong \mathbf{C}(c, Gd)$ for all $c \in \mathbf{C}$ and $d \in \mathbf{D}$,*
- (ii) *natural transformations $\eta: \text{id}_{\mathbf{C}} \Rightarrow GF$ and $\epsilon: FG \Rightarrow \text{id}_{\mathbf{D}}$ so that $G\epsilon \cdot \eta G = \text{id}_G$ and $\epsilon F \cdot F\eta = \text{id}_F$,*

I've heard other mathematicians credit their experiences with computer formalization with improving their precision with pen-and-paper theorem statements but in my case this is as much the result of my exposure to dependent type theory.



1. Pedagogy: a reintroduction to proofs?
2. Human–computer interactions: a norm for AI-generated mathematics?
3. Strange new worlds: a mathematical multiverse of synthetic mathematics?



1

Pedagogy: a reintroduction to proofs?

An introduction to proofs



In Spring 2019, I pioneered an **Introduction to Proofs** course, which was not taught in the mathematics department at Johns Hopkins at the time, with relatively standard contents and following a reasonable textbook:

- truth tables and logical connectives
- operations on sets
- quantifiers
- induction over the natural numbers
- functions
- infinite sets

plus a few more interesting features:

- a week-long introduction to construction mathematics and computer proof assistants (**Rocq**)
- a month long “inquiry based learning” exploration of metric spaces.

The next fall, I reworked all the lectures, using a different textbook ...

An infinite descent into pure mathematics

Forget truth tables! Instead:

◆ Definition 1.1.21

The **implication** operator is the logical operator \Rightarrow (L^AT_EX code: `\Rightarrow`), defined according to the following rules:

- (\Rightarrow I) If q can be derived from the assumption that p is true, then $p \Rightarrow q$ is true;
- (\Rightarrow E) If $p \Rightarrow q$ is true and p is true, then q is true.

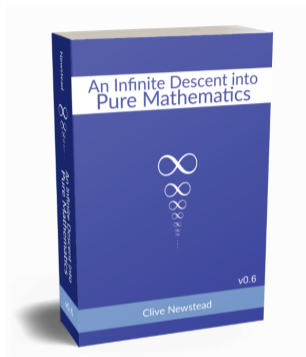
The expression $p \Rightarrow q$ represents ‘if p , then q ’.

◆ Strategy 1.1.22 (Proving implications)

In order to prove a proposition of the form $p \Rightarrow q$, it suffices to assume that p is true, and then derive q from that assumption.

◆ Strategy 1.1.25 (Assuming implications—modus ponens)

If an assumption in a proof has the form $p \Rightarrow q$, and p is also assumed to be true, then we may deduce that q is true.



Clive Newstead

PhD Thesis:
“Algebraic models of
dependent type theory”

An introduction to proofs level construction in type theory



Theorem. For any types A , B , and C , we may define functions

$$\begin{aligned} \text{glue} &: ((A \rightarrow C) \times (B \rightarrow C)) \rightarrow ((A + B) \rightarrow C) \text{ and} \\ \text{split} &: ((A + B) \rightarrow C) \rightarrow ((A \rightarrow C) \times (B \rightarrow C)). \end{aligned}$$

Construction: By \rightarrow intro, suppose given $h : (A \rightarrow C) \times (B \rightarrow C)$; our goal is an element of type $(A + B) \rightarrow C$. By \times elim₁ and \times elim₂, we obtain elements $\pi_1 h : A \rightarrow C$ and $\pi_2 h : B \rightarrow C$. By \rightarrow intro, to define an element of type $(A + B) \rightarrow C$ it suffices to assume an element $x : A + B$ and define an element of type C . By $+$ elim, it suffices to construct an element of C in two cases: assuming an arbitrary element $a : A$ or assuming an arbitrary element $b : B$. By \rightarrow elim, we obtain an element $\pi_1 h(a) : C$ in the first case and obtain $\pi_2 h(b) : C$ in the second case, completing the construction of **glue**.

For **split**, by \rightarrow intro and \times intro, suppose given $k : A + B \rightarrow C$; our goal is to define elements of type $A \rightarrow C$ and $B \rightarrow C$. By \rightarrow intro, assume $a : A$ and use $+$ intro₁ and \rightarrow elim to obtain an element $k(\iota_1 a) : C$ defining a function of type $A \rightarrow C$. Similarly, by \rightarrow intro, $+$ intro₂, and \rightarrow elim, we obtain $\lambda b.k(\iota_2 b) : B \rightarrow C$, defining **split**. \square

Corollary: an introduction to constructive mathematics



The **negation** of a proposition P is the proposition $\neg P := P \Rightarrow \perp$. Here \perp is the false proposition, which can be thought of as the type with no elements.

The general constructions of functions **glue** : $((A \rightarrow C) \times (B \rightarrow C)) \rightarrow ((A + B) \rightarrow C)$ and **split** : $((A + B) \rightarrow C) \rightarrow ((A \rightarrow C) \times (B \rightarrow C))$ specialize when C is \perp to prove:

Theorem. For any propositions P and Q

$$((\neg P) \wedge (\neg Q)) \Rightarrow \neg(P \vee Q) \quad \text{and} \quad (\neg(P \vee Q)) \Rightarrow ((\neg P) \wedge (\neg Q)).$$

Similar constructions prove the following tautologies of constructive logic:

$$(P \Rightarrow Q) \Rightarrow ((\neg Q) \Rightarrow (\neg P)) \quad ((\neg P) \vee (\neg Q)) \Rightarrow \neg(P \wedge Q) \quad (P \wedge \neg Q) \Rightarrow \neg(P \Rightarrow Q)$$

but not their converses, which need axioms such as the **law of excluded middle**.

Reimagining the foundations of mathematics?



What if we taught students informal dependent type theory as an implicit foundation for mathematics?

Questions:

- Could we adopt terminological conventions that are compatible with their later coursework?

Possible convention: “types” unify “sets” and “propositions.”

$$\text{is-Prop}(A) := \prod_{x,y:A} x =_A y$$

- How do we avoid alienating set theorists/logicians?

A reintroduction to proofs



Reintroduction to Proofs

An introduction to proofs in dependent type theory.


$$\begin{array}{l} A B C : \text{Type} \vdash (A \times B \rightarrow C) = (A \rightarrow B \rightarrow C) \\ P Q R : \text{Prop} \vdash (P \wedge Q \rightarrow R) \leftrightarrow (P \rightarrow Q \rightarrow R) \end{array}$$

This game introduces mathematical proof writing in a formal vocabulary that is designed to make it easier to be fully precise about every step in a proof, as is required to convince a computer that a logical argument is sound.

The language of dependent type theory can be used in parallel to state and prove mathematical theorems as well as to specify and define examples of mathematical objects.

Prerequisites

Worlds 17

Levels 155

Language en

In Fall 2025, I taught an experimental introduction to proofs in informal dependent type theory using [Lean](#).

The result is the [Reintroduction to Proofs](#) game hosted on the [The Lean Game Server](#):

adam.math.hhu.de

with source files on github:

github.com/emilyriehl/ReintroductionToProofs

Contributions welcome!



2

Human-computer interactions: a norm for
AI-generated mathematics?



How should the human mathematics community respond to advances in artificial intelligence?

Will this lead to new paradigms for mathematical proof?

Vibe proving



Is it true that an ∞ -category is idempotent complete (in the ∞ -categorical sense) if and only if its homotopy category is idempotent complete (in the 1-categorical sense)?

◆ Show thinking ▾

Yes, it is true that an ∞ -category \mathcal{C} is idempotent complete (in the ∞ -categorical sense) if and only if its homotopy category $h\mathcal{O}(\mathcal{C})$ is idempotent complete (in the 1-categorical sense). ☺

Let's establish the definitions and then prove the equivalence.

...

Conclusion:

The two conditions are equivalent. The definitions of idempotent completeness are "homotopy-invariant" in a way that translates directly between the ∞ -category and its homotopy category. This result is also confirmed by the fact that $h(Kar(\mathcal{C})) \cong Kar(h\mathcal{C})$, where $Kar(-)$ denotes the Karoubi envelope (idempotent completion) and $h(-)$ denotes taking the homotopy category. An (∞) -category is idempotent complete if and only if the canonical functor into its Karoubi envelope is an equivalence.

The omitted text contains a clearly written argument with a subtle error, the sort of optimistic mistake often made by human mathematicians. **The conclusion is incorrect.**

Your answer uses the "incoherent" definition of an idempotent in an ∞ -category? Can you answer the same question using the coherent definition of an idempotent in an ∞ -category?

◆ Show thinking ▾

You're right to press on the subtleties of "coherence" in ∞ -categories. Let's clarify how these definitions are typically handled and why the previous conclusion still holds.

...

If a framework requires a more elaborate "coherent idempotent object" (e.g., an object in a category of idempotents, or an algebra for a certain monad), the definition of "idempotent completeness of \mathcal{C} " itself usually still boils down to the splitting condition for any e satisfying $e \circ e \simeq e$. The power of ∞ -category theory is that such "weak" conditions, when interpreted within a coherently structured environment, often suffice and lead to robust theories.

Thus, the answer remains the same: **Yes, an ∞ -category is idempotent complete (in the standard ∞ -categorical sense) if and only if its homotopy category is idempotent complete (in the 1-categorical sense).** The "coherence" is sufficiently handled by the definitions used.

How do we defend the mathematical literature against vibe proving?



A lot of the mathematical text generated by top “reasoning” models is pretty good and all of it looks good.*

*Caveat: the large language model architecture is not well-suited for numerical calculations.

But pretty good is not good enough: one minor error could make an entire logical argument reach the wrong conclusion: true instead of false.

As a journal editor specializing in a subfield where papers tend to be long, technical, and difficult to referee, I am **extremely concerned** about AI generated mathematical “proofs.”



Why has mathematics largely avoided the replication crisis that has confronted other fields?

Peer review in theory: careful refereeing should lead to an error-free publications.

Unfortunately, the mathematical literature contains famous mistakes as well as contradictory theorems.

Peer review in practice: in theory any proof should be **reproducible** by any reader — allowing the reader to understand for themselves why the result is true.

When papers have enough readers, mistakes are eventually caught (often by the authors themselves).

Importantly: human mathematicians are careful in claiming they have a proof.

Proof as peer review?



But sometimes these ideals break down:

One Fields medalist was dismayed to find mistakes in his published, well-studied papers:

“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.”

— Vladimir Voevodsky

Another Fields medalist expressed doubts about a particular proof he had discovered — and also doubted that anyone else would check it:

“...while I was very happy to see many study groups on condensed mathematics throughout the world, to my knowledge all of them have stopped short of this proof. (Yes, this proof is not much fun...)”

— Peter Scholze

A new paradigm for proof checking



"I think it was at this moment that I largely stopped doing what is called 'curiosity-driven research' and started to think seriously about the future. I didn't have the tools to explore the areas where curiosity was leading me and the areas that I considered to be of value and of interest and of beauty..."

— Vladimir Voevodsky

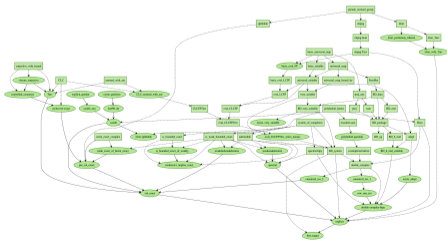
Voevodsky and Scholze both turned to computer formalization to resolve their doubts about the veracity of their own proofs.

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 more productive realm for human-AI interactions?



For *Category Theory in Context* v2, I wanted to add a counterexample illustrating that the unit of a left Kan extension along a fully faithful functor may not be invertible when the Kan extension is not **pointwise**.

An LLM-powered literature search gave me a lot of nonsense including:

- “This is well-known counterexample to Riehl’s *Category Theory in Context* Exercise 6.2.v.” (which does not exist).
- Various claimed examples of a non-pointwise left Kan extension with invertible unit that fail to actually define left Kan extensions.

After pointing out these errors, Chat GPT 5.2 gave me the following “honest admission”:

Honest Admission

I'm struggling to produce a fully explicit counterexample. The standard references (Kelly's *Basic Concepts of Enriched Category Theory*, Riehl's *Category Theory in Context*) mention that non-pointwise Kan extensions exist but rarely give concrete examples.

Instead, I developed my own counterexamples and asked Harmonic’s **Lean** autoformalization agent Aristotle to check that they were correct.

A norm for machine-generated mathematical proof



Despite well-known imperfections, the mathematical community can take deep pride in our overwhelmingly reliable and continually improving standards for mathematical proof.

We should demand the same for AI when it comes to the mathematical realm.

Maintaining high standards will frustrate near term progress, delaying the arrival of a machine we validate as having “artificial mathematical intelligence,” but should be beneficial for overall reliability in the long run, in mathematics and beyond.

Specifically, I want to propose the following norm for the mathematical community when it comes to original mathematics produced by an AI system:

Any artificially generated mathematical text will **not be considered as a proof** unless:

- It has been communicated in both a natural language text paired with a computer formalization of all definitions, theorems, and proofs.
- The formalization has been accepted by the proof assistant and human expert referees have vetted both the formalization and the paired text.



Any artificially generated mathematical text will **not be considered as a proof** unless:

- It has been communicated in both a natural language text paired with a computer formalization of all definitions, theorems, and proofs.
- The formalization has been accepted by the proof assistant and human expert referees have vetted both the formalization and the paired text.

This proposed norm raises various questions:

- What about mathematical areas that are (currently) beyond the scope of formalization?
- Can/should we support the development of automated theorem-proving / autoformalization agents targeting a wide range of proof assistants?
- Will anyone want to read AI-generated formal proofs at all? Are there cases in which they do more harm than good?



3

Strange new worlds: a mathematical multiverse
of synthetic mathematics?

No place for ugly mathematics



“The mathematician’s patterns, like the painter’s or the poet’s must be beautiful; the ideas like the colours or the words, must fit together in a harmonious way. Beauty is the first test: there is no permanent place in the world for ugly mathematics.”

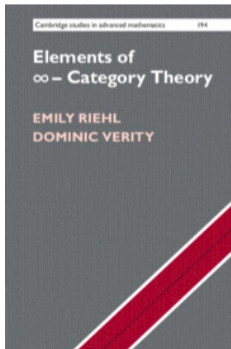
– G.H. Hardy, *A Mathematician's Apology*

I would argue this holds for formal proofs as well as pen-and-paper proofs.

A pen-and-paper proof



A joint book with [Dominic Verity](#) reviews the construction of the reflective embedding of 1-categories into ∞ -categories in less than one page:



1.1.10. DEFINITION (the homotopy category [44, §2.4]). By 1-truncating, any simplicial set X has an underlying reflexive directed graph with the 0-simplices of X defining the objects and the 1-simplices defining the arrows:

$$X_1 \begin{array}{c} \xrightarrow{\delta^1} \\ \xleftarrow{\delta^0} \\ \xrightarrow{\delta^0} \end{array} X_0,$$

By convention, the source of an arrow $f \in X_1$ is its 0th face $f \cdot \delta^1$ (the face opposite 1) while the target is its 1st face $f \cdot \delta^0$ (the face opposite 0). The **free category** on this reflexive directed graph has X_0 as its object set, degenerate 1-simplices serving as identity morphisms, and nonidentity morphisms defined to be finite directed paths of nondegenerate 1-simplices. The **homotopy category** $\mathbf{h}X$ of X is the quotient of the free category on its underlying reflexive directed graph by the congruence¹ generated by imposing a composition relation $h = g \circ f$ witnessed by 2-simplices

$$\begin{array}{ccc} & X_1 & \\ f \nearrow & & \searrow g \\ x_0 & \xrightarrow{h} & x_2 \end{array}$$

This relation implies in particular that homotopic 1-simplices represent the same arrow in the homotopy category.

The homotopy category of the nerve of a 1-category is isomorphic to the original category, as the 2-simplices in the nerve witness all of the composition relations satisfied by the arrows in the underlying reflexive directed graph. Indeed, the natural isomorphism $\mathbf{h}C \cong C$ forms the counit of an adjunction, embedding \mathbf{Cat} as a reflective subcategory of \mathbf{sSet} .

1.1.11. PROPOSITION. *The nerve embedding admits a left adjoint, namely the functor which sends a simplicial set to its homotopy category:*

$$\mathbf{Cat} \begin{array}{c} \xleftarrow{h} \\ \perp \\ \xrightarrow{\quad} \end{array} \mathbf{sSet}$$

The adjunction of Proposition 1.1.11 exists for formal reasons (see Exercise 1.1.i), but nevertheless, a direct proof can be enlightening.


PROOF. For any simplicial set X , there is a natural map from X to the nerve of its homotopy category $\mathbf{h}X$; since nerves are 2-coskeletal, it suffices to define the map $\mathbf{sk}_2 X \rightarrow \mathbf{h}X$, and this is given immediately by the construction of Definition 1.1.10. Note that the quotient map $X \rightarrow \mathbf{h}X$ becomes an isomorphism upon applying the homotopy category functor and is already an isomorphism whenever X is the nerve of a category. Thus the adjointness follows from Lemma B.4.2 or by direct verification of the triangle equalities. \square

A formalized proof

It took three months (part time) of joint work with [Mario Carneiro](#) to formalize this result in [Lean](#).



It then took another six months for this code, which totalled 2240 lines split across seven PRs to pass the review process to be integrated into [Lean's Mathlib](#).

In summary, there was an $19\times$ scaling factor from the original pen-and-paper proof to a pen-and-paper account of the formalization.




We wrote an 19 page paper to explain the formalization and the challenges we encountered along the way:

Formalizing colimits in $\mathcal{C}at$

[Mario Carneiro](#)  

Chalmers University of Technology, Sweden

[Emily Riehl](#)¹  

Department of Mathematics, Johns Hopkins University, 3400 N Charles Street, Baltimore, MD, USA

Abstract

Certain results involving “higher structures” are not currently accessible to computer formalization because the prerequisite ∞ -category theory has not been formalized. To support future work on formalizing ∞ -category theory in Lean’s mathematics library, we formalize some fundamental constructions involving the 1-category of categories. Specifically, we construct the left adjoint to the nerve embedding of categories into simplicial sets, defining the homotopy category functor. We prove further that this adjunction is reflective, which allows us to conclude that $\mathcal{C}at$ has colimits. To our knowledge this is the first formalized proof that the category of categories is cocomplete.

What happened here?

A taxonomy of invisible mathematics



By **invisible mathematics**, we refer to any mathematics that is not visible in a pen-and-paper proof but becomes visible in a computer formalized proof. For example:

- design choices,
- notation/encoding,
- universe levels,
- inclusions and coercions,
- dependent equality,
- coherence theorems,
- equivalent definitions,
- technical constructions,
- mathematical theories, or
- (im)precise statements.

Is there any hope of ever formalizing ∞ -category theory in **Lean**?

Is there another way?

Is there another way?

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

Voevodsky's solution to the problem of invisible mathematics was to create a new foundation system to facilitate efficient synthetic reasoning about higher structures.

"At the time, when I started to look for a practical proof assistant around 2000, I could not find any. ... The primary challenge that needed to be addressed was that the foundations of mathematics were unprepared for the requirements of the task."

Simplicial HoTT and synthetic ∞ -categories 

Info

This project originated as a fork of [emilyriehl/yoneda](#).

This is a formalization library for simplicial Homotopy Type Theory (sHoTT) with the aim of proving resulting in synthetic ∞ -category theory, starting with the results from the following papers:

- "A type theory for synthetic ∞ -categories" ¹
- "Synthetic fibered $(\infty,1)$ -category theory" ²
- "Limits and colimits of synthetic ∞ -categories" ³

This formalization project follows the philosophy laid out in the article "[Could \$\infty\$ -category theory be taught to undergraduates?](#)" ⁴.

The formalizations are implemented using [rzk](#), an experimental proof assistant for a variant of type theory with shapes.

Formalizing the ∞ -categorical Yoneda lemma



The ∞ -categorical Yoneda lemma is out of scope in **Lean** for the foreseeable future, but we have formalized it in the experimental proof assistant **Rzk**.

```
#def Contra-yoneda-lemma uses (funext)
  ( A : U)
  ( is-pre- $\infty$ -category-A : Is-pre- $\infty$ -category A)
  ( a b : A)
  : is-equiv ((z : A)  $\rightarrow$  Hom A z a  $\rightarrow$  Hom A z b) (Hom A a b) (Contra-evid A a b)
  :=
  ( ( ( Contra-yon A is-pre- $\infty$ -category-A a b)
    , ( Contra-yon-evid A is-pre- $\infty$ -category-A a b))
  , ( ( Contra-yon A is-pre- $\infty$ -category-A a b)
    , ( Contra-evid-yon A is-pre- $\infty$ -category-A a b)))
```

Given two terms $a, b : A$ in a pre- ∞ -category, the type **Hom A a b** of arrows from a to b in A is equivalent to the type of natural transformations between the contravariant representable functors at a and b .

In the synthetic language of **sHoTT** used here, any fiberwise function $\phi : (z : A) \rightarrow \mathbf{Hom} A z a \rightarrow \mathbf{Hom} A z b$ is automatically a natural transformation!

Formalizing the ∞ -categorical Yoneda lemma



In `sHoTT`, the ∞ -categorical Yoneda asserts that two easily definable functors are inverse equivalences:

```
#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)
```

```
#def Contra-yon
  ( A : U)
  ( is-pre- $\infty$ -category-A : Is-pre- $\infty$ -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- $\infty$ -category A is-pre- $\infty$ -category-A z a b f v
```

Formalizing the ∞ -categorical Yoneda lemma



By one of the identity laws — a theorem for, rather than an axiom about pre- ∞ -categories — **contra-yon** followed by **contra-evid** is homotopic to the identity:

```
#def Contra-evid-yon
  ( A : U)
  ( is-pre- $\infty$ -category-A : Is-pre- $\infty$ -category A)
  ( a b : A)
  ( v : Hom A a b)
  : Contra-evid A a b (Contra-yon A is-pre- $\infty$ -category-A a b v) = v
:=
  Id-comp-is-pre- $\infty$ -category A is-pre- $\infty$ -category-A a b v
```

Formalizing the ∞ -categorical Yoneda lemma



The other composite carries ϕ to an a priori distinct natural transformation. We first show that these are pointwise equal at all $x : A$ and $f : \text{Hom } A \times a$ in two steps.

```
#section contra-yon-evid

#variable A : U
#variable is-pre- $\infty$ -category-A : Is-pre- $\infty$ -category A
#variables a b : A
```

The composite `Contra-yon-evid` of ϕ equals ϕ at all $x : A$ and $f : \text{Hom } A \times a$.

To show that `contra-evid` followed by `contra-yon` is homotopic to the identity, we compare a natural transformation

$$\phi : (z : A) \rightarrow \text{Hom } A \ z \ a \rightarrow \text{Hom } A \ z \ b$$

with its image under the composite of these functions and verify that both natural transformations define the same function given $x : A$ and $f : \text{Hom } A \ x \ a$.

Formalizing the ∞ -categorical Yoneda lemma



```
#def Contra-yon-evid-twice-pointwise
  (  $\phi$  : (z : A)  $\rightarrow$  Hom A z a  $\rightarrow$  Hom A z b)
  ( x : A)
  ( f : Hom A x a)
  : ( ( Contra-yon A is-pre- $\infty$ -category-A a b)
      ( ( Contra-evid A a b)  $\phi$ )) x f =  $\phi$  x f
:=
concat
  ( Hom A x b)
  ( ( ( Contra-yon A is-pre- $\infty$ -category-A a b)
      ( ( Contra-evid A a b)  $\phi$ )) x f)
  (  $\phi$  x (Comp-is-pre- $\infty$ -category A is-pre- $\infty$ -category-A x a a f (Id-hom A a)))
  (  $\phi$  x f)
  ( Naturality-contravariant-fiberwise-representable-transformation
      A is-pre- $\infty$ -category-A a b x a f (Id-hom A a)  $\phi$ )
  ( ap
    ( Hom A x a)
    ( Hom A x b)
    ( Comp-is-pre- $\infty$ -category A is-pre- $\infty$ -category-A x a a
      f (Id-hom A a))
    ( f)
    (  $\phi$  x)
    ( Comp-id-is-pre- $\infty$ -category A is-pre- $\infty$ -category-A x a f))
```

By function extensionality, this completes the proof of the Yoneda lemma.



BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 61, Number 2, April 2024, Pages 257–270
<https://doi.org/10.1090/bull/1830>
Article electronically published on February 15, 2024

STRANGE NEW UNIVERSES: PROOF ASSISTANTS AND SYNTHETIC FOUNDATIONS

MICHAEL SHULMAN

ABSTRACT. Existing computer programs called proof assistants can verify the correctness of mathematical proofs but their specialized proof languages present a barrier to entry for many mathematicians. Large language models have the potential to lower this barrier, enabling mathematicians to interact with proof assistants in a more familiar vernacular. Among other advantages, this may allow mathematicians to explore radically new kinds of mathematics using an LLM-powered proof assistant to train their intuitions as well as ensure their arguments are correct. Existing proof assistants have already played this role for fields such as homotopy type theory.

“Out of nothing I have created a strange new universe.”

– János Bolyai, one of the inventors of non-Euclidean geometry

Strange new universes



“I believe that future mathematics software will combine the ease of interaction of an LLM with the near-absolute trustworthiness of a proof assistant.

...just as a present-day mathematician can exploit a computer’s calculational ability to explore numerical realms undreamed-of decades ago, these imagined future mathematicians may exploit a computer’s logical ability to explore conceptual realms undreamed-of today.

...What new kinds of mathematics, then, may be waiting for us to explore with the proof assistants of tomorrow?”

— Michael Shulman

Strange new universes of synthetic mathematics



I often explain **synthetic mathematics** as the project of developing and working with a domain specific language for a particular mathematical subdiscipline expressed using type theory. This is related to, but more powerful than, the idea of axiom systems and abstraction boundaries.

Active areas of synthetic mathematics include:

- synthetic geometry (aka Euclidean/absolute, geometry)
- synthetic differential geometry
- synthetic homotopy theory (aka homotopy type theory)
- synthetic category theory (aka directed/simplicial type theory)
- synthetic algebraic geometry
- synthetic topology (aka real cohesive HoTT, condensed type theory, synthetic stone duality)
- synthetic probability theory ...



To support a mathematical multiverse of synthetic mathematical universes, we need to answer a few questions:

- How quickly can we build experimental proof assistants to explore domain specific type theories still under development?
- For those synthetic frameworks that survive the prototype stage, how do we improve the user experience of formalizers?
- How can we make synthetic libraries speak to each other? What should we be trying to achieve with translations?
- Should we ask AI for help in any of the above or is this infrastructure development a task for human experts only?

Towards the future of type theory in mathematics



I contend that mathematics and mathematicians would benefit from greater understanding of dependent type theory.

The following might help bring this future about:

- Work towards building scientific and social ties with the logic community, e.g., by showing up at conferences and striking up conversations.
- Exposition directed at mathematics students: a natural language formalized reintroduction to proofs in dependent type theory?
- Exposition directed at mathematicians: e.g., Angiuli and Gratzer's "Principles of Dependent Type Theory" for mathematicians?
- Work to improve user experiences across the ecosystem of proof assistants.
- Possibly leverage funding/interest in AI for mathematics?

Tack själv!