HoTTLean: Formalizing the Meta-Theory of HoTT in Lean

Joseph Hua[1], Steve Awodey[1], Mario Carneiro[2],
Sina Hazratpour[3], Wojciech Nawrocki[1], Spencer Woolfson[1], and Yiming Xu[4]

[1] Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
{jlhua,awodey,wnawrock,swoolfso}@andrew.cmu.edu
[2] Chalmers University of Technology, Gothenburg, Sweden
marioc@chalmers.se
[3] Stockholm University, Stockholm, Sweden
sinahazratpour@gmail.com
[4] Ludwig Maximilian University of Munich, Munich, Germany
yiming.xu@lmu.de

Introduction. While elegant synthetic proofs such as those in *Homotopy Type Theory* (HoTT) are expected to "compile" to proofs of classical theorems when interpreted in suitable models, this idea has not yet been exploited in proof assistants. For example, Cubical Agda supports synthetic reasoning about cubical types, but its proofs have not been translated formally to facts about cubical sets, let alone their topological realizations. The HoTTLean project aims to bridge this gap by formalizing in Lean the semantics of a type theory we call HoTT0, a fragment of HoTT where univalence holds only on set-truncated types. We define the class of natural models [Awo18] of HoTT0, and prove that its syntax has a sound interpretation in any such model. As one concrete instance, we formalize the groupoid model of HoTT0 [HS98], providing a specific "compilation target" for synthetic proofs. Finally, we work toward embedding HoTT0 as a domain-specific language using Lean's extensible syntax and meta-programming facilities [UdM20]. Overall, this allows users to write synthetic proofs in HoTT0 and use the interpretation to produce constructions pertaining to groupoids as defined in Mathlib, the standard library for mathematics in Lean [Com20]. Through its compositional and modular approach, our project not only provides a bridge between synthetic and classical mathematics, but also lays the foundations for formalized semantics of other internal languages such as (complete) HoTT [Uni13] or directed type theory [Nor19].

Project structure. This project consists of the following components:

- **Type theory.** The type theory HoTT0 is a variant of that described in the HoTT book [Uni13]. HoTT0 has N Russell-style universes $U_1 : \cdots : U_N$. There can only be finitely many universes due to Gödelian constraints. HoTT0 has Σ-types, Π-types, and intensional identity types. We define subuniverses of set-truncated types internally in HoTT0, and require univalence axioms for each of these subuniverses. Though univalence implies functional extensionality in the subuniverses, to get functional extensionality for all types we postulate it as another axiom.

  In our Lean formalization, we present HoTT0 by an inductive type `Expr` of raw terms (quotient-inductive-inductive types [ACD+18] are not available to us) together with proof-irrelevant typing judgments `EqTp` and `EqTm` specifying type and term equality, respectively. We view these as partial equivalence relations, defining $\Gamma \vdash t : A \triangleq \Gamma \vdash t \equiv t : A$ and correspondingly for typehood. To reduce proof burden, we build presuppositions into some of the inference rules, as well as postulate rather than prove symmetry, transitivity, and closure under substitution. We establish only very basic syntactic metatheorems, preferring to work with the semantics to the maximal extent possible.

- **Interpretation.** Natural model semantics were developed by Awodey [Awo18], with additional simplifications in the definition of identity types suggested by Richard Garner. The construction makes use of polynomial functor machinery, which is currently being formalized in the parent project Poly, available at github.com/sinhp/Poly.

  We define the class of HoTT0 natural models, and show that HoTT0 syntax has a sound interpretation into any such model. Thanks to this compartmentalization, our project could be extended with other model constructions (such as simplicial sets) without altering the syntax or having to reprove soundness.

  In our Lean formalization, a model of HoTT0 is a sequence of `NaturalModelBase` structures connected by *universe morphisms* `UHom`, together with additional data to support type constructors. Each morphism provides the semantics for one universe. We construct an interpretation of HoTT0 into any `UHomSeq` as a partial function defined by recursion on raw terms. We then show that this function is total on well-formed types and terms, and respects judgmental equality.

- **Groupoid model.** In our Lean formalization, the category `Ctx` of contexts is the category of 1-groupoids `Grpd.{N+1,N+1}` with `Type (N+1)`-sized objects and morphisms (as defined in Mathlib). This supports a natural model structure with N universes, Σ, Π, and identity types. Semantically, univalent subuniverses of set-truncated types correspond to subcategories of discrete groupoids.

  The type classifier `Ty` is defined to be the presheaf on `Ctx` that takes a context Γ to the set of functors $\Gamma \to$ `Grpd.{N+1,N+1}`. Up to isomorphism, a type $\Gamma \to$ `Ctx` corresponds to an isofibration of groupoids, but this view does not provide a strict interpretation of the syntax (substitution equations only hold up to isomorphism). The interpretation of Σ-types is thus not simply defined using the composition of isofibrations. For each piece of syntax, we explicitly describe the action on the classifiers as natural transformations between presheaves.

  Terms are classified by the presheaf that takes a context Γ to the set of functors $\Gamma \to$ `PGrpd.{N+1,N+1}` where `PGrpd.{N+1,N+1}` is the category of *pointed* groupoids and functors preserving the point up to isomorphism. Context extension is defined using the Grothendieck construction `Grothendieck` from Mathlib.

- **Embedded proof assistant.** An aim of this project is to demonstrate that internal proofs in HoTT0 can be translated into proofs of theorems about Mathlib's groupoids, with the proof assistant offering automated translation. We are working on an embedded HoTT0 proof mode with support for seamless transitions between reasoning internally and reasoning about the groupoid model. Thanks to Lean's extensible syntax, macro system, and elaboration facilities [UdM20], we may reuse existing tactics and user interface elements in the HoTT0 proof mode, providing Lean users with a familiar interactive environment.

Formalization progress. HoTTLean is work in progress. The repository is accessible at sinhp.github.io/groupoid_model_in_lean4. So far we have constructed a fragment of the syntax with universes, Σ and Π-types and an interpretation of this syntax into its corresponding natural model semantics. For the groupoid model, we have constructed semantics for universes and Σ-types, with Π-types in the making.

Related work. We know of one other project that worked towards a formalization of the groupoid model of HoTT, by Sozeau and Tabareau [ST14]. Although their groupoid semantics follow the style of categories with families, they did not formalize categories with families as a class of models. Meanwhile, our project develops abstract natural model semantics, and then constructs the groupoid model as a particular instance. Another key difference is that their groupoids are enriched in setoids, while ours are not. Finally, a novelty of HoTTLean is in building a domain-specific proof mode that allows users to develop formal internal language arguments.

Ahrens, North, and van der Weide formalized the semantics of bicategorical type theory in Coq [ANvW23]. Unlike HoTTLean, they did not formalize the syntax or its interpretation, nor did they consider Σ and Π-types.

Maillard and Xu are constructing a deep embedding of geometric logic in Lean that they unfold into presheaf semantics in order to obtain theorems about Mathlib's algebraic structures [XM25].

The Flypitch project of Han and van Doorn established the independence of the continuum hypothesis in Lean [HvD20]. Mathlib now contains model and set theory libraries, and methods similar to ours could be employed to facilitate (often syntactically complex) internal arguments there.

Future work.

- HoTT0 can and should be extended to offer a class of higher inductive types. Possible candidates that are consistent with groupoid semantics include higher W-types [Vid18] and groupoid quotients [VvdW21]. From groupoid quotients one could construct the classifying space BG of a group G, as well as 0-truncations (used to define homotopy groups).

- Here we focus on semantic foundations, but one could use HoTT0 to develop univalent set-level mathematics. Univalence for sets can be used to implement the Structure Identity Principle, allowing for rigorous identification of isomorphic structures [Acz11].

- It may be possible to integrate our work with `Ground Zero` (github.com/forked-from-1kasper/ground_zero), a recent HoTT library using Lean 4.

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0009.

References

[ACD+18] Thorsten Altenkirch, Paolo Capriotti, Gabe Dijkstra, Nicolai Kraus, and Fredrik Nordvall Forsberg. *Quotient Inductive-Inductive Types*, page 293–310. Springer International Publishing, 2018.

[Acz11] Peter Aczel. On Voevodsky's univalence axiom. *Mathematical Logic: Proof Theory, Constructive Mathematics, Samuel R. Buss, Ulrich Kohlenbach, and Michael Rathjen (Eds.). Mathematisches Forschungsinstitut Oberwolfach, Oberwolfach*, page 2967, 2011.

[ANvW23] Benedikt Ahrens, Paige Randall North, and Niels van der Weide. Bicategorical type theory: semantics and syntax. *Mathematical Structures in Computer Science*, 33(10):868–912, 2023.

[Awo18] Steve Awodey. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28(2):241–286, 2018.

[Com20] The Mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2020, page 367–381, New York, NY, USA, 2020. Association for Computing Machinery.

[HS98] Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36 of *Oxford Logic Guides*, pages 83–111. Oxford Univ. Press, New York, 1998.

[HvD20] Jesse Michael Han and Floris van Doorn. A formal proof of the independence of the continuum hypothesis. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020*, 2020.

[Nor19] Paige Randall North. Towards a directed homotopy type theory. *Electronic Notes in Theoretical Computer Science*, 347:223–239, 2019.

[ST14] Matthieu Sozeau and Nicolas Tabareau. Towards an internalization of the groupoid model of type theory. *Types for Proofs and Programs 20th Meething (TYPES 2014), Book of Abstracts*, 2014.

[UdM20] Sebastian Ullrich and Leonardo de Moura. Beyond notations: Hygienic macro expansion for theorem proving languages. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning*, pages 167–182, Cham, 2020. Springer International Publishing.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

[Vid18] Jakob Vidmar. *Polynomial functors and W-types for groupoids*. PhD thesis, University of Leeds, 2018.

[VvdW21] Niccolò Veltri and Niels van der Weide. Constructing higher inductive types as groupoid quotients. *Logical Methods in Computer Science*, Volume 17, Issue 2, April 2021.

[XM25] Yiming Xu and Kenji Maillard. Geometric reasoning in Lean: from algebraic structures to presheaves. In *31st International Conference on Types for Proofs and Programs*, 2025.