HoTTLean: Semantics of HoTT in Lean

by S. Awodey, M. Carneiro, S. Hazratpour, J. Hua, W. Nawrocki, S. Rong, S. Woolfson, Y. Xu

Blueprint (PDF) Blueprint (web) Documentation GitHub

HoTTLean is maintained by the HoTTLean team. Visit the GitHub repository for more information.

Jlh18 feat: links to papers ✕ 31133dd · 2 days ago 429 Commits

.Notes (outdated)	remove notes	2 years ago
.github/workflows	refactor: Unstructured (#152)	4 months ago
.vscode	Add standard vscode settings of lean projects	2 years ago
HoTTLean	chore: rename Path.Id to PathType.Path (#169)	2 months ago
attic	refactor: Unstructured (#152)	4 months ago
bench	Reorganize files and update documentation (#141)	5 months ago
blueprint/src	Reorganize files and update documentation (#141)	5 months ago
home_page	Reorganize files and update documentation (#141)	5 months ago
scripts	Create update_mathlib.sh	2 years ago
test	fix: isProp, isSet	last month
.gitignore	fix: blueprint build	5 months ago
HoTTLean.lean	fix: typechecker	4 months ago
LICENSE	Add Apache License 2.0	2 months ago
README.md	feat: links to papers	2 days ago
flake.lock	chore: remove field (#160)	4 months ago
flake.nix	feat: benchmarks, optimizations, unitt demo (#137)	5 months ago
lake-manifest.json	chore: bump to Lean v4.25.0-rc2 (#153)	4 months ago
lakefile.lean	chore: bump to Lean v4.25.0-rc2 (#153)	4 months ago
lean-toolchain	chore: bump to Lean v4.25.0-rc2 (#153)	4 months ago
test.lean	feat: add axioms for HoTT0 (#134)	5 months ago

README | Apache-2.0 license

HoTTLean: Semantics of HoTT in Lean

HoTTLean is a Lean formalization of the groupoid model of homotopy type theory, as well as of category-theoretic models of Martin-Löf type theories in general, together with a proof mode for developing mathematics synthetically in those type theories. HoTTLean is work-in-progress.

It currently contains the following components:

- A deep embedding of MLTT syntax with Π, Σ, Id types, and base constants (`HoTTLean.Syntax`).
- **SynthLean**, a proof assistant for these deeply embedded theories (`HoTTLean.Frontend`). It includes a certifying typechecker (`HoTTLean.Typechecker`).
- Natural model semantics of MLTT in presheaf categories (`HoTTLean.Model`) and a proof that this semantics is sound for the syntax (`ofType_ofTerm_sound`).
- A `sorry` -free construction of the groupoid model of type theory with Π, Σ, Id types (`HoTTLean.Groupoids`).
- Pieces of general mathematics and category theory such as the Grothendieck construction for groupoids (`HoTTLean.ForMathlib` , `HoTTLean.Grothendieck` , `HoTTLean.Pointed`). These are being upstreamed to Mathlib.

Documentation of the Lean code can be found here. (We also have a blueprint, but it is very outdated.) Relevant papers to this project are listed below

- A Certifying Proof Assistant for Synthetic Mathematics in Lean
- Path Types in Algebraic Type Theory
- Polynomial Functors in π-clans for the Semantics of Type Theory

Dependencies

We rely on Mathlib and Poly, a formalization of polynomial functors.

How can I try out HoTTLean?

First make sure that Lean is installed. Clone this repository, then in a terminal execute

```
# Fetch pre-built Mathlib cache
lake exe cache get
# Build the project
lake build
# Run tests (optional)
lake test
```

Examples demonstrating the usage of SynthLean (for internal reasoning) and of our model definitions (for external reasoning) can be found under `test/` :

- `test/unitt` specifies a type theory with a unit type.
- `test/hott0` defines HoTT0, a version of HoTT where univalence only holds for h-sets.

Can I contribute to HoTTLean?

Contributions are very welcome! We use GitHub issues to organize the project and track formalization progress. New contributors should pay attention to issues marked with `O-help` (help wanted) and `D-low` (low difficulty). You can also reach out to us on the Lean Zulip.

How does this differ from GroundZero?

GroundZero is a purely synthetic development of HoTT, whereas HoTTLean focuses on showing that synthetic results in HoTT can be interpreted as theorems about classical models that use definitions from Mathlib. A target for HoTTLean is to eventually allow interpreting theorems of GroundZero (as well as other HoTT libraries) in such a way. This is not quite feasible *yet* because HoTTLean does not currently support (higher) inductive types, and our model construction is not yet complete.

About

sinhp.github.io/HoTTLean/

Readme
Apache-2.0 license
Activity
53 stars
9 watching
11 forks
Report repository

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 12

Languages

Lean 79.8% · TeX 14.0% · Jupyter Notebook 4.8% · CSS 0.9% · HTML 0.5% · Ruby 0.0%

HoTT Lean

Semantics of HoTT in Lean

HoTT Lean

~~Semantics of HoTT in Lean~~

A proof assistant for HoTT based on Lean, with integrated translation of synthetic proofs into analytic ones.

1) What is HoTT?

See my CST talk from 28.1.26!

HoTT is MLTT with:

- $\Pi, \Sigma, \mathrm{Id}, \mathcal{U}_0, \mathcal{U}_1, \dots$
- Inductive types $\mathbb{N}$, $A+B$, $W_A B$, ...
- HITs: $X/\sim$, S^n, $\|X\|_n$, ...
- Univalence:

$$(A = B) \simeq (A \simeq B).$$

1) What is HoTT?

See my CST talk from 28.1.26!

HoTT is MLTT with:

standard MLTT
- $\Pi, \Sigma, \mathrm{Id}, \mathcal{U}_0, \mathcal{U}_1, \ldots$
- Inductive types $\mathbb{N}$, $A+B$, $W_A B$, ...

Motivated by
- HITs: $X/\sim$, S^n, $\|X\|_n$, ...
- Univalence:

$$(A = B) \simeq (A \simeq B) \quad .$$

HoTT admits an interpretation into abstract homotopy theory & ∞-Topos theory.

1) What is HoTT?

See my CST talk from 28.1.26!

HoTT is MLTT with:

Standard MLTT
- $\Pi, \Sigma, \mathrm{Id}, \mathcal{U}_0, \mathcal{U}_1, \ldots$
- Inductive types $\mathbb{N}$, $A+B$, $W_A B$, ...

Motivated by
- HITs: $X/_\sim$, S^n, $\|X\|_n$, ...
- Univalence:

$$(A = B) \simeq (A \simeq B).$$

HoTT admits an interpretation into abstract homotopy theory & ∞-Topos theory.

permits

HoTT can be used as a DSL for "higher maths" to formalize results in proof assistants based on MLTT like Roq, Agda, Lean.

2) Previous implementations of HoTT

- Voevodsky's UniMath Library in Coq
  Univalence Axiom, type : type, no HITs

- Subsequent / parallel libraries :
  Coq/Rocq, Agda, Agda-UniMath

- Cubical Agda : Computes w/UA,
  synthetic homotopy theory

- Lean 2, 3, 4 : each more
  challenging than the last!

3) Why do HoTT in Lean?

- Provide access to higher math via synthetic methods: use Lean to formalize results in Homotopy theory directly, w/o going via sSets, etc.

- Provide univalence in Lean, e.g. for algebraic structures via the Structure Identity Principle.

- Use Lean's infrastructure & tools in HoTT formalizations.

- Integrate results from MathLib into HoTT formalizations.

- Conversely, use HoTT to produce "classical" results that can go into MathLib.

4) Difficulties of using "Cold Lean" to do HoTT.

– Main issue is Lean's treatment of equality $x = y$:

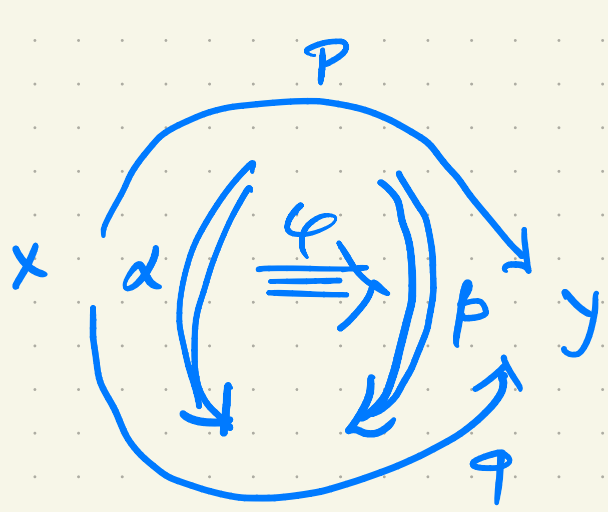
- Equality reflection: $$\frac{x = y}{x \equiv y}$$
  Ext. MLTT

- Proof irrelevance: $$\frac{p, q : x = y}{p = q}$$
  Coq

- Definitional Proof irrelevance: $$\frac{p, q : x = y}{p \equiv q}$$
  Lean

- Intensional MLTT: $$\frac{p, q : x = y}{? : p = q}$$
  Agda

- HoTT: $$\cfrac{\cfrac{p, q : x = y}{\alpha, \beta : p = q}}{\varphi, \psi : \alpha = \beta}$$

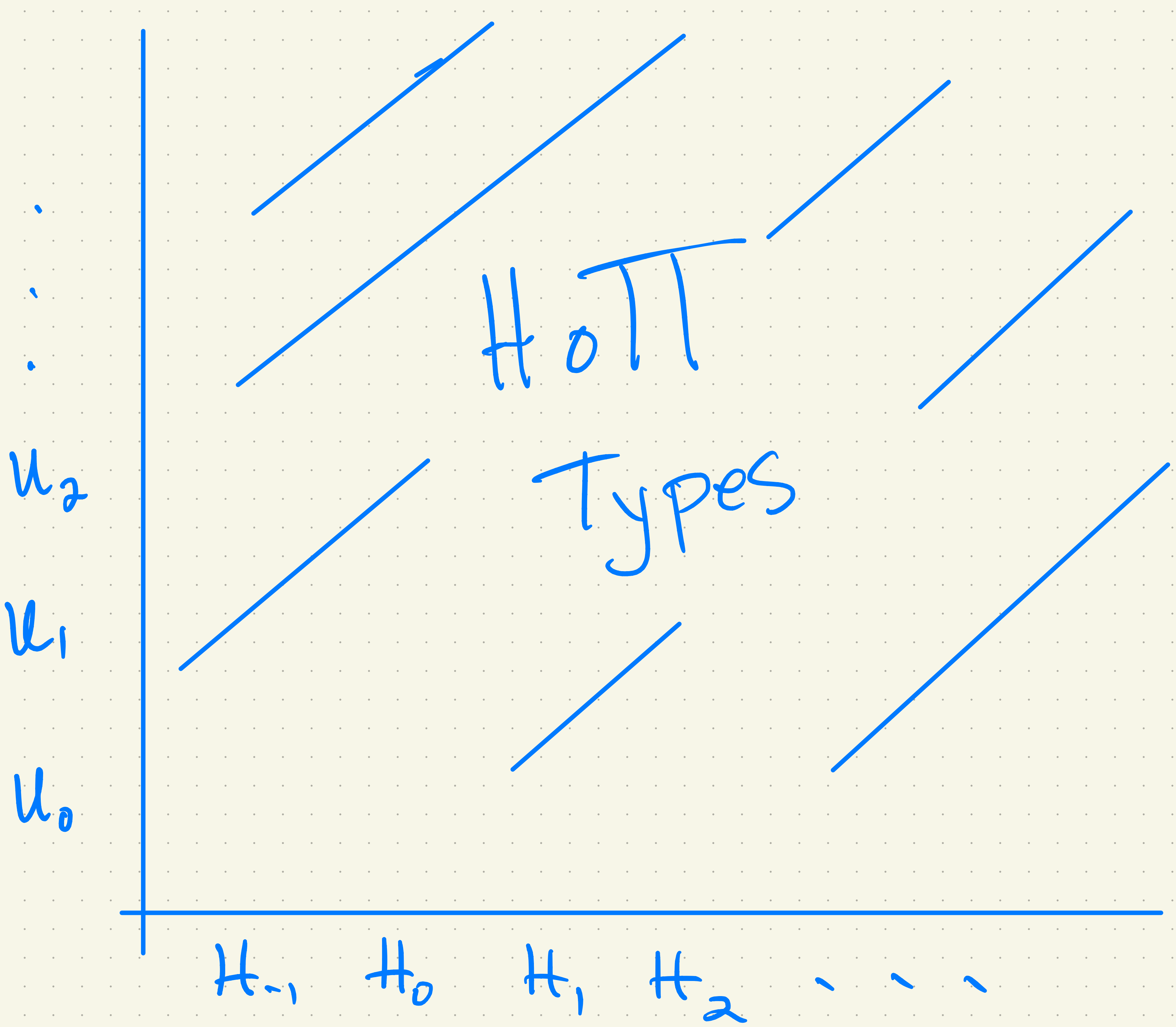


...

4) Difficulties of using "Cold Lean" to do HoTT.

- But also the classical logic built into Lean & MathLib is incompatible with univalence for higher types.

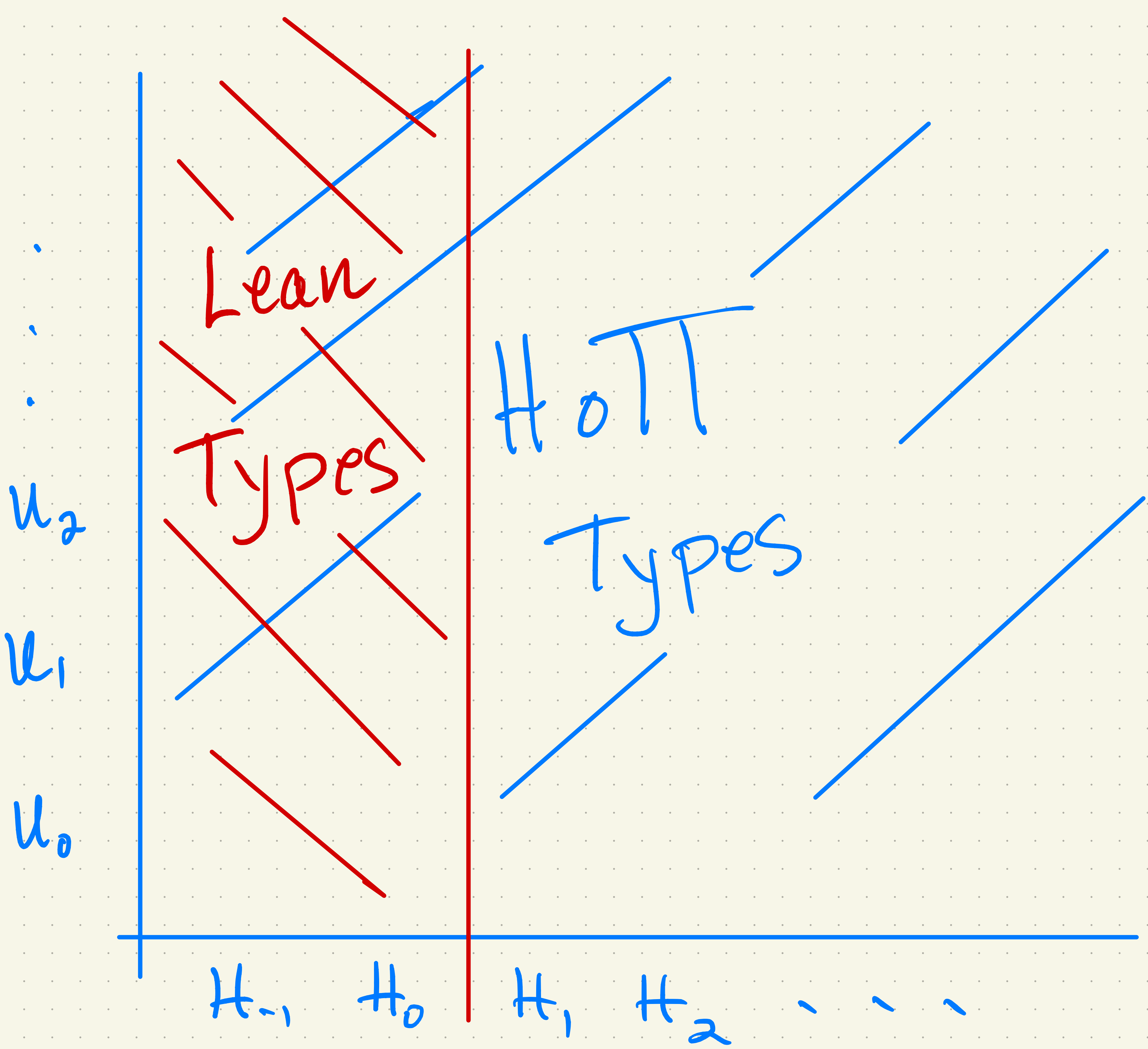
- No higher types means no synthetic homotopy theory.



4) Difficulties of using "Cold Lean" to do HoTT.

- But also the classical logic built into Lean & MathLib is incompatible with univalence for higher types.

- No higher types means no synthetic homotopy theory.



5) The HoTT Lean Approach

(i) Formalize a model of HoTT in Lean:
the Hofmann-Streicher Groupoid Model

(ii) Translate HoTT types & terms into it & use Lean to reason about the translation.

(iii) Deep embedding of HoTT into Lean.

(iv) Uses HoTT as a Domain Specific Language for the Groupoid model.

(v) HoTT proofs can use synthetic methods.

(iv) Their translations into the model are analytic results that can be integrated into MathLib.

6) Overview of HoTT Lean

User sees

Synth Lean

interactive proof assistant for synthetic math

checks proofs

Type theory in Lean

HoTT0

fragment of HoTT w/ univalent $U_0 = \mathrm{Set}$

interpreted

Axiomatic Category theory

Abstract Semantics

using Natural Models, Polynomial functors, ...

instantiated

Concrete Algebraic Structure

Groupoid Model of HoTT0

7) Synth Lean

Wojciech's talk
next week!

8) HoTT0 Type Theory

- MLTT w/ finitely many universes

$$\mathcal{U}_0 : \mathcal{U}_1 : \ldots : \mathcal{U}_n \qquad \text{(not cumulative)}$$

- Each $\mathcal{U}_i$ has Σ, Π, Id (for now...)

- Axioms :

  - FunExt (globally)
  - UA for each subuniverse

$$\mathrm{Sets}_0 \subseteq \mathcal{U}_0$$
$$\mathrm{Sets}_1 \subseteq \mathcal{U}_1$$
$$\ldots$$
$$\mathrm{Sets}_n \subseteq \mathcal{U}_n$$

- Implies the Structure Identity Principle for all Set-based algebraic structures.
  E.g. for Groups G, H :

$$G \cong H \Rightarrow G = H .$$

9) Natural Model Semantics

- Uses presheaf categories
  $[Ctx^{op}, Set]$
- with natural transformations
  $tp : Tm \longrightarrow Ty$
- determining polynomial functors
  $$P_{tp}(X) = \sum_{A:Ty} X^A$$
- to axiomatize the semantics of types & terms:

$$\text{"}\Gamma \vdash a : A\text{"} \quad \rightsquigarrow \quad \begin{tikzcd} & Tm \arrow[d, "tp"] \\ y\Gamma \arrow[ur, dashed, "a"] \arrow[r, "A"'] & Ty \end{tikzcd}$$

- and the type formers Σ, Π, Id:

$$\begin{tikzcd} P_{tp}Tm \arrow[r, "\lambda"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & Tm \arrow[d] \\ P_{tp}Ty \arrow[r, "\Pi"'] & Ty \end{tikzcd}$$

10) The Groupoid Model of HoTT

- Category of "contexts & substitutions":

  $\mathcal{C}tx$ $\quad h : G \longrightarrow H \quad \mathrm{Gpd}$

- Universal family of types:

$$\begin{array}{ccc} \mathcal{T}m & = & \mathrm{Gpd}_* \\ \downarrow & & \downarrow \\ \mathcal{T}y & = & \mathrm{Gpd} \end{array}$$

gpd of pt.d gpds

gpd of gpds

10) The Groupoid Model of HoTT

- Category of "Contexts & Substitutions":

  Ctx $\quad h : G \longrightarrow H \qquad$ ~~Gpd~~ GPD

- Universal family of types:

$$\begin{array}{ccc} Tm & = Gpd_* & \text{gpd of small pt.d gpds} \\ \downarrow & \downarrow & \\ Ty & = Gpd & \text{gpd of small gpds} \end{array}$$

- Dependent types & terms:

$\Gamma \vdash A$ type $\qquad \Gamma \xrightarrow{A} Gpd$

10) The Groupoid Model of HoTTO

- Category of "Contexts & Substitutions":

Ctx $\qquad h : G \longrightarrow H \qquad \cancel{\mathrm{Gpd}}$ GPD

- Universal family of types:

$$\begin{tikzcd} \mathrm{Tm} = \mathrm{Gpd}_* \arrow[d] \\ \mathrm{Ty} = \mathrm{Gpd} \end{tikzcd}$$

$\mathrm{Tm} = \mathrm{Gpd}_*$ gpd of small pt.d gpds

$\mathrm{Ty} = \mathrm{Gpd}$ gpd of small gpds

- Dependent types & terms:

$\Gamma \vdash a : A$ type

$$\begin{tikzcd} & \mathrm{Gpd}_* \arrow[d] \\ \Gamma \arrow[ur, dashed, "a"] \arrow[r, "A"'] & \mathrm{Gpd} \end{tikzcd}$$

- Substitution $\sigma : \Delta \longrightarrow \Gamma$:

$\Delta \vdash a[\sigma] : A[\sigma]$

$$\begin{tikzcd} & & \mathrm{Gpd}_* \arrow[d] \\ \Delta \arrow[r, "\sigma"'] \arrow[urr, bend left, "{a[\sigma]}"] \arrow[rr, bend right, "{A[\sigma]}"'] & \Gamma \arrow[ur, "a"] \arrow[r, "A"'] & \mathrm{Gpd} \end{tikzcd}$$

10) The Groupoid Model of HoTT

- Category of "Contexts & Substitutions":

  Ctx $h : G \longrightarrow H$ ~~Gpd~~ GPD

- Universal family of types:

$$\begin{array}{ccc} Tm & = & Gpd_* \\ \downarrow & & \downarrow \\ Ty & = & Gpd \end{array}$$

gpd of small pt.d gpds

gpd of small gpds

- Dependent types & terms:

$\Gamma \vdash a : A$ type

$$\begin{tikzcd} & Gpd_* \arrow[d] \\ \Gamma \arrow[ur, dashed, "a"] \arrow[r, "A"'] & Gpd \end{tikzcd}$$

- Substitution $\sigma : \Delta \longrightarrow \Gamma$:

$\Delta \vdash a[\sigma] : A[\sigma]$

$$\begin{tikzcd} & & Gpd_* \arrow[d] \\ \Delta \arrow[r, "\sigma"'] \arrow[urr, bend left, "a[\sigma]"] \arrow[rr, bend right, "A[\sigma]"'] & \Gamma \arrow[ur, "a"] \arrow[r, "A"'] & Gpd \end{tikzcd}$$

N.B.:

$A[\sigma][\tau] \equiv A[\sigma[\tau]]$

etc.

10) The Groupoid Model of HoTT

- Context extension: $$\frac{\Gamma \vdash A \text{ type}}{\Gamma.A \to \Gamma}$$

  is modeled by pullback:

$$\begin{tikzcd}
\Gamma.A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Gpd}_* \arrow[d] \\
\Gamma \arrow[r, "A"'] & \mathrm{Gpd}
\end{tikzcd}$$

10) The Groupoid Model of HoTTO

- <u>Context extension:</u>

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma.A \to \Gamma}$$

is modeled by pullback:

$$\begin{tikzcd} \int_\Gamma A = \Gamma.A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & Gpd_* \arrow[d] \\ \Gamma \arrow[r, "A"'] & Gpd \end{tikzcd}$$

Grothendieck construction

- <u>Type formers</u> Σ, Π :

$$\begin{tikzcd} \Gamma.A.B \arrow[d] \\ \Gamma.A \arrow[d] \\ \Gamma \end{tikzcd}$$

10) The Groupoid Model of HoTTO

- Context extension:
$$\frac{\Gamma \vdash A \text{ type}}{\Gamma.A \to \Gamma}$$

is modeled by pullback:

$$\begin{tikzcd} \int_\Gamma A = \Gamma.A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Gpd}_* \arrow[d] \\ \Gamma \arrow[r, "A"'] & \mathrm{Gpd} \end{tikzcd}$$

Grothendieck construction

- Type formers Σ, Π:

$$\frac{\Gamma.A \vdash B}{\Gamma \vdash \Sigma_A B}$$

$$\frac{\Gamma.A \vdash B}{\Gamma \vdash \Pi_A B}$$

$$\begin{tikzcd} & \Gamma.A.B \arrow[d] & \\ \Gamma.\Sigma_A B \arrow[dr] & \Gamma.A \arrow[d] & \Gamma.\Pi_A B \arrow[dl] \\ & \Gamma & \end{tikzcd}$$

are modeled as adjoints to pullback along $\Gamma.A \to \Gamma$.

10) The Groupoid Model of HoTT

- <u>Context extension:</u> $\dfrac{\Gamma \vdash A \text{ type}}{\Gamma . A \to \Gamma}$

is modeled by pullback:

$$\int_\Gamma A = \begin{tikzcd} \Gamma . A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Gpd}_* \arrow[d] \\ \Gamma \arrow[r, "A"'] & \mathrm{Gpd} \end{tikzcd}$$

Grothendieck construction

- <u>Type formers</u> Σ, Π :

$$\frac{\Gamma . A \vdash B}{\Gamma \vdash \Sigma_A B}$$

$$\frac{\Gamma . A \vdash B}{\Gamma \vdash \Pi_A B}$$

$$\begin{tikzcd} & \Gamma . A . B \arrow[d] & \\ \Gamma . \Sigma_A B \arrow[dr] & \Gamma . A \arrow[d] & \Gamma . \Pi_A B \arrow[dl] \\ & \Gamma & \end{tikzcd}$$

are modeled as <u>adjoints</u> to pullback along $\Gamma . A \to \Gamma$.

- Uses the <u>Poly Project</u> in MathLib!

10) The Groupoid Model of HoTTO

- Identity type:

$$\frac{\Gamma \vdash A}{\Gamma, x:A, y:A \vdash \mathrm{Id}_A(x,y)}$$

is modeled by the arrow groupoid

$$\begin{tikzcd} A^{\downarrow} \arrow[d] & \\ \Gamma.A.A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \Gamma.A \arrow[d] \\ \Gamma.A \arrow[r] & \Gamma \end{tikzcd}$$

- So for the Id type of $\mathrm{Sets} \leq \mathcal{U}$,

$$\sum_{A,B:\mathrm{Sets}} \mathrm{Id}_{\mathcal{U}}(A,B) = \mathrm{Sets}^{\downarrow} = \sum_{A,B:\mathrm{Sets}} \mathrm{Iso}(A,B),$$

which is univalence:

$$(A = B) \simeq (A \cong B).$$

11) What can HoTTLean do?

- Some synthetic topology is w/in reach, even though we only have 0 & 1-types:

  - Classifying spaces for groups BG & the theory of torsors & bundles,
  - Covering spaces & fundamental groups,
  - Synthetic group theory as in the Symmetry Book of Buchholtz et al.

- SIP for algebraic structures permits some univalent maths, simplifying algebra, category theory, sheaf theory, ...

- All of MathLib's "classical" set-level maths lives inside HoTT0's universes of sets,

Lean $<$ HoTT0 .

11) What can HoTTLean do?

- Finally, the HoTTLean approach is <u>modular</u>, so we can <u>extend HoTT0</u> with higher levels of <u>n-types</u>, and <u>extend the model</u> in groupoids to <u>n-groupoids</u>, …, <u>∞-groupoids</u>:

$$\text{Lean} \subset \text{HoTT0} \subset \text{HoTT1} \subset \ldots \subset \text{HoTT}.$$

11) What can HoTTLean do?

- Finally, the HoTTLean approach is modular, so we can extend HoTT0 with higher levels of n-types, and extend the model in groupoids to n-groupoids, ..., ∞-groupoids:

$$\text{Lean} \subset \text{HoTT0} \subset \text{HoTT1} \subset \ldots \subset \text{HoTT}.$$

THANKS!

Next week: SynthLean