A Certifying Proof Assistant for Synthetic Mathematics in Lean

Wojciech Nawrocki[1] with Joseph Hua[1], Mario Carneiro[2], Yiming Xu[3], Spencer Woolfson[4], Shuge Rong[1], Sina Hazratpour[5], and Steve Awodey[1]

[1] Carnegie Mellon University [2] Chalmers University of Technology
[3] LMU Munich [4] Chapman University [5] Stockholm University

January 12th | CPP 2026 | Rennes, France

This material is based upon work supported by the Air Force Office of Scientific Research under MURI award FA9550-21-1-0009, the National Science Foundation under grant number 2434614, and the European Union (ERC, Nekoka, 101083038).

Synthetic vs. analytic mathematics

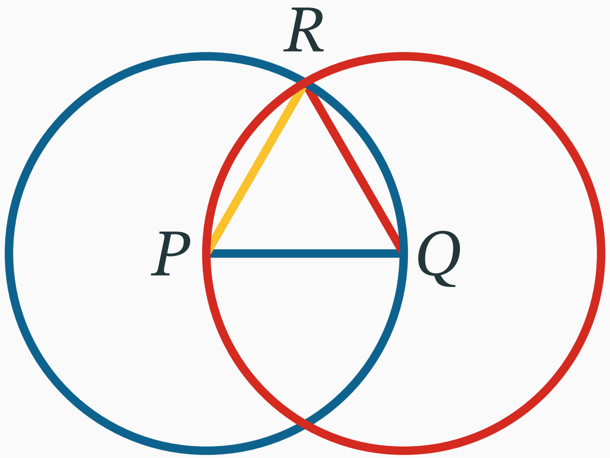
Axiomatize an area of inquiry in terms of irreducible primitives.

vs.

Break primitives **apart** in terms of finer entities.

Synthetic Euclidean geom. (theory)

Analytic Cartesian geom. (model)



$$P = \begin{pmatrix} x_P \\ y_P \end{pmatrix} \qquad Q = \begin{pmatrix} x_Q \\ y_Q \end{pmatrix}$$

$$R = P + \begin{pmatrix} \cos 60° & -\sin 60° \\ \sin 60° & \cos 60° \end{pmatrix} (Q - P)$$

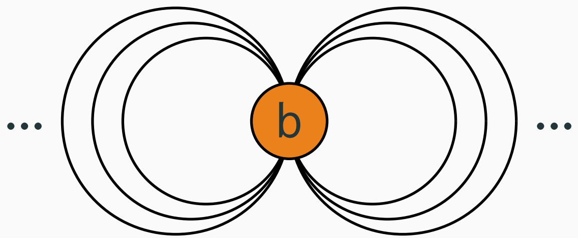
Synthetic homotopy theory

Modern synthetic mathematics uses dependent types and categories.

Homotopy type theory (HoTT) is an axiomatization of homotopy theory.
Types can be interpreted as (∞-)groupoids (i.e., categories with iso.s only).

```
data Bool : Type where
  true false : Bool
```


```
data S¹ : Type where
  base : S¹
  loop : base ≡ base
```



$\cong (\mathbb{Z}, +)$

Synthetic algebraic geometry

Corollary 3.9. *Let X be a scheme. If X is reduced, the ring $\mathcal{O}_X$ is a field from the internal point of view, in the sense that*

$$\mathrm{Sh}(X) \models \forall s : \mathcal{O}_X.\ \neg(\ulcorner s \text{ invertible} \urcorner) \Rightarrow s = 0.$$

Conversely, if $\mathcal{O}_X$ is a field in this internal sense, then X is reduced.

Blechschmidt [2] freely mixes theory-based and model-based reasoning.

Our goal: formally verified theory-model connection.

Our contributions

1. Deep embedding of Martin-Löf type theories with U, Π, Σ, Id types, and base constants, e.g. HoTT (without inductive types).
2. Natural model semantics (cf. CwFs) and their soundness for the syntax.
3. SynthLean, an embedded proof assistant for 1. and 2.

Usage of SynthLean

```
declare_theory pointt

pointt axiom X : Type
pointt axiom p : X

pointt #print p
-- axiom p : X

#print p
-- def p : CheckedAx [X] :=
--   { tp := .el X.val,
--     wf_tp := (… : [X] | [] ⊢[0] tp),
--     … }
```

```
def 𝓜 : Universe := …

def 𝓜_X : 𝟏_ _ ⟶ 𝓜.Ty := …
def 𝓜_p : 𝟏_ _ ⟶ 𝓜.Tm := …

def I : Interpretation 𝓜 where
  ax := fun
    | ``X => 𝓜_X
    | ``p => 𝓜_p

pointt def q : X := p
example : I.interpDef q = 𝓜_p
```

Disclaimer: the API is oversimplified here.

Syntax

MLTTs stratified by universe level

$\boxed{\mathbb{T} \vdash \Gamma}$ $\boxed{\mathbb{T} \mid \Gamma \vdash_{\ell} A}$ $\boxed{\mathbb{T} \mid \Gamma \vdash_{\ell} t : A}$

$\boxed{\mathbb{T} \mid \Gamma \vdash_{\ell} A \equiv B}$ $\boxed{\mathbb{T} \mid \Gamma \vdash_{\ell} t \equiv u : A}$

$$\frac{\ell < \ell_{\mathrm{max}}}{\mathbb{T} \mid \Gamma \vdash_{\ell+1} U_{\ell}} \qquad \frac{\mathbb{T} \mid \Gamma \vdash_{\ell+1} a : U_{\ell}}{\mathbb{T} \mid \Gamma \vdash_{\ell} \mathsf{El}\, a}$$

$$\frac{\mathbb{T} \mid \Gamma \vdash_{\ell} A \qquad \mathbb{T} \mid \Gamma.A_{\ell} \vdash_{\ell'} B}{\mathbb{T} \mid \Gamma \vdash_{\mathrm{max}(\ell,\ell')} \Pi_{\ell,\ell'} A.\, B} \qquad \frac{\cdots \qquad (c :_{\ell} T) \in \mathbb{T}}{\mathbb{T} \mid \Gamma \vdash_{\ell} c : T}$$

MLTTs stratified by universe level

$\boxed{\mathbb{T} \vdash \Gamma}$ $\boxed{\mathbb{T} \mid \Gamma \vdash_{\ell} A}$ $\boxed{\mathbb{T} \mid \Gamma \vdash_{\ell} t : A}$

$\boxed{\mathbb{T} \mid \Gamma \vdash_{\ell} A \equiv B}$ $\boxed{\mathbb{T} \mid \Gamma \vdash_{\ell} t \equiv u : A}$

```
mutual
inductive WfCtx : Theory χ → Ctx χ → Prop
inductive WfTp : Theory χ → Ctx χ → Nat → Expr χ → Prop
...
```

Basic syntactic metatheory

Theorem (Admissible substitution). If $\Gamma \vdash_{\ell} \mathcal{J}$ and $\Delta \vdash \sigma : \Gamma$, then $\Delta \vdash_{\ell} \mathcal{J}[\sigma]$.

Theorem (Inversion for $\equiv$). If $\Gamma \vdash_{\ell} A \equiv B$, then $\Gamma \vdash_{\ell} A$ and $\Gamma \vdash_{\ell} B$.

Axiom (Injectivity of Π types). If $\Gamma \vdash_{\cancel{k}} \Pi_{\ell_0,\ell_0'} A.\, B \equiv \Pi_{\ell_1,\ell_1'} A'.\, B'$,
then $\Gamma \vdash_{\ell_0} A \equiv A'$ and $\Gamma.A \vdash_{\ell_0'} B \equiv B'$.

Semantics

A **natural model universe** in $[\mathbf{Ctx}^{\mathrm{op}}, \mathbf{Set}]$

Let $\Gamma \vdash_0 A$ be a type, $\vdash \Gamma.A$ the extended context, and $\Gamma \vdash_0 t : A$ a term.

$$\begin{tikzcd}
y[\![\Gamma.A]\!] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathsf{Tm}_0 \arrow[d, "\mathsf{tp}_0"] \\
y[\![\Gamma]\!] \arrow[r, "[\![A]\!]"'] \arrow[u, bend left, "[\![t]\!]"] & \mathsf{Ty}_0
\end{tikzcd}$$

```
structure Universe where
  Tm : Psh Ctx
  Ty : Psh Ctx
  tp : Tm ⟶ Ty
  ext {Γ : Ctx} (A : y(Γ) ⟶ Ty) : Ctx
  ...
```

Soundness of interpretation

Following Streicher [7:Ch. III], define

$$[\![\Gamma]\!] : (\Gamma : \mathsf{List\ Expr}) \rightharpoonup \mathbf{Ctx}$$

$$[\![A]\!]_{X,\ell} : (A : \mathsf{Expr})\,(X : \mathbf{Ctx})\,(\ell : \mathbb{N}) \rightharpoonup [\mathbf{Ctx}^{\mathrm{op}}, \mathbf{Set}](yX, \mathsf{Ty}_\ell)$$

Theorem (Soundness).

If $\Gamma \vdash_\ell A$, then $[\![A]\!]_{[\![\Gamma]\!],\ell}\downarrow$

If $\Gamma \vdash_\ell A \equiv B$, then $[\![A]\!]_{[\![\Gamma]\!],\ell} = [\![B]\!]_{[\![\Gamma]\!],\ell}$

Previously De Boer/Brunerie/Lumsdaine/Mörtberg [3] in Agda.

Proof assistant

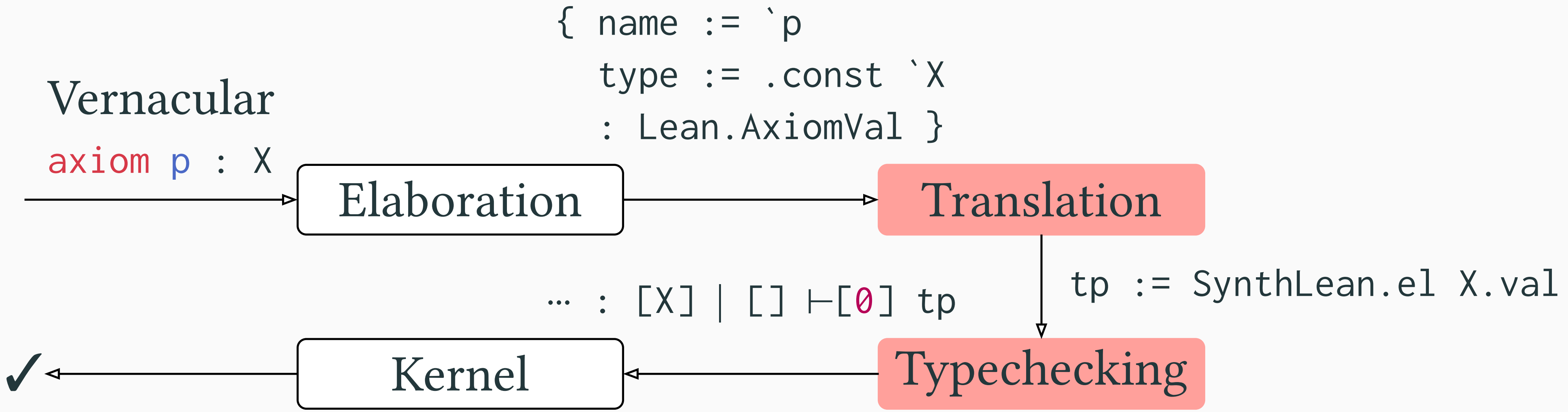
Certificate-producing proof assistant

```
declare_theory pointt

pointt axiom X : Type
pointt axiom p : X
```

SynthLean ⇝

```
def p : CheckedAx [X] :=
  { tp := SynthLean.el X.val,
    wf_tp := (… : [X] | [] ⊢[0] tp),
    … }
```



Typechecking via normalization by evaluation (NbE)

```
partial def evalTp (env : Q(List NbE.Val)) (T : Q(SynthLean.Expr)) :
  Lean.MetaM (
    -- Return value:
    (vT : Q(NbE.Val)) ×
      -- Postcondition certificate:
      Q(∀ {Γ Δ σ l},
        (Γ ⊢[l] $T) → (Δ ⊩ $env ≈ σ : Γ) → (Δ ⊩[l] $vT ≈ $T.subst σ)))
```

🛑 No termination encoding (e.g. Bove-Capretta [4]) needed.

🏎️ Optimizations: $\mathcal{O}(0)$ weakening via De Bruijn levels, defunctionalized closures à la Abel [1], certificate subterm sharing.

Other aspects of the formalization

- ✍️ Dependent rewriting tactic `rw!`, j.w.w. Aaron Liu, added to mathlib.
- ⏳ Performance issues formalizing "obvious" naturality laws (see discussion).
- ⚡ Deeply embedded judgmental equalities without QIITs ~ setoid hell.
  But QIITs ~ quotient hell [5, 6]?
- 🪜 NbE certificates reminiscent of A. Kolomatskaia's Stepping by Evaluation.
- 🦜 Relying on library of polynomial functors. S. Hazratpour & E. Riehl. Poly.

github.com/sinhp/HoTTLean

Bibliography

[1] Andreas Abel. 2013. Normalization by evaluation: Dependent types and impredicativity. Doctoral dissertation.

[2] Ingo Blechschmidt. 2018. Using the Internal Language of Toposes in Algebraic Geometry. Doctoral dissertation.

[3] Menno de Boer. 2020. A Proof and Formalization of the Initiality Conjecture of Dependent Type Theory. Licentiate Thesis. Department of Mathematics, Stockholm University.

[4] Ana Bove and Venanzio Capretta. 2005. Modelling general recursion in type theory. Mathematical. Structures in Comp. Sci. 15, 4 (August 2005), 671–708. https://doi.org/10.1017/S0960129505004822

[5] Liang-Ting Chen, Fredrik Nordvall Forsberg, and Tzu-Chun Tsai. 2026. Can We Formalise Type Theory Intrinsically without Any Compromise? A Case Study in Cubical Agda. In Proceedings of the 15th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP '26), 2026. Association for Computing Machinery, Rennes, France, 201–215. https://doi.org/10.1145/3779031.3779090

[6] Ambrus Kaposi and Loïc Pujet. 2025. Type Theory in Type Theory using a Strictified Syntax. Proc. ACM Program. Lang. 9, ICFP (August 2025). https://doi.org/10.1145/3747535

[7] Thomas Streicher. 1991. Semantics of type theory: correctness, completeness, and independence results. Birkhäuser Boston Inc., USA.