Compiling HoTT with Lean: Progress Update

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

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

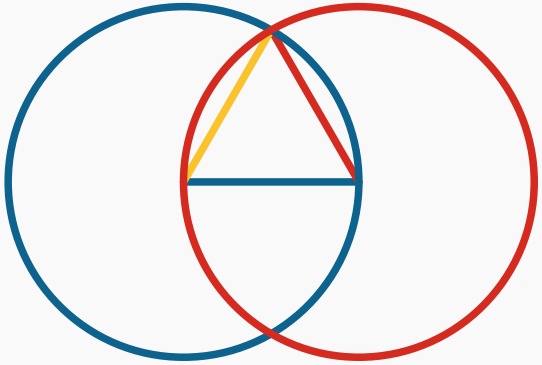
August 4th | MURI Meeting 2025

Slides at voidma.in/muri.pdf

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

Synthetic (axiomatic) mathematics

Axiomatize an area of inquiry by viewing its basic objects as irreducible primitives.



$$P \neq Q : \text{Point}$$
$$\exists (C_1 : \text{Circle}).\ \text{Center}(P, C_1) \wedge Q \in C_1$$
$$\exists (C_2 : \text{Circle}).\ \text{Center}(Q, C_2) \wedge P \in C_2$$
$$\text{Intersect}(C_1, C_2)$$

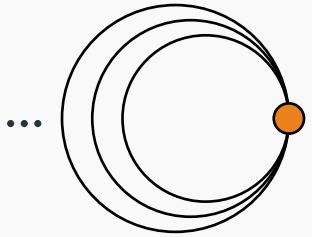
$$P \neq Q \in \mathbb{R}^2$$
$$C_1 = \left\{(x, y) \mid (x + \ldots)^2 + (y + \ldots)^2 = \ldots\right\}$$
$$C_2 = \left\{(x, y) \mid (x + \ldots)^2 + (y + \ldots)^2 = \ldots\right\}$$
$$\exists R \in C_1 \cap C_2$$

Synthetic abstract homotopy theory

Homotopy type theory (HoTT) is an axiomatization of abstract homotopy theory. In many interesting models, types are ∞-groupoids.

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

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



Our goal: formally verified theory-model connection.

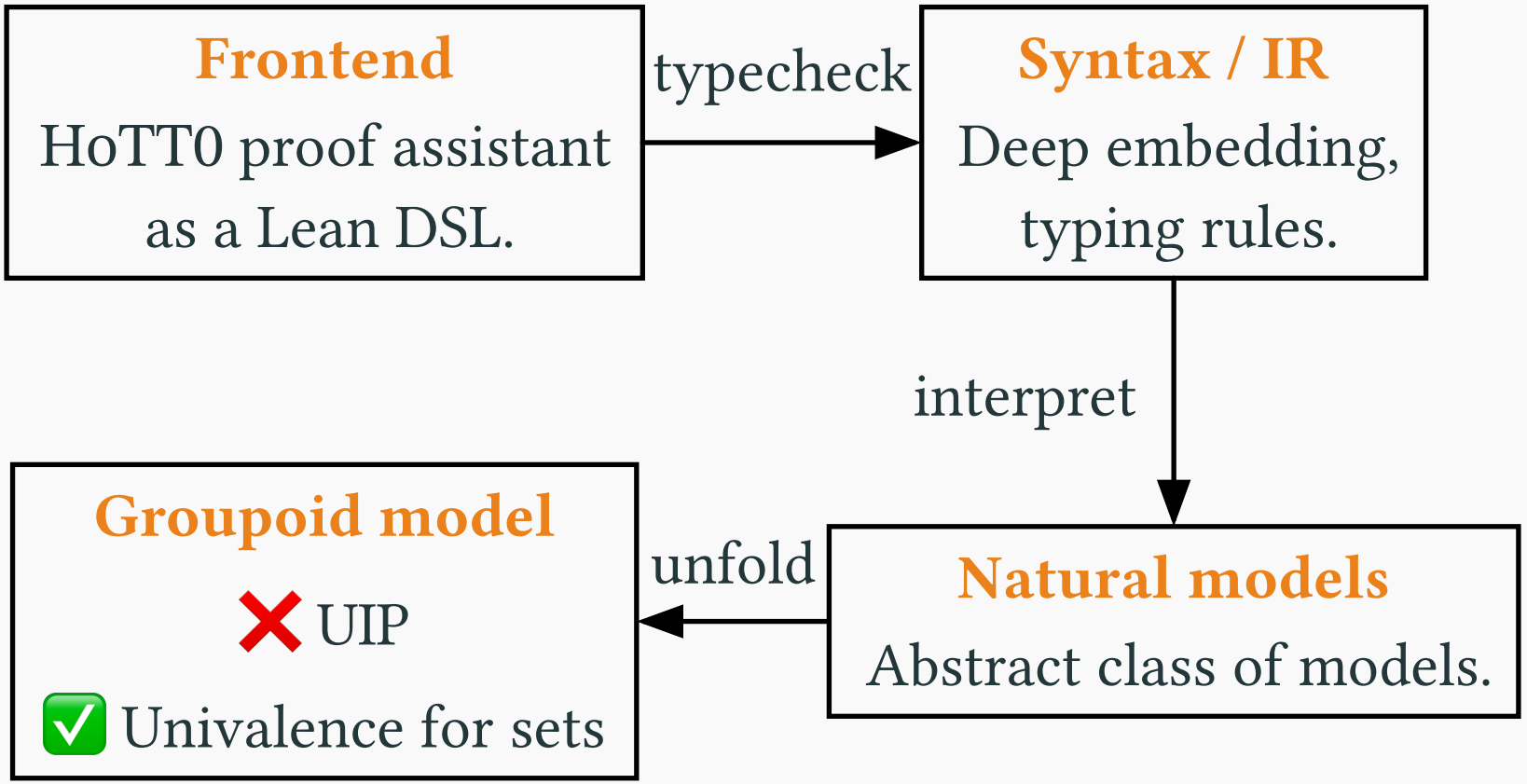
Corollary 3.9. *Let X be a scheme. If X is reduced, the ring $\mathcal{O}_X$ is a field from the internal point of view, in the sense that*

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

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

From Blechschmidt [1].

Compilation pipeline



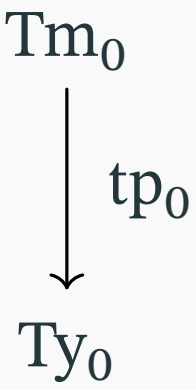
Natural models

Semantics: natural models in $[\mathcal{C}^{\mathrm{op}}, \mathrm{Set}]$

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

Semantics: natural models in $[\mathcal{C}^{\mathrm{op}}, \mathrm{Set}]$

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



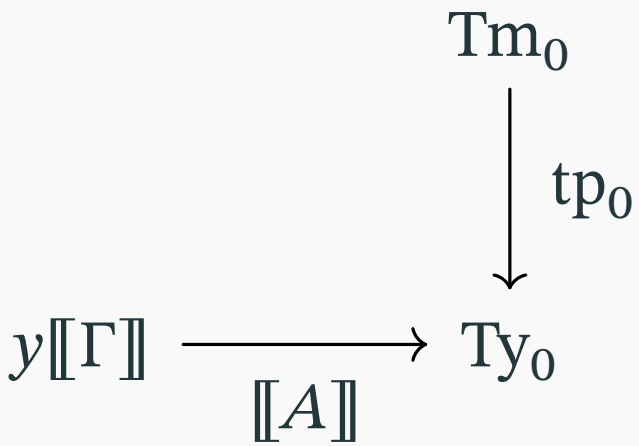
Semantics: natural models in $[\mathcal{C}^{\mathrm{op}}, \mathrm{Set}]$

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

$$\begin{array}{ccc} & & \mathrm{Tm}_0 \\ & & \downarrow{\scriptstyle \mathrm{tp}_0} \\ y[\![\Gamma]\!] & & \mathrm{Ty}_0 \end{array}$$

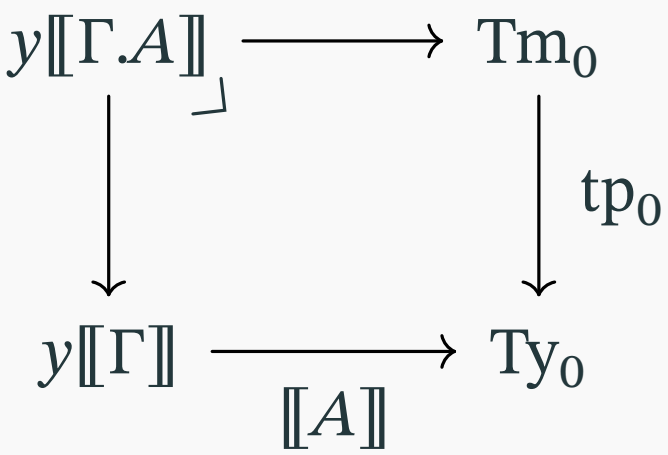
Semantics: natural models in $[\mathcal{C}^{\mathrm{op}}, \mathrm{Set}]$

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



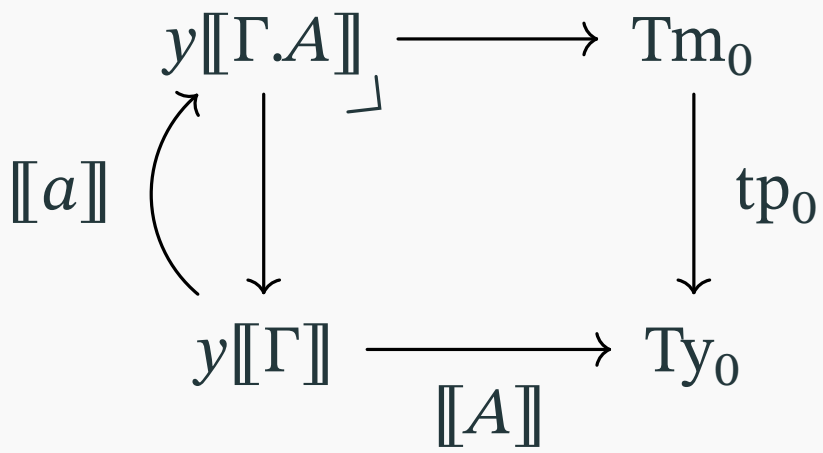
Semantics: natural models in $[\mathcal{C}^{\mathrm{op}}, \mathrm{Set}]$

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



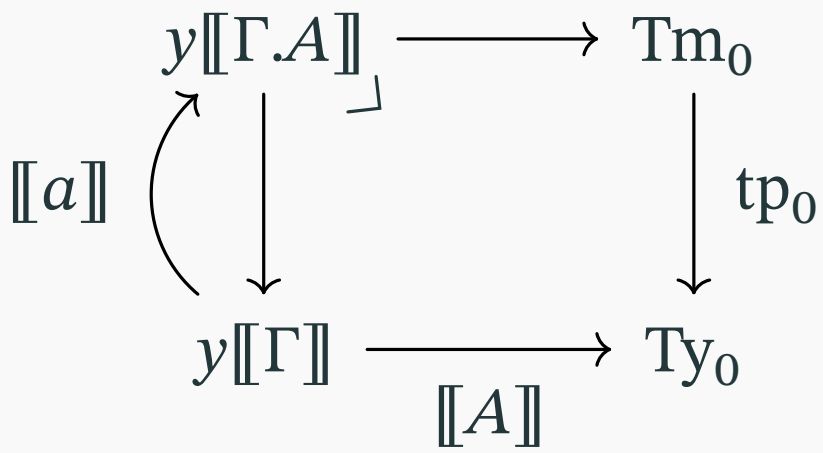
Semantics: natural models in $[\mathcal{C}^{\mathrm{op}}, \mathrm{Set}]$

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



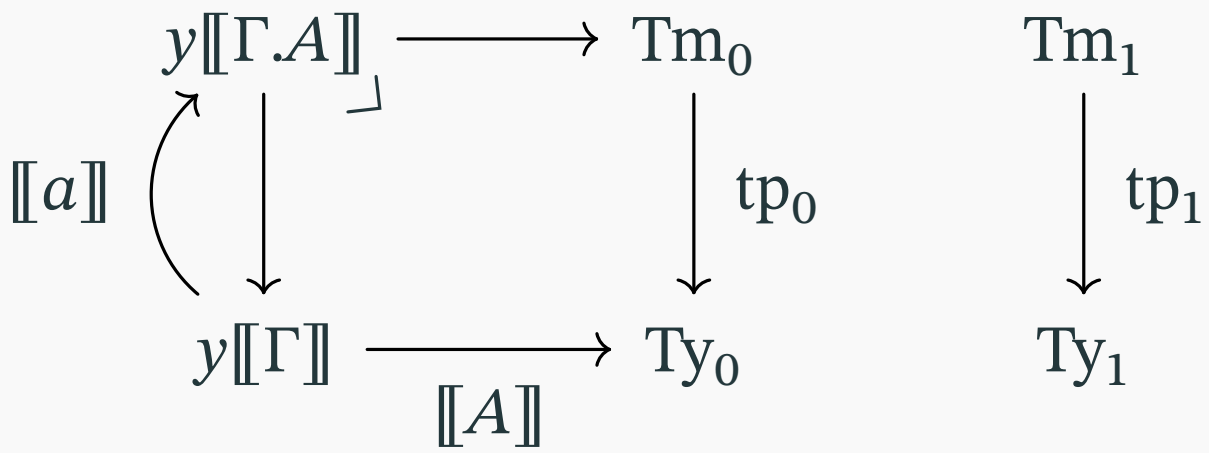
Semantics: universes

Let $\Gamma \vdash_0 A$ be a type. We have $\Gamma \vdash_1 U_0$, so that $\Gamma \vdash_1 \mathrm{code}\, A : U_0$.



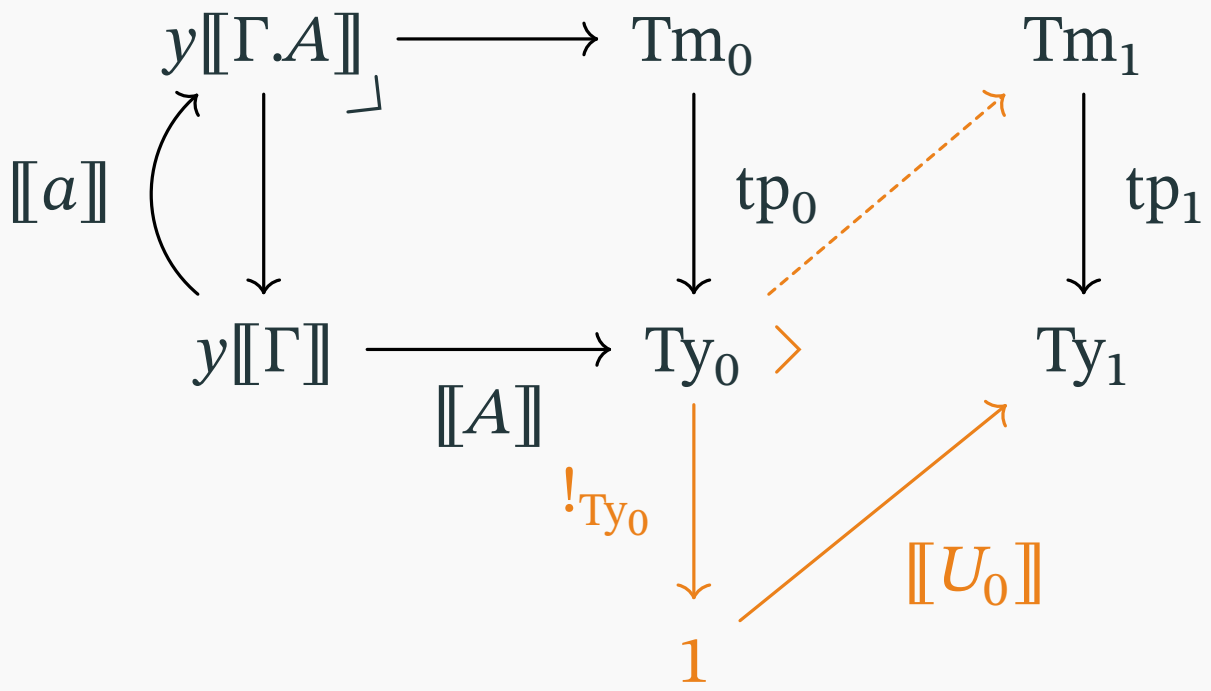
Semantics: universes

Let $\Gamma \vdash_0 A$ be a type. We have $\Gamma \vdash_1 U_0$, so that $\Gamma \vdash_1 \mathrm{code}\, A : U_0$.



Semantics: universes

Let $\Gamma \vdash_0 A$ be a type. We have $\Gamma \vdash_1 U_0$, so that $\Gamma \vdash_1 \mathrm{code}\, A : U_0$.

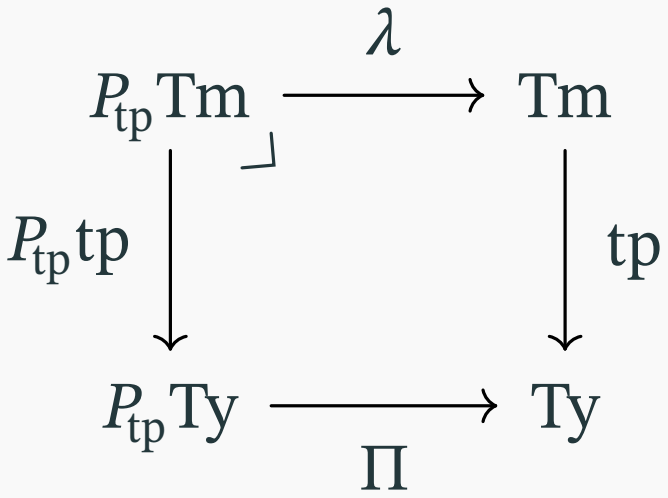


Semantics: Π types

The projection $\mathrm{Tm} \overset{\mathrm{tp}}{\to} \mathrm{Ty}$

defines a polynomial endofunctor $[\mathcal{C}^{\mathrm{op}}, \mathrm{Set}] \overset{P_{\mathrm{tp}}}{\to} [\mathcal{C}^{\mathrm{op}}, \mathrm{Set}]$

such that $\mathrm{Hom}\big(y[\![\Gamma]\!], P_{\mathrm{tp}}\mathrm{Ty}\big) \simeq (A : \mathrm{Hom}(y[\![\Gamma]\!], \mathrm{Ty})) \times \mathrm{Hom}(y[\![\Gamma.A]\!], \mathrm{Ty})$.

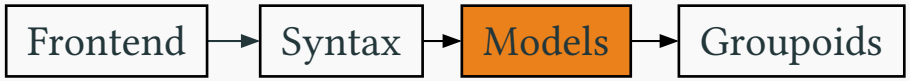


$$\frac{\Gamma \vdash A \qquad \Gamma.A \vdash B}{\Gamma \vdash \Pi A.B}$$

Semantics in Lean

```
structure NaturalModelBase (Ctx : Type u) [Category Ctx] where
  Tm : Psh Ctx
  Ty : Psh Ctx
  tp : Tm ⟶ Ty
  ext {Γ : Ctx} (A : y(Γ) ⟶ Ty) : Ctx
  disp {Γ : Ctx} (A : y(Γ) ⟶ Ty) : ext A ⟶ Γ
  var {Γ : Ctx} (A : y(Γ) ⟶ Ty) : y(ext A) ⟶ Tm
  disp_pullback {Γ : Ctx} (A : y(Γ) ⟶ Ty) : IsPullback (var A) ym(disp A) tp A

structure UHom (M N : NaturalModelBase Ctx) extends Hom M N where
  U : y(𝟙_ Ctx) ⟶ N.Ty
  U_pb : ∃ v : M.Ty ⟶ N.Tm, IsPullback v (yUnit_terminal.from M.Ty) N.tp U
```



Syntax

Presentation: MLTT stratified by universe + presuppositions

$\boxed{\Gamma \vdash_\ell A}$

$$\frac{\ell < \ell_{\max}}{\Gamma \vdash_{\ell+1} U_\ell} \qquad \frac{\Gamma \vdash_{\ell+1} \mathrm{a} : U_\ell}{\Gamma \vdash_\ell \mathrm{El}\ \mathrm{a}} \qquad \frac{\Gamma \vdash_\ell A \qquad \Gamma.A_\ell \vdash_{\ell'} \mathrm{B}}{\Gamma \vdash_{\max(\ell,\ell')} \Sigma_{\ell,\ell'} A.\, B}$$

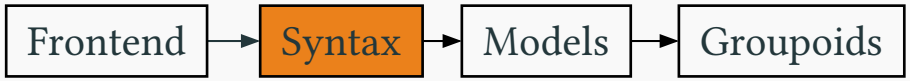
$\boxed{\Gamma \vdash_\ell t : A}$

$$\frac{\ell < \ell_{\max} \qquad \Gamma \vdash_\ell \mathrm{A}}{\Gamma \vdash_{\ell+1} \mathrm{code}\, A : U_\ell}$$

$\boxed{\Gamma \vdash_\ell A \equiv B}$

$$\frac{\ell < \ell_{\max} \qquad \Gamma \vdash_\ell \mathrm{A}}{\Gamma \vdash_\ell \mathrm{El}\ \mathrm{code}\, A \equiv A}$$

+ axioms: set univalence & function extensionality



Syntactic metatheory — selected results NEW

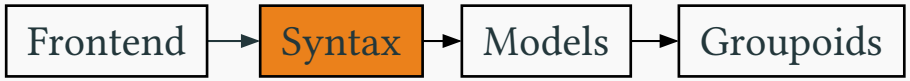
Theorem (admissible substitution). If $\Gamma \vdash_\ell \mathcal{J}$ and $\Delta \vdash \sigma : \Gamma$, then $\Delta \vdash_\ell \mathcal{J}[\sigma]$.

Theorem (application inversion). If $\Gamma \vdash_\ell \mathrm{app}_{\ell,\ell',B}(f, a) : C$, then exists A s.t. $\Gamma \vdash_{\max(\ell,\ell')} f : \Pi_{\ell,\ell'} A.\, B$ and $\Gamma \vdash_\ell a : A$ and $\Gamma \vdash_{\ell'} C \equiv B[a]$.

Conjecture (Σ/Π injectivity). If $\Gamma \vdash_{\max(\ell,\ell')} \Sigma_{\ell,\ell'} A.\, B \equiv \Sigma_{\ell,\ell'} A'.\, B'$, then $\Gamma \vdash_\ell A \equiv A'$ and $\Gamma.A \vdash_{\ell'} B \equiv B'$; and similarly for Π.

↯ Proving judgmental equalities without QIITs ~ setoid hell.

? Is the standard QIIT presentation of MLTT definable with Lean quotients?

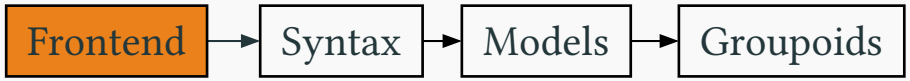


Frontend

Demo

```
hott def idfun : Π {A : Type}, A → A := fun a ⇒ a

/- { l := 1,
     val := lam 1 0 (univ 0) (lam 0 0 (el (bvar 0)) (bvar 0)),
     tp := pi 1 0 (univ 0) (pi 0 0 (el (bvar 0)) (el (bvar 1))),
     wf := (… : [] ⊢[l] val : tp) } -/
#print idfun.checked
```



NbE-based typechecker NEW

❌ Verified ✅ Verifying

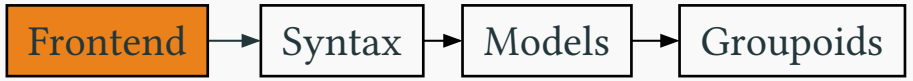
Correctness: inductive relation between values and terms/types.

```
partial def evalTp (env : Q(List NbE.Val)) (T : Q(HoTT0.Expr)) :
  Lean.MetaM ((v : Q(Val)) × Q(∀ {Γ Δ σ l}, EnvEqSb Δ $env σ Γ → (Γ ⊢[l] $T) →
    ValEqTp Δ l $v (($T).subst σ))
```

No special construction (e.g. Bove-Capretta [2]) needed for termination.

Optimizations: $\mathcal{O}(0)$ weakening (de Bruijn levels), defunctionalized closures à la Abel [3], proof term sharing.

❓ No single typing derivation *needs* injectivity of Σ/Π.



Interpretation

Partial interpretation

❓ $[\![\Gamma \vdash_\ell A]\!] : (\Gamma : \mathrm{List\ Expr})\,(\ell : \mathbb{N})\,(A : \mathrm{Expr}) \to (\Gamma \vdash_\ell A) \to (y[\![\Gamma]\!], \mathrm{Ty}_\ell)$

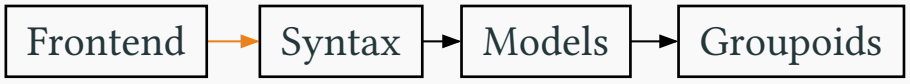
❌ Recursive-recursive! Not directly definable in Lean.

Instead, define $[\![A]\!]_{X,\ell} : (A : \mathrm{Expr})\,(X : \mathcal{C})\,(\ell : \mathbb{N}) \rightharpoonup \mathrm{Hom}(yX, \mathrm{Ty}_\ell)$

✅ Soundness: if $\Gamma \vdash_\ell A \equiv B$, then $[\![A]\!]_{[\![\Gamma]\!],\ell}\downarrow$ and $[\![A]\!]_{[\![\Gamma]\!],\ell} = [\![B]\!]_{[\![\Gamma]\!],\ell}$

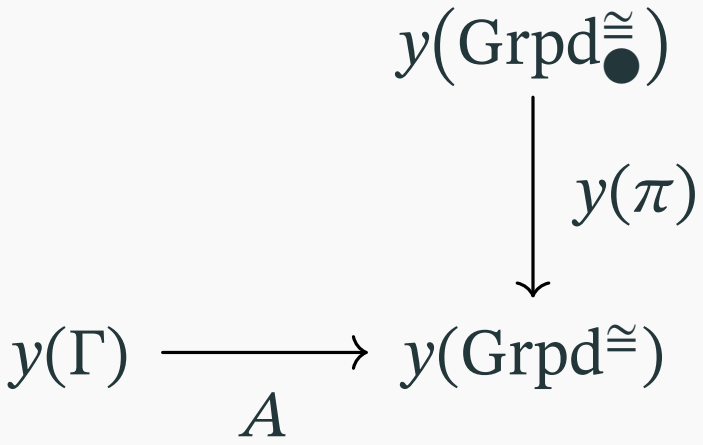
↯ Formalizing "obvious" naturality laws (see Zulip discussion).

NEW `rw!` tactic, developed with Aaron Liu, now in mathlib.



Groupoid model

Model in $[\mathrm{Grpd}^{\mathrm{op}}, \mathrm{Set}]$



1. Get $\tilde{A} : \Gamma \to \mathrm{Grpd}^{\cong}$
2. Form Grothendieck construction $\int A = (\gamma : \Gamma) \times A(\gamma)$
3. This a groupoid; take $\Gamma.A \triangleq \int A$

? Everything is representable; work in Grpd directly?

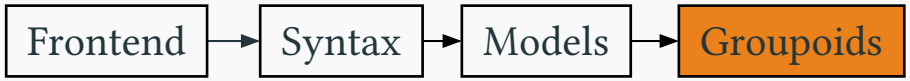
Recent developments in the model NEW

Construction of Σ and Π types (nearly `sorry`-free).

Search for best definition of 1-categorical Grothendieck constructions:

- Originally defined explicitly.
- Redefined to use `mathlib`'s 2-categorical definition: proofs too slow.
- Redefined again to original, proved equivalence with 2-cat. version.
- Upstreaming to `mathlib`: #27321.

↯ Pervasive proof-checking performance issues.



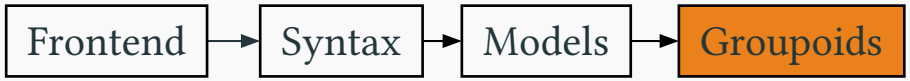
Necessary evil

Mathlib avoids "evil" category theory: equal objects, equal functors, isomorphic categories…

Groupoid model *necessitates* evil: e.g. $\int A$ is a strict 1-pullback of categories.

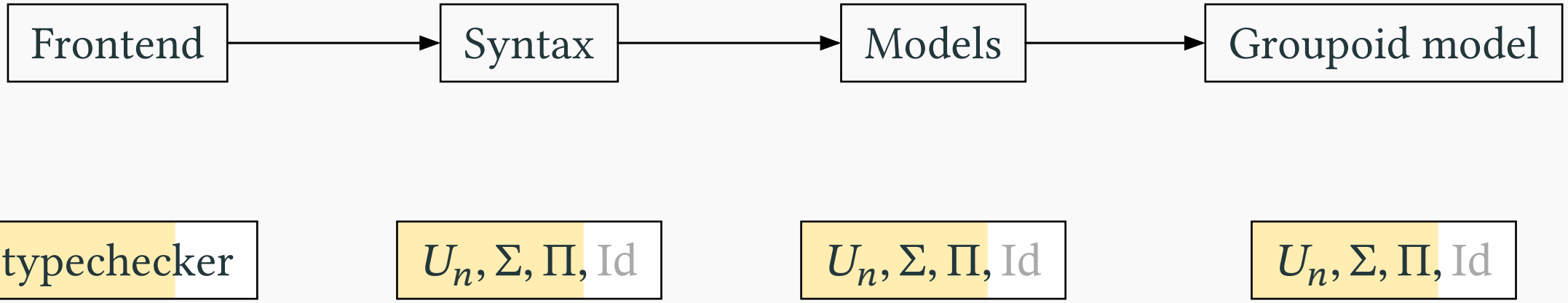
Mathlib definitions not general enough: due to non-cumulativity of universes, there is no category of κ-small categories in which strict 1-pullbacks live — we developed a `MegaPullback` API.

↯ Equal functors into `Cat` ⇒ equal types ⇒ DTT hell.




Friendship ended with
FUNCTOR.EXT
Now
FUNCTOR.HEXT
is my
best friend
imgflip.com

Project status



+ Set Univalence + Function Extensionality

Thank you

We welcome contributions!

- Polynomial functor theory
- Construction of Id types
- Injectivity of Σ/Π
- Many small lemmas

Issue tracker: github.com/sinhp/groupoid_model_in_lean4

Related work

S. Hazratpour, E. Riehl. Poly.

A. Kolomatskaia. Stepping by Evaluation.

[4] M. Sozeau and N. Tabareau, “Towards an internalization of the groupoid model of type theory,” *Types for Proofs and Programs 20th Meething (TYPES 2014), Book of Abstracts*, 2014.

[5] K. Maillard and Y. Xu, “Geometric Reasoning in Lean: from Algebraic Structures to Presheaves,” *TYPES 2025*, 2025.

Bibliography

[1] I. Blechschmidt, "Using the internal language of toposes in algebraic geometry," 2021.

[2] A. Bove and V. Capretta, "Modelling general recursion in type theory," *Math. Struct. Comput. Sci.*, vol. 15, no. 4, pp. 671–708, 2005, doi: 10.1017/S0960129505004822.

[3] A. Abel, "Normalization by Evaluation: Dependent Types and Impredicativity," 2013.

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

[5] K. Maillard and Y. Xu, “Geometric Reasoning in Lean: from Algebraic Structures to Presheaves,” *TYPES 2025*, 2025.