Localizations of models of dependent type theory

Author: Matteo Durante Advisor: Hoang-Kim Nguyen

Regensburg University

February 14, 2026

Objective

A modern proof of the following theorem.

Theorem (Kapulkin 2015)

Given a dependent type theory $\mathbf{T}$ *with* Σ-, Id- *and* Π*-types, the* ∞*-localization of its syntactic category* $\mathrm{Syn}(\mathbf{T})$ *is a locally cartesian closed* ∞*-category.*

Dependent type theory

What

A theory of computations and a foundation of mathematics.

Dependent type theory

What

A theory of computations and a foundation of mathematics.

Objects

Dependent types A and their *terms* $x : A$ in *contexts* $\Gamma = (x_0 : A_0, \dots, x_n : A_n)$.

Dependent type theory

What

A theory of computations and a foundation of mathematics.

Objects

Dependent types A and their *terms* $x : A$ in *contexts* $\Gamma = (x_0 : A_0, \ldots, x_n : A_n)$.

Structural rules

How to work with *variables*.

Dependent type theory

What

A theory of computations and a foundation of mathematics.

Objects

Dependent types A and their *terms* $x : A$ in *contexts* $\Gamma = (x_0 : A_0, \dots, x_n : A_n)$.

Structural rules

How to work with *variables*.

Logical rules

Construct new types and their terms from old, carry out computations. They provide Σ-*types* $\Sigma(A, B)$, Π-*types* $\Pi(A, B)$, Id-*types* Id_A, *natural-numbers-type* Nat. . .

Models

Idea

To reason about a theory we can look at its interpretations.

Models

Idea

To reason about a theory we can look at its interpretations.

Problem

Providing a model of dependent type theory is hard.

Models

Idea

To reason about a theory we can look at its interpretations.

Problem

Providing a model of dependent type theory is hard.

Solution

Defining a class of algebraic models.

Modeling structural rules

Definition (contextual categories)

A category C with:

1. a grading on objects (or *contexts*) $\operatorname{Ob} \mathsf{C} = \coprod_{n \in \mathbb{N}} \operatorname{Ob}_n \mathsf{C}$;
2. a unique and terminal object in $\operatorname{Ob}_0 \mathsf{C}$, the *empty context*;
3. a map $\mathrm{ft}_n \colon \operatorname{Ob}_{n+1} \mathsf{C} \to \operatorname{Ob}_n \mathsf{C}$ for each $n \in \mathbb{N}$;
4. *basic dependent projections* $p_A \colon \Gamma.A \to \mathrm{ft}_n(\Gamma.A) = \Gamma$;
5. a functorial choice of pullback squares

$$\begin{tikzcd}
\Delta.f^*A \arrow[r, "q(f{,}A)"] \arrow[d, "p_{f^*A}"'] & \Gamma.A \arrow[d, "p_A"] \\
\Delta \arrow[r, "f"'] & \Gamma
\end{tikzcd} .$$

Modeling logical rules

Extra structure

Id-types require from $\Gamma.A$ an Id-object $\Gamma.A.A.\,\mathrm{Id}_A$...
Π-types require from $\Gamma.A.B$ a Π-object $\Gamma.\Pi(A,B)$, an evaluation map $\mathrm{app}_{A,B}\colon \Gamma.\Pi(A,B).A \to \Gamma.A.B$, $(f,a) \mapsto (a, \mathrm{app}(f,a))$...

Modeling logical rules

Extra structure

Id-types require from $\Gamma.A$ an Id-object $\Gamma.A.A.\operatorname{Id}_A$...
Π-types require from $\Gamma.A.B$ a Π-object $\Gamma.\Pi(A, B)$, an evaluation map $\operatorname{app}_{A,B}\colon \Gamma.\Pi(A, B).A \to \Gamma.A.B$, $(f, a) \mapsto (a, \operatorname{app}(f, a))$...

Example

If $\mathbf{T}$ has some logical rules, then its (contextual) syntactic category $\operatorname{Syn}(\mathbf{T})$ has the corresponding logical structures. It is freely generated by the theory: objects are contexts, morphisms $[x_0 : A_0, \ldots, x_n : A_n] \to [y_0 : B_0, \ldots, y_m : B_m]$ are tuples of terms $(f_0 : B_0, \ldots, f_m : B_m)$ derivable from $x_0 : A_0, \ldots, x_n : A_n$.

Bi-invertibility

Definition (bi-invertible map)

A map $f : \Gamma \to \Delta$ *in a contextual category with* Id*-structure* C *for which we can provide:*

1. *maps* $g_1 \colon \Delta \to \Gamma$, $\eta \colon \Gamma \to \Gamma.(1_\Gamma, g_1 \cdot f)^* \,\mathsf{Id}_\Gamma$*;*
2. *maps* $g_2 \colon \Delta \to \Gamma$, $\epsilon \colon \Delta \to \Delta.(1_\Delta, f \cdot g_2)^* \,\mathsf{Id}_\Delta$*.*

Bi-invertibility

Definition (bi-invertible map)

A map $f\colon \Gamma \to \Delta$ *in a contextual category with* Id*-structure* C *for which we can provide:*

1. *maps* $g_1\colon \Delta \to \Gamma$, $\eta\colon \Gamma \to \Gamma.(1_\Gamma, g_1 \cdot f)^* \mathrm{Id}_\Gamma$;
2. *maps* $g_2\colon \Delta \to \Gamma$, $\epsilon\colon \Delta \to \Delta.(1_\Delta, f \cdot g_2)^* \mathrm{Id}_\Delta$.

Question

What if we localize at bi-invertible maps?

Fibrational structure

Definition (∞-categories with weak equivalences and fibrations)

A triple $(\mathcal{C}, W, Fib)$ *where:*
...a weakening of the definition of fibration categories, with $\mathcal{C}$ *an* ∞*-category.*

Fibrational structure

Definition (∞-categories with weak equivalences and fibrations)

A triple $(\mathcal{C}, W, Fib)$ where:
...a weakening of the definition of fibration categories, with $\mathcal{C}$ an ∞-category.

Theorem (Avigad-Kapulkin-Lumsdaine 2013)

A contextual category with Σ- and Id-structures defines a fibration category, where weak equivalences are bi-invertible maps and fibrations are maps isomorphic to compositions of basic dependent projections $p_A \colon \Gamma.A \to \Gamma$.

Localizing fibrational categories

Construction (fibrant slice $\mathcal{C}(x)$)

Given a fibrant object x in $\mathcal{C}$, lift the fibrational structure through $\mathcal{C}/x \to \mathcal{C}$ and then take the subcategory of fibrant objects of $\mathcal{C}/x$.

Localizing fibrational categories

Construction (fibrant slice $\mathcal{C}(x)$)

Given a fibrant object x in $\mathcal{C}$, lift the fibrational structure through $\mathcal{C}/x \to \mathcal{C}$ and then take the subcategory of fibrant objects of $\mathcal{C}/x$.

Proposition (Cisinski)

Given an ∞-category with weak equivalences and fibrations $\mathcal{C}$, if for every fibration $f : x \to y$ between fibrant objects the pullback functor between fibrant slices $f^* : \mathcal{C}(y) \to \mathcal{C}(x)$ has a right adjoint preserving trivial fibrations, then $L(\mathcal{C})$ is locally cartesian closed.

Localizations of models are cartesian closed

Theorem (Kapulkin 2015)

Given a dependent type theory $\mathbf{T}$ *with* Σ-, Id- *and* Π-*types, the localization of its syntactic category* $\mathrm{Syn}(\mathbf{T})$ *is a locally cartesian closed* ∞-*category.*

Localizations of models are cartesian closed

Theorem (Kapulkin 2015)

Given a dependent type theory $\mathbf{T}$ *with* Σ*-,* Id*- and* Π*-types, the localization of its syntactic category* $\mathrm{Syn}(\mathbf{T})$ *is a locally cartesian closed* ∞*-category.*

Proof.

For any basic dependent projection $p_A \colon \Gamma.A \to \Gamma$, there exists a right adjoint to $p_A^* \colon \mathrm{Syn}(\mathbf{T})(\Gamma) \to \mathrm{Syn}(\mathbf{T})(\Gamma.A)$ given by

$$(p_A)_*(\Gamma.A.\Theta) = \Gamma.\Pi(A, \Theta)$$

with counit induced by $\mathrm{app}_{A,\Theta}$. It preserves the fibrational structure. □

Thank you for your attention!

Issues

Essentially, folklore.

Extended structures

We used extensions of the Id-, Σ- and Π-structures: from $\Gamma.A.\Theta$ we have a Π-object $\Gamma.\Pi(A, \Theta)$ only when $\mathrm{ft}(\Gamma.A.\Theta) = \Gamma.A$, however Θ represents an arbitrary extension, like $\Gamma.A.B$ or $\Gamma.A.B.C$. These extended structures were mentioned in the literature, but not entirely defined.

Internal languages

Researchers often argue by relying on them, intuitive but undefined tools. We chose to work with syntactic categories because then we can reason as we wish.

Why is dependent type theory cool?

1. Closely linked to *computations* and *computer science*, makes proof assistants possible.
2. Enough by itself as a foundation, unlike set theory or propositional calculus.
3. *Proofs* are internal objects.
4. Better treatment of *equality*.
5. Makes "fully faithful + essentially surjective = equivalence" independent from the axiom of choice.
6. Homotopical interpretation in ∞-groupoids.

Internal languages conjecture

Conjecture (Kapulkin-Lumsdaine 2016)

The horizontal maps, given by simplicial localization, induce equivalences of ∞-categories.

$$\begin{tikzcd}
\mathsf{CxlCat}_{\Sigma,1,\mathsf{Id},\Pi} \arrow[r] \arrow[d] & \mathsf{LCCC}_{\infty} \arrow[d] \\
\mathsf{CxlCat}_{\Sigma,1,\mathsf{Id}} \arrow[r] & \mathsf{Lex}_{\infty}
\end{tikzcd}$$

A proof by Nguyen-Uemura has recently become available on arxiv. One hopes to extend this to an equivalence between $\mathsf{CxlCat}_{\mathsf{HoTT}}$ and $\mathsf{ElTopos}_{\infty}$.