Localizations of Models of Dependent Type Theory

Matteo Durante
Advisor: Hoang-Kim Nguyen
Coadvisor: Denis-Charles Cisinski

Regensburg University

June 22, 2022

Introduction

Dependent Type Theory is a candidate foundation for mathematics. Why is it interesting?

1. closely linked to *computations* and *computer science*, thereby allowing the creation of programming languages and software tools like Coq, Lean and Agda to formalize and check mathematical reasoning and even automatically generate some tedious bits;
2. sufficient by itself, unlike Set Theory and Propositional Calculus;
3. *proofs* are internal objects;
4. allows a more nuanced concept of *equality*, where a *proof of equality* expresses in what sense two objects are equal.

Dependent Type Theory

Dependent Type Theory talks about *dependent types* A *over contexts* Γ and their *terms* $x : A$.
It gives *structural rules* to specify how to work with variables and *logical rules* to construct new types and their terms from old and how to carry out computations through extra structure, like Σ-*types* $\Sigma(A, B)$ (corresponding to coproducts), Π-*types* $\Pi(A, B)$ (also called *function types*) and Id-*types* Id_A.
Problem: providing a model of Dependent Type Theory is hard because we have to deal with a lot of bureaucracy, especially from the structural rules.
Solution: defining an algebraic model so that we can build new models from them.

Contextual Categories

Definition

A category C with:

1. a grading on objects $\mathrm{Ob}\,\mathsf{C} = \coprod_{n \in \mathbb{N}} \mathrm{Ob}_n\,\mathsf{C}$, where $\mathrm{Ob}_0\,\mathsf{C}$ has only one element, which is the terminal;
2. a map $\mathrm{ft}_n \colon \mathrm{Ob}_{n+1}\,\mathsf{C} \to \mathrm{Ob}_n\,\mathsf{C}$ for each $n \in \mathbb{N}$;
3. for each object $\Gamma.A \in \mathrm{Ob}_{n+1}\,\mathsf{C}$ a *basic dependent projection* $p_A \colon \Gamma.A \to \mathrm{ft}(\Gamma.A) = \Gamma$;
4. for each morphism $f \colon \Delta \to \Gamma$ and $p_A \colon \Gamma.A \to \Gamma$, a functorial choice of a pullback square;

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

This only models the structural rules.

Extra Structure

Modeling logical rules requires some extra structure. Id-types require:

1. from $\Gamma.A$ an object $\Gamma.A.A.\mathrm{Id}_A$;
2. ...

Π-types require:

1. from $\Gamma.A.B$ an object $\Gamma.\Pi(A, B)$;
2. a map $\mathrm{app}_{A,B} : \Gamma.\Pi(A, B).A \to \Gamma.A.B$ modeling function application;
3. ...

The data needed to model the logical rules we mentioned are called Σ-structure, Id-structure and Π-structure.

Definition

A categorical model of type theory is a contextual category C *with* Σ, Id *and* Π *structure. The associated category with structure-preserving functors as maps is* $Cxl_{\Sigma,\mathrm{Id},\Pi}$.

Bi-Invertibility

In Dependent Type Theory, the right notion of invertible map to consider is the one of *bi-invertible map*. This is modeled in contextual categories with Id-structure by the following definition.

Definition

Under the above setting, a map $f : \Gamma \to \Delta$ is bi-invertible if we can give:

1. *maps* $g_1 \colon \Delta \to \Gamma$, $\eta \colon \Gamma \to \Gamma.(1_\Gamma, g_1 f)^* \mathrm{Id}_\Gamma$*;*
2. *maps* , $g_2 \colon \Delta \to \Gamma$, $\epsilon \colon \Delta \to \Delta.(1_\Delta, fg_2)^* \mathrm{Id}_\Delta$*.*

Essentially η, ϵ are showing that the maps coincide pointwise. Assuming function extensionality, this tells us that f is invertible. Question: what happens if we localize a contextual category at bi-invertible maps?

Internal Languages Conjecture

Conjecture (Kapulkin-Lumsdaine, 2016)

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

$$\begin{tikzcd} Cxl_{\Sigma,1,\mathrm{Id},\Pi} \arrow[r] \arrow[d] & LCCC_\infty \arrow[d] \\ Cxl_{\Sigma,1,\mathrm{Id}} \arrow[r] & Lex_\infty \end{tikzcd}$$

A proof by Uemura and Nguyen has recently become available on arxiv and one hopes to extend this to an equivalence between Cxl_{HoTT} and $ElTopos_\infty$.

Fibrational Structure

Definition

An ∞-category with weak equivalences and fibrations is a triple $(\mathcal{C}, W, Fib)$ where:

1. *$\mathcal{C}$ is an ∞-category and W, Fib are subcategories;*
2. *$\mathcal{C}$ has a terminal object;*
3. *W has the 2-out-of-3 property;*
4. *the pullback of a map in Fib (or $W \cap Fib$) with a fibrant codomain still lies in Fib (respectively, $W \cap Fib$);*
5. *every morphism can be factored as a morphism in W followed by one in Fib.*

Theorem

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 dependent projections.

Localizing Fibrational Categories

Theorem

The localization at weak equivalences of an ∞-category with weak equivalences and fibrations $\mathcal{C}$ is a finitely complete ∞-category $L(\mathcal{C})$.

Theorem

Given an ∞-category with weak equivalences and fibrations $\mathcal{C}$,