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 substitute variables and *logical rules* to construct new types and their terms from old and how to carry out computations through extra logical 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 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* $Cxl_{\Sigma,\mathrm{Id},\Pi}$.

Internal Languages Conjecture

Dependent type theories also have models provided by homotopy theory, like the category of ∞-groupoids. In particular, it has been conjectured that Dependent Type Theory is the *internal language of ∞-categories*.

Conjecture

The horizontal maps by quasi-localizing contextual categories induce equivalences of ∞-categories.

$$
\begin{array}{ccc}
Cxl_{\Sigma,1,\mathrm{Id},\Pi} & \longrightarrow & LCCC_{\infty} \\
\downarrow & & \downarrow \\
Cxl_{\Sigma,1,\mathrm{Id}} & \longrightarrow & Lex_{\infty}
\end{array}
$$

A proof by Uemura and Nguyen has recently been published on arxiv and one hopes to build from this an equivalence between Cxl_{HoTT} and $ElTopos_{\infty}$.