E/Ea Thompson

Lean 4: Formalization of Mathematics

– In Pursuit of Abstract Nonsense –

Tuesday 5th September, 2023

Springer Nature

Preface

This text consists of a collection of Lean 4 notes taken during PIMS Math 681: Formalization of Mathematics, taught by Dr. Adam Topaz.

Place(s),
month year

Firstname Surname
Firstname Surname

Contents

Part I

Foundations

Chapter 1

Basics

1.1 Types

Informally, types are “things" that “contain" terms. In dependent type theory, everything is a **term**, and every term has a **type**. Notationally, we write $t\ :\ \tau$ to denote the fact that t is a **term** of **type** τ.

Example 1.1 *We have the type of natural numbers, $\mathbb{N}$, as well as the type of rationals, $\mathbb{Q}$, reals, $\mathbb{R}$, etc.*

As all types are terms, what is the type of $\mathbb{N}$? Well it is **Type**! (well technically **Type** 0) This means we need a heirarchy of types, so that **Type** 0 can also be a term of the type, **Type** 1, etc.

We can also put types together to form **function types**, such as $\mathbb{N} \to \mathbb{N}, \mathbb{N} \to \mathbb{Q}$, and $\mathbb{N} \to \mathbb{R}$. In type theory, the function constructor on types is a **primitive object**.

Example 1.2 *If we want a squaring function, for example, in Lean4 we can specify it by:*

$$\textit{fun}\ n\ :\ \mathbb{N} \mapsto n^2$$

The “fun" keyword in Lean4 can be used as a lambda syntax, but it also has a lambda syntax as well.

1.1.1 Propositions

In dependent type theory, and Lean4 specifically, propositions are also modeled by types. In order to discuss this we introduce the type hierarchy “Sort $n\ :\ $ Sort $(n+1)$" where Type $u\ :=\ $ Sort $(u+1)$. In this hierarchy, Sort 0 is given the special notation Prop, and denotes the type of propositions.

Lean’s type theory has a special feature known as **proof irrelevance**, which is inconsistent with HoTT. Proof irrelevance means that all proofs are equal in Lean4.

Note that dependent type theory is useful for propositions, and mathematics generally, so that we can perform quantification in our propositions.

1.2 Curry-Howard-Correspondence

Appendix A
Formal Type Theory

All's well that ends well