CMSC631 Final Project Report

Jeffery, Sai & David

Fall 2025

Objective

Initially, we wanted to prove the big results in Model Theory: Gödel's First Incompleteness Theorem, Łoś's Theorem, and Löwenheim–Skolem Theorem. We decided we would focus primarily on Gödel's First Incompleteness if we did not make sufficient progress. Since the result does not need as many definitions, theorems, or lemmas.

We picked this project since we did not see much Model Theory formalization out there, especially in Rocq. We initially considered doing it in Lean, since it is more adopted for Math formalization than Rocq. We did not end up doing that to avoid having to learn both Model Theory and Lean at the same time. Which did not seem feasible given the project's timeframe.

Results

We ended up with different files representing Kripke's structure or Tarski's semantics. Plenty of definitions for elementary Model Theory. Some Lemmas/Theorems, and small proofs/examples. We had quite a bit of unresolved proofs, since making progress was seemingly very difficult.

Difficulties

Model Theory

Some of the group picked this project with the goal of learning Model Theory along the way. This ended up being a bigger hurdle than one would have imagined. Mainly because of the time commitment to consuming information and retaining it, but also because formalizing elementary Model Theory requires a lot deeper of an understanding than one would have thought. There are multiple ways of proving our desired theorems in Math, but we did not really have enough awareness to be able to pick a system which would lend itself well to Rocq.

Satisfying Proof Goals

A huge hurdle was actually being able to prove things about our definitions. Even simple Lemmas which follow directly from definitions were hard to prove due to weak definitions. Eventually, we decided to overhaul our project from the ground up, in order to have stronger definitions. Then we ran into the issue of getting stuck writing definitions.

Non-Trivial Construction

It's worth noting that we have a lot of *unused* definitions. This is because we did not get around to proving things about every definition, or even constructing examples. This is because translating a mathematical definition into a valid Rocq implementation is quite difficult. Representing invariants is not obvious in Rocq, such as if a set $S \neq \emptyset$. Or how

Const should have been encoded in our records. We ended up with *no* explicit Constants type/set, since we could embed them in functions as functions with no arguments.

Unfamiliarity With Rocq

In class we focused mainly on solving proofs using tactics. Where structures and ideas were provided to us. For this project, we had to translate Mathematical ideas into Rocq. This was very different than anything we had done before in Rocq. We had a lot of trouble starting. Since our definitions for $\mathcal{L}$-terms ended up being *too* weak. In addition, we kept running into issues where Rocq was not convinced of something (such as terminating, or not being able to guess the decreasing argument). In these instances, we simply had to write around these problems. Since we did not have enough time to become very familiar with Rocq.