CMSC631 Final Project Report

Jeffery, Sai & David

Fall 2025

Code: https://github.com/119254998/model-theory-rocq

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, with secondary fallbacks to proving the soundness and completeness of first order logic (which implies compactness).

We initially picked this project because of the lack of Model Theory formalization 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 ended up not being feasible given the project's timeframe.

In the end, we opted to work on a dual approach of implementing FOL using Kripke-style semantics, and proving soundness, completeness, and compactness along both these axes.

An Overview of Previous Work

With regards to the formalization of first order logic, almost all work had been done in the past either under Tarski's definition of truth (see the Goedel project, by far the most complete and a multi year collaboration with hydra that's currently hosted at https://github.com/rocq-community/goedel), along with other approaches, including:

- Synthetically, UDS (Saarland University)'s Dominik Kirst and Benjamin Peters proved synthetic incompleteness at https://github.com/uds-psl/coq-synthetic-incompleteness
- Trivial compactness at https://github.com/pi314mm/compactness
- And a basic encoding of peano at https://gist.github.com/andrejbauer/6ff643f672bb1a52ddfbf948b558cd20

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 initially thought, in part because of the time commitment to consuming information and retaining it, but also because formalizing elementary Model Theory requires a deep understanding.

On paper, our desired theorems are easy to prove, but they rely on human syntactic sugar and understanding; due to the low level nature of logic proofs, these theorems are much harder to prove in a formal system. Additionally, some of the group had trouble switching from Tarski to Kripke style semantics, which fundamentally change the way the model is encoded, and thus (even if the on paper proof is the same) changes the way that you do things in Rocq (since it looks at the underlying structure when inducting over it, etc).

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. In order to get around this, we `Admitted` a lot of lemmas regarding substitution or other technical conerns so that we could prove our main goal of completeness and compactness.

Additionally, time constraints meant that we could not spend as much time on the project as we would have liked, which meant that some goals were admitted purely out of necessity for progress. For instance, the substitution lemmas defined under basics.v are likely provable given time, but because they are noncritical for our main goal, are obviously correct (with trivial on paper proofs), and mainly exist to support the structure of Kripke semantics under the bar lambda mu mu tilde calculus, they were omitted.

We additionally omitted some forcing lemma proofs; in general the rule for me was that after an hour of trying to prove a lemma that wasn't critical, it was admittable. I also admitted the proof of existence under completeness, since it's similar to the proof of universal quantification, and we had already admitted higher above due to time constraints. (although it should be clear to the reader that all admitted sections of the Completeness theorem proof are trivial to complete with time, since they amount to tedious copying of already proven cases.)

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.

Results

We were unable to prove the incompleteness of Peano Arithmetic, our primary goal (nor our reach goals of course!) which we initially wanted to do under the conventional Tarski's definition of truth and semantics. This approach, which we adapted from Aris Papadopoulos' work in MATH445, was unable to get very far due to our limited ability to work on the complex induction schemes and definitions that were necessary for this approach.

After our initial struggles with even coming up with an induction scheme in Tarski's, we decided to downgrade our goals to proving soundness and completeness in first order logic. We further struggled on defining the necessary term structure and induction schemes under Tarski, at which point I found a Kripke-style encoding of first order logic under bar lambda mu mu tilde calculus. This encoding was much easier to work with because, as a fundamentally computational bridge kind of model, it allowed us to just use Rocq's native induction schemes and Type system. We were able to prove soundness, completeness, and compactness in this encoding, which is a significant result to our knowledge, as we could only find encodings of FOL in the Kripke style, but could not a proof of soundness/ completeness or compactness.

Compactness Theorem

The compactness theorem states that TFAE: if all finite subsets of a theory have a model (eg are satisfiable) then the entire theory is satisfiable.

Some of our model structure was derived (from non coding papers on how something could be formulated) from other work, which we cited when necessary. It can be noted that the definition of compactness follows immediately from soundness and completeness for reasons cited in Aris' 445 notes, and consequently the proof of compactness was quite quick as can be seen in the code.

During our proof of compactness, we actually proved an analog; eg that the local context of a system has a valid proof system (the Kripke equivalent of being modelable) iff the global context / signature has a valid proof system.

Soundness and Completeness

The soundness theorem was by far the hardest, and is proven in basics. It's structurally foundational, and is a direct consequence of the fact that our Kripke-style semantics is sound. The proof involves the claim that a Gamma that proves phi is a model of phi; we proved this the typical way, by induction over the complexity of phi.

Since we were working in Rocq, we did so not by the typical induction on the complexity of phi but via an analogue utilizing depth. Additionally in order to do so, we defined all relevant terms and theorems, including forcing, which by Rocq's requirements required a fueling mechanism and a consequential really long proof to indicate a secondary definition that fueling is not required is valid.

Completeness was extremely tricky to formulate under Kripke, and I ended up using a construction proposed by a paper in CICM (cited in the code). This construction requires us to prove, essentially, that there existed a reify-reflect pair (eg interpretation of syntax to semantics and vice versa, similar to Tarski's definition of truth but in a more computational style) that witnesses the existence of a bijection isomorphism between syntax and semantics. This demonstrates completeness because it shows that for every sentence in our language, there is a model that satisfies it. One direction was done by induction over the complexity of the sentence, and the other direction was done by induction over the complexity of the model.