

# Elementary Model Theory in Rocq

## Current Progress Report

Tuesday, December 9 2025

Jeffery, Sai & David

# 1 Introduction

# What even is “Model Theory?”

3 / 12

The study of Mathematical “models” and their capabilities, interpretations, and theories.

*Models?*

- Formulas
- Languages
- Sentences
- What are they composed of?
- How do we write things down?

Idea: Given a language, a sentence, and a theory, what does it mean?  
Is the theory valid? Are there some things which cannot be proven in this language?

# Goal

- Implement Elementary Model Theory in Rocq
  - Gödel's First Incompleteness Theorem
  - Łoś's Theorem (ultraproducts)
  - Löwenheim–Skolem Theorem
  - Concerning completeness/compactness
  - Stability

Proofs requires non-trivial definitions, theorems and lemmas.

# What Already Exists

From what he have seen, it is very difficult to find Model Theory libraries and or implementations in Rocq.

- Why?

# Approach

- Define primary concepts and syntax.
- State lemmas and work upwards towards desired theorem.
- Restrictions on `Stdlib`

## 2 Our Progress

# Sections & Definitions

Definitions so Far:

- First-order language:  $\mathcal{L}$ 
  - Logical symbols of  $\mathcal{L}$ :  $\text{Var}, \doteq, \wedge, \vee, \neg, \rightarrow, \leftrightarrow, \forall, \exists$
  - Signature of  $\mathcal{L}$ :  $\text{Const}(\mathcal{L}), \text{Rel}(\mathcal{L}), \text{Fun}(\mathcal{L})$
- Terms of  $\mathcal{L}$  or  $\mathcal{L}$ -*terms*
- Formulas of  $\mathcal{L}$  or  $\mathcal{L}$ -*formulas*

Sections:

- Languages
- Semantics

# Difficulties

- Translating definitions into Rocq:
  - Enforcing properties?
- Weak definitions and or induction schema
- Proving simple things are hard
- Proving in Rocq requires deep knowledge of proof.
  - Learning Model Theory
- Getting stuck

# 3 What's Next

# Readjusting Expectations

Fallbacks:

- Focus on most important results:
  - Gödel's Results
  - Completeness/Compactness

# Takeaways

- Rocq's "Limitations"
- No Sandboxing
- Acknowledging Previous Efforts

