

Project Description

Andreas Salhus Bakseter

January 17, 2023

1 Formalization of mathematical problems

When solving mathematical problems, one often uses proofs to assert some claim. Proofs can be written either *informally* or *formally*.

An informal proof is often written in a natural language, where the truth of the proof is determined by if the reader is convinced by the proof or not. ref (<https://softwarefoundations.cis.upenn.edu/lf-current/Induction.html#lab58>)

This can lead to mistakes in the proofs reasoning, as it is hard to be 100% certain that every step in the proof is solid (especially with larger ones). ref (marc slides)

2 Proof assistants

A formal proof can be written like a computer program, where all the arguments can be checked mechanically; often done using a proof assistant.

Coq is a proof assistant that enables us to write formal proofs and verify them. Coq uses type theory to verify proofs, but can also be used as a functional programming language. ref (<https://en.wikipedia.org/wiki/Coq>)

Other examples of proof assistants include Agda, Isabelle, Lean and HOL.

- 3 Type theory & the Curry-Howard correspondence
- 4 Our case
- 5 Approach & design choices
- 6 Implementation
- 7 Examples & results
- 8 Evaluation
- 9 Conclusion