The Evolution of Proof Assistant Math Repositories

Mahsa Bazzaz
Northeastern University
Boston, Massachusetts, USA

Minsung Cho
Northeastern University
Boston, Massachusetts, USA

Gwen Lincroft
Northeastern University
Boston, Massachusetts, USA

1 INTRODUCTION

Proof assistants (also known as interactive theorem provers) are programming languages with the primary purpose of providing computational meaning to mathematical proof through code. The first proof assistants, developed over fifty years ago, were designed to help reason about the correctness of software and hardware through mathematical abstraction. However, as proof assistants became more expressive and user-friendly, they have gained traction in numerous fields, including formal methods, artificial intelligence, and computer science education TODO: add citation..

However, over the past 20 years, proof assistants garnered attention from communities beyond computer science. In particular, the mathematics community took interests in proof assistants with their potential to bridge software development and mathematical proof. This potential has been realized through multiple open-source, well-documented libraries of formalized proofs as programs. Although there have been many such libraries over time, the predominant ones today are written in three languages: Lean, Coq, and Isabelle/HOL.

Lean has been the focal point of math library development in the past five years, gaining the attention of prominent mathematicians by demonstrating its ability to prove research-level mathematics TODO: add citation.. Its math library, known as `mathlib` , has been open-sourced on GitHub under an Apache license. It has extremely rigid conventions for contributing, including guidelines on style, naming, commit messages, and pull request labeling.

Coq is one of the most widely-used proof assistants in computer science today. However, its math library is not centralized like `mathlib` . Instead, it offers a *loosely federated* list of open-source GitHub repositories. Many of these repositories are under the same GitHub organizations, for example `math-comp` or `coq-community`, but not all of them are and their interdependence is not obvious.

Isabelle/HOL is another major proof assistant that offers an extremely strong general automation, which the other proof assistants lack. Although its math library is open source, it is not hosted natively on GitHub and external contribution is not done through GitHub's standard means. In particular, contribution is vetted behind closed doors by academics and is not done through a standard pull request-merge system. Furthermore, there is no centralized repository á la `mathlib` and most are hosted on the website *Archive of Formal Proofs*, where the proofs lack version control but give a detailed account of authors, dependencies, and proof techniques.

It is clear that each of these proof assistants and their math libraries have different software development practices. However, they all attempt to demonstrate the power of proof assistants by formalizing known mathematics through code. This project aims to compare and contrast the evolution of these three main repositories through a software engineering lens. In particular, for this project we have two main research questions:

(1) What mathematical theorems did different formalized mathematics libraries prove? How has this developed over time? How have the proofs changed?
(2) How has the popularity of contributing to theorem provers over time changed? Are there any factors we need to take into account to evaluate proof assistant software different than traditional software?

2 PROJECT RESULTS

3 PROJECT CHALLENGES