Formal Verification with F* and Meta-F*

Lecturers: Nikhil Swamy and Guido Martínez

Summary

We propose an introduction to F*, a language aimed at (semi-automatic) verification. It combines the power of dependent types with the ease of a SMT-backed program verifier. Proofs in F* usually go via an SMT solver, but the programmer can also build proof terms or use tactics. F* includes an extensible effect system using which one can verify the behaviour of effectful programs.

F* has been used successfully to both write verified programs and libraries (at varying levels of abstraction, down to assembly language), and to formalize theorems such as parts of the metatheory of F* itself. It is developed jointly by Microsoft Research and Inria, and is the main language for ""Project Everest"", a project aiming to build a fully verified HTTPS library. F* is freely available and open source, see https://www.fstar-lang.org/ and and https://project-everest.github.io/.

The course will start with the basics and move up to the use of tactics.

Brief Index

- Programming language design
- Dependently typed programming
- Effect systems
- Program verification
- Modeling and verifying effectful programs
- Proofs by reflection
- Metaprogramming
- Combining automated and interactive proof techniques

Required Background

Students do not need a background in formal verification or proof assistants, although it would likely help. Students should have a background in functional programming.