CAREER: Designing Formal Specifications

Research Questions

- How to translate informal specifications into formal, machine-checkable specifications? Minimize the human effort required.
- How to debug specifications? Need to determine whether they express the desired behavior, and whether the intended behavior has undesirable consequences.

Methods

- Logical modeling (alloy-style)
- Random testing, mutation testing
- LLM prompting in a feedback loop

Examples / Application Areas

- Internet protocols: translate design documents to logical models with domain-specific visualization.

  Inspiration: `https://datatracker.ietf.org/rg/ufmrg/about/`

  Related: Jest, compiling ECMAscript specification to an interpreter.
- UAV routing (local mentor Henderson): synthesize informal requirements from a variety of stakeholders (e.g., UAV manufacturers, city officials) into temporal logic specifications. Map solver errors to stakeholder constraints.
- Rhombus programming language (local mentor Flatt): when a new feature is implemented for the language prototype, generate tests that explore its interactions with existing features. Look for “surprising” combinations.

  Related: Java unsoundness due to null and generics.
  Modula 2 confusion with implicit coercions.

Abstract

Formal specifications are a prerequisite for correct software, yet programmers ofter choose not to write them because tool support for designing specification lags far behind tools for designing code. Specifications can be wrong; tools must embrace this fact and encourage exploration and debugging. This proposal aims to uncover scientific principles to support the nimble design of formal specifications. It focuses on three target areas: type specifications for untyped programs, logic specifications for uncrewed aircraft routes, and models for reactive systems. The third domain will serve as the focus for outreach efforts. The PI will develop on online environment for system modeling and deploy it in an undergraduate course that gives students hands-on experience writing specifications and in K-12 activities through the University of Utah GREAT summer camp program. To engage students, the online environment will frame specification as a game between the specifier (student) and an adversary who invents a wrong system while obeying all rules from the specification.