Applications of Lightweight Formal Methods in Engineering and Scientific Software Design

Tristan Dyer

May 21, 2019

Abstract

Boop.

1 Introduction/Background

Called a third pillar of science, computation is an indispensible tool for scientists and engineers who simulate physical and natural processes. Recent studies on reliability, reproducibility of results, and productivity have brought forth concern that existing practices of constructing scientific software are inadequate and limiting the pace of technological advancement. A disconnect between existing modern software engineering practice and scientific computation has become apparent and must be addressed. Additionally, the unique challenges facing developers of scientific software, namely the lack of test oracles, software lifetimes and evolving needs that span decades, and the competing objectives of performance, maintainability, and portability, must also be recognized.

I seek to address fundamental design and quality assurance challenges that are intrinsic to scientific computation and engineering software design. While numerous directions might be taken, my premise and motivating viewpoint is the central role that modeling can and must play in the process of designing and working with scientific programs. Culturally, the fit may be a natural

one: scientists and engineers are accustomed to working with models anyway, and with the kind of automatic, push-button analysis supported by some state-based formalisms, those who develop software can focus on modeling and design instead of theorem proving.

Although the tools and techniques most identified with scientific computation are those of numerical analysis—where error prediction, stability, and convergence are central concerns—such an enterprise offers little guidance in the development process, where early decisions about decomposition and organization establish program structure. I suggest an approach that separates concerns: isolating the structural and behavioral components from the numerics, allowing scientists and engineers to more effectively reason about the programs they create. The approach is well-suited for lightweight tools like Alloy [**?**], a state-based formalism that combines declarative modeling and bounded model checking.

2 Problem Statement

The goal of this research is to establish the role of lightweight formal methods as an invaluable tool in bridging the gap between modern software engineering practices and the unique challenges presented by scientific and engineering software. This involves the following specific objectives:

1. To clearly define common paradigms of lightweight formal methods within the context of scientific software development. These include but are not limited to: specification, verification and validation, correctness, refinement, and predicate abstraction.

2. To demonstrate the utility of lightweight formal methods through the development of actual models and their respective software. These examples will reflect concepts commonly found in scientific and engineering software, and will span a broad range of abstractions in order to demonstrate the

utility and applicability of the approach. Examples are to include the moment distribution method, sparse matrix multiplication, the finite element method, web-based analysis tools, and user interface design.

3. To develop tools that aid scientists and engineers in applying these methods in practice. These tools include the a web-based domain specific visualization tool for use in the modeling of finite element software, the development of a model sharing utility that leverages version control and the visualization tool, and the embedding of the visualization tool and the Alloy analyzer into an existing IDE (integrated development environment).

3 Literature Review

Daniel Jackson.

4 Methodology

Find common programming paradigms in scientific software design and model them.

5 Overview of Chapters

- Moment distribution
- Sparse matrices
- Finite element
- User interface design
- Mesh editing and rendering
- Utilities for Alloy

6 Plan of Work

Get it all done this year.

7 Bibliography

Citations.

8 Appendices

Papers we've already written.