Rigorous Concurrency Analysis of Multithreaded Programs

Yue Yang, Ganesh Gopalakrishnan,
and Gary Lindstrom

UUCS-03-026

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

November 18, 2003

Abstract

This paper explores the practicality of conducting program analysis for multithreaded software using constraint solving. By precisely defining the underlying memory consistency rules in addition to the intra-thread program semantics, our approach offers a unique advantage for program verification — it provides an *accurate* and *exhaustive* coverage of all thread interleavings for any given memory model. We demonstrate how this can be achieved by formalizing sequential consistency for a source language that supports control branches and a monitor-style mutual exclusion mechanism. We then discuss how to formulate programmer expectations as constraints and propose three concrete applications of this approach: execution validation, race detection, and atomicity analysis. Finally, we describe the implementation of a formal analysis tool using constraint logic programming, with promising initial results for reasoning about small but non-trivial concurrent programs.