Debugging Concurrent Programs

Vineet Kahlon
NEC Laboratories - Princeton, NJ

1. ABSTRACT

Detecting errors in concurrent programs is a notoriously difficult task. A key reason for this is the behavioral complexity resulting from the large number of interleavings of different threads. This exacerbates the state-explosion problem thereby rendering a full-fledged state space exploration of concurrent programs hard.

Traditionally, verification of concurrent software has focused on using explicit-state model checking techniques for state-space exploration while relying on partial order reduction (POR) to isolate a subset of thread interleavings that need be considered without sacrificing precision. These techniques, however, do not exploit the power of symbolic methods, i.e., the use of of SAT/SMT solvers, to efficiently explore large state spaces. We discuss recent advances in integrating POR with symbolic search which then enables us to leverage the power of both techniques in combating state explosion.

While symbolic-POR techniques go a long way in ameliorating the state explosion problem, they still do not enable a full-fledged exploration of states paces of real-life concurrent programs. We next discuss, trace-based or runtime error detection techniques have been gaining in popularity in recent years. These come in many variants. Runtime monitoring aims at identifying violations exposed by a given execution trace. However, due to the large number of interleavings of the program, triggering a concurrency bug by exploring just one interleaving is unlikely. In contrast, runtime prediction aims at detecting violations in all feasible interleavings of events of the given trace. Predictive analysis seems to offer a good compromise between runtime monitoring and full-edged model checking in that it guarantees better coverage than runtime monitoring but mitigates the state explosion inherent in model checking.

We next focus on dataflow analysis which is an effective and indispensable technique for analyzing large-scale real-life sequential programs. In recent years, Pushdown Systems (PDS) has emerged as a powerful, unifying framework for efficiently encoding inter-procedural dataflow analyses. In this framework, dataflow analysis can be cast as a PDS model checking problem. Sequential program dataflow analysis then exploits the fact that PDS model checking is efficiently decidable for very expressive classes of temporal properties. Analogous to the sequential domain, dataflow analysis for concurrent programs reduces to the problem of model checking multi-PDS systems which, unfortunately, is undecidable even for simple reachability properties. However, we discuss state-of-the-art results which show that by exploiting frequently-used programming patterns efficient decidability can be achieved in practice.

We close the talk by briefly demonstrating how synergistic combinations of these techniques can be exploited for effectively debugging real-life concurrent programs.

2. SPEAKER BIOGRAPHY

Vineet Kahlon is a Research Staff Member at NEC Laboratories in Princeton, NJ. He received his BTech degree (1997) from Indian Institute of Technology, Kanpur, (India) and his M.S. (1999) and PhD. (2004) degrees from the University of Texas at Austin where he was awarded the MCD fellowship. He joined NEC in August 2004. Vineet Kahlon's research interests include Automated Verification, Software Engineering, Programming Languages, and Embedded Systems with a focus on the analysis of concurrent programs. He has made important contributions to model checking, dataflow analysis and predictive analysis of concurrent programs many of which have appeared in top conferences in the field. At the University of Texas, Vineet Kahlon developed new techniques for reasoning about parameterized and infinite state systems. At NEC Labs, he has contributed to the development of a software verification tool called F-Soft and a concurrent software debugging tool called CoBe. He is the recipient of NEC Laboratories SEEDS and Technology Commercialization Awards. Vineet Kahlon has served on program committees of leading conferences and also co-organized the EC2 workshops on Exploiting Concurrency Efficiently and Correctly in 2008-2009.

`http://www.cs.utexas.edu/~kahlon`