

Project Proposal

When developers need to circumvent the problems of multithreading, thread-safe libraries can represent a considerable help. Finding atomicity violations in this libraries when methods are invoked is crucial in order to take first steps towards guaranteeing thread-safety. The tool INTRUDER, discussed in the paper [1], synthesize multithreaded tests by running the library methods using a sequential seed test-suite with random parameters. The atomicity violations are exposed after invoking concurrently the methods that are identified by constructing a set of three interleaved accesses.

CONTEGE, a similar tool discussed in the paper [2] introduces a brute-force approach which is less efficient because the test cases generated are very large and the defect revealing is low compared to INTRUDER [1]. On the other hand, the tool NARADA discussed on [3] represents a directed approach to synthesize multithreaded test to enable detection of race conditions but a race detector is sufficient to identify atomicity violations.

In this project, instead of using sequential or pure random tests, we propose to test this framework (INTRUDER) using feedback directed random tests as an input. With this approach, we are trying to improve the coverage and, therefore, to expose methods that were not evaluated using the sequential approach.

We are planning to evaluate the accuracy and performance of our approach by implementing INTRUDER and reproducing the experiments performed in [1] only changing the test-suite and then comparing the results. To generate the feedback directed random test-suits, we are going to use RANDOOP.

References

- [1] Samak, M., & Ramanathan, M. K. *Synthesizing Tests for Detecting Atomicity Violations*. Bergamo, Italy: ESEC/FSE 2015.
- [2] M. Pradel and T. R. Gross. Fully automatic and precise detection of thread safety violations. In Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '12, 2012
- [3] M. Samak, M. K. Ramanathan, and S. Jagannathan. Synthesizing racy tests. In Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2015