Schemmel, Daniel; Büning, Julian; Rodríguez, César; Laprell, David; Wehrle, Klaus
Symbolic partial-order execution for testing multi-threaded programs.  (English)


Summary: We describe a technique for systematic testing of multi-threaded programs. We combine Quasi-Optimal Partial-Order Reduction, a state-of-the-art technique that tackles path explosion due to interleaving non-determinism, with symbolic execution to handle data non-determinism. Our technique iteratively and exhaustively finds all executions of the program. It represents program executions using partial orders and finds the next execution using an underlying unfolding semantics. We avoid the exploration of redundant program traces using cutoff events. We implemented our technique as an extension of KLEE and evaluated it on a set of large multi-threaded C programs. Our experiments found several previously undiscovered bugs and undefined behaviors in memcached and GNU sort, showing that the new method is capable of finding bugs in industrial-size benchmarks.

For the entire collection see [Zbl 1453.68017].

MSC:

68N19 Other programming paradigms (object-oriented, sequential, concurrent, automatic, etc.)

68Q60 Specification and verification (program logics, model checking, etc.)

Keywords:

software testing; symbolic execution; partial-order reduction

Full Text: DOI arXiv