

Capturing Concurrency Aspects of Software Under Development to Reduce Testing Effort

Writing tests to check for data races in concurrent software is challenging due to the exponential growth of possible interleavings as code size and thread count increase. The Eraser-Lockset algorithm can detect a subset of races given a trace of accessed memory locations, but running the entire test suite after every change to generate these traces can be slow and, especially for small changes, unnecessary.

By exploiting the knowledge that the previous iteration found no data races, this project aims to reduce both execution time and the size of newly generated traces by culling redundant tests.