

Smart concurrent test scenarios generation

Currently, Lin-Check tests are generated as a random set of operations chosen from a user-defined set. If an operation has some parameter then by default its value is chosen randomly from a specified range. For many structures, like a hash-table, a set or alike, random choice of key-parameters is not optimal to generate "interesting" scenarios. Now it's handled with narrowing the parameter value range to increase the probability of repeated values appearing. It seems that structure behavior analysis could automatically detect operations that will affect each other to produce concurrent test scenarios that more sensitive for interleaving.

Related works:

- O. Shacham et al., Testing atomicity of composed concurrent operations. ACM SIGPLAN Notices. 46, 51 (2011) [COLT] – <http://theory.stanford.edu/~aiken/publications/papers/oopsla11b.pdf>
Related presentation: https://www.slideserve.com/Pat_Xavi/colt-effective-testing-of-concurrent-collection-usage
- Michael Pradel and Thomas R. Gross. 2012. Fully automatic and precise detection of thread safety violations. *SIGPLAN Not.* 47, 6 (June 2012), 521-530. – <http://mp.binaervarianz.de/pldi2012.pdf>
- Valerio Terragni and Shing-Chi Cheung. 2016. Coverage-driven test code generation for concurrent classes. In *Proceedings of ICSE '16*, 1121-1132 – http://www.cse.ust.hk/~vterragni/files/Terragni_ICSE2016.pdf
- Steenbuck, Sebastian, and Gordon Fraser. "Generating unit tests for concurrent classes." *Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on*. IEEE, 2013. <https://pdfs.semanticscholar.org/b4a8/e794a4056e6d71d9b2e8c78446775f97989b.pdf>