

Reference

- [1] S. Anand, E. K. Burke, T. Y. Chen, J. Clark, M. B. Cohen, W. Grieskamp, M. Harman, M. J. Harrold, and P. McMinn, “An orchestrated survey of methodologies for automated software test case generation,” Aug. 2013.
- [2] H. Yoshida, S. Tokumoto, M. R. Prasad, I. Ghosh, T. Uehara, “FSX: Fine-grained Incremental Unit Test Generation for C/C++ Programs”, Proceedings of the 25th International Symposium on Software Testing and Analysis ser. ISSTA 2016, pp. 106-117, 2016.
- [3] J. Burnim, K. Sen, “Heuristics for scalable dynamic test generation”, in ASE '08, 2008.
- [4] G. Fraser, A. Arcuri, “Whole test suite generation”, IEEE Trans. Softw. Eng., vol. 39, no. 2, pp. 276-291, Feb. 2013.
- [5] S. J. Galler, B. K. Aichernig, “Survey on test data generation tools”, STTT, vol. 16, no. 6, pp. 727-751, 2014.
- [6] H. Tanida, T. Uehara, G. Li, I. Ghosh, “Automatic unit test generation and execution for javascript program through symbolic execution,” Proceedings of the Ninth International Conference on Software Engineering Advances, pp. 259-265, 2014.
- [7] X. Deng, Robby, J. Hatcliff, Kiasan/KUnit, “Automatic Test Case Generation and Analysis Feedback for Open Object-oriented Systems,” in Proc. of MUTATION 2007, IEEE CS, 2007.
- [8] Tassey, G, “The economic impacts of inadequate infrastructure for software testing. National Institute of Standards and Technology,” RTI Project Number 7007.011, 2002.
- [9] Hong Zhu, Patrick A. V. Hall, John H. R. May, “Software unit test coverage and adequacy,” ACM Computing Surveys, 29(4):366–427, 1997.
- [10] M. Ellims, J. Bridges, D. C. Ince, “The economics of unit testing”, Empir. Softw. Eng., vol. 11, no. 1, pp. 5-31, 2006.

- [11] Paul Ammann, Jeff Offutt, “Introduction to Software Testing,” Cambridge University Press, New York, NY, USA, 2008.
- [12] G. Li, I. Ghosh, and S. P. Rajan, “KLOVER: A symbolic execution and automatic test generation tool for C++ programs,” in CAV, 2011.
- [13] Li, G., Ghosh, I., Rajan, S. P., Gopalakrishnan, G. GKLEE, “Concolic Verification and Test Generation for GPUs,” 2012.
- [14] P. Garg et al., “Feedback-Directed Unit Test Generation for C/C++ Using Concolic Execution”, Proc. 35th Int'l Conf. Software Eng. (ICSE 13), pp. 132-141, 2013.
- [15] Jacob Burnim, Koushik Sen, “Heuristics for Scalable Dynamic Test Generation” in ASE '08, 2008.
- [16] Olivier Crameri, RekhaBachwani, Tim Brecht, Ricardo Bianchini, DejanKostic, Willy Zwaenepoel, “Concolic Execution Driven by Test Suites and Code Modifications,” 2009.
- [17] Y Kim, M Kim, YJ Kim, Y Jang, “Industrial Application of Concolic Testing Approach,” A Case Study on libexif by Using CREST-BV and KLEE, 2012.
- [18] Cristian Cadar, Daniel Dunbar, Dawson Engler, “Klee: Unassisted and automatic generation of high-coverage tests for complex systems programs”, Proceedings of the USENIX Symposium on Operating System Design and Implementation, 2008.
- [19] Pietro Braione, Giovanni Denaro, Andrea Mattavelli, MattiaVivanti, Ali Muhammad, “Software testing with code-based test generators: data and lessons learned from a case study with an industrial software component,” Software Qual J, vol. 22, no. 2, pp. 311-333, 2014.
- [20] Duc Anh, Nguyen & Ngoc Hung, Pham, “A Test Data Generation Method for C/C++ Projects,” 2017.
- [21] Patrice Godefroid, Nils Klarlund, and Koushik Sen, “DART: Directed Automated Random Testing,” In Proceedings of the 2005 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '05). ACM, New York, NY, USA, 2005.

- [22] Koushik Sen, Darko Marinov, and Gul Agha, “CUTE: A Concolic Unit Testing Engine for C”, In Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE-13). ACM, New York, NY, USA, 2005.
- [23] Nicky Williams, Bruno Marre, Patricia Mouy, and Muriel Roger, “PathCrawler: Automatic Generation of Path Tests by Combining Static and Dynamic Analysis”, In Proceedings of the 5th European Conference on Dependable Computing (EDCC’05). Springer-Verlag, Berlin, Heidelberg, 2005.
- [24] McMinn, Phil, “Search-based software test data generation”, a survey: Research Articles. *Softw. Test*, 2004.
- [25] Runeson, Per. (2006). A Survey of Unit Testing Practices. *IEEE Software*. 23. 10.1109/MS.2006.91.
- [26] Kassab, Mohamad & DeFranco, Joanna & Laplante, Phillip. (2017). Software Testing: The State of the Practice. *IEEE Software*. 34. 46-52. 10.1109/MS.2017.3571582.
- [27] Thummalapenta, Suresh & Xie, Tao & Tillmann, Nikolai & Halleux, Jonathan & Schulte, Wolfram, “MSeqGen: Object-Oriented Unit-Test Generation via Mining Source Code.” 2009.
- [28] Jaygarl, Hojun. “Capture-based Automated Test Input Generation”, 2010.
- [29] Yatoh, Kohsuke & Sakamoto, Kazunori & Ishikawa, Fuyuki & Honiden, Shinichi, “Feedback-Controlled Random Test Generation”, 2015.
- [30] Cseppentő, Lajos & Micskei, Zoltan. “Evaluating code-based test input generator tools. *Software Testing, Verification and Reliability*”, 2017.
- [31] Zhang, Chengyu & Su, Ting & Yan, Yichen & Wu, Ke & Pu, Geguang. “Towards Efficient Data-flow Test Data Generation Using KLEE”, 2018.