

An Efficient Method for Automatic Generation of Linearly Independent Paths in White-box Testing

Xinyang Wang¹, Yaqiu Jiang², Wenhong Tian^{3*}

School of Software and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

Received 08 April 2014; received in revised form on 07 April 2015; accepted on 08 April 2015

Abstract

Testing is the one of most significant quality assurance measures for software. It has been shown that the software testing is one of the most critical and important phases in the life cycle of software engineering. In general, software testing takes around 40-60% of the effort, time and cost. Structure-oriented test methods define test cases on the basis of the internal program structures and are widely used. Path-based test is one of the important Structure-oriented test methods during software development. However, there is still lack of automatic and highly efficient tool for generating basic paths in white-box testing. In view of this, an automatic and efficient method for generating basic paths is proposed in this paper. This method firstly transforms the source-code program into corresponding control flow graph (CFG). By modifying the original CFG to a strongly connected graph, a new algorithm (ABPC) is designed to automatically construct all basic paths. The ABPC algorithm has computational complexity linear to the number of total edges and nodes in the CFG. Through performance evaluation of many examples, it is shown that the proposed method is correct and scalable to very large test cases. The proposed method can be applied to basis path testing easily.

Keywords: White box test; Path-Based Test; Control flow graph; Complexity Measure; Elementary circuit.

References

- [1] D. Berndt, J. Fisher, L. Joshson, "Breeding Software Test Cases with Genetic Algorithms," Proceedings of the 36th Hawaii International Conference on System Sciences, Big Island, Hawaii, USA, 2003.
- [2] D. J. Berndt, A. Watkins, "Investigating the Performance of Genetic Algorithms-Based Software Test Case Generation," Proceedings of the Eighth IEEE International Symposium on High Assurance Systems Engineering ,Tampa Florida, USA, pp. 261-262, 2004.
- [3] A. G. Holzman, A. Kentand, J. Belzer, Encyclopedia of Computer Science and Technology, CRC Press, pp. 36-367, 1992.
- [4] Y. Chen, Z. Li, H. Jin, J. He, "Control Flow Paths Subset of Tested Program Generation Algorithm Based on LCC," Computer Engineering, vol. 35, no. 7, pp. 39-41, May 2009. (in Chinese)
- [5] Q. F. Du, D. Xiao, "An improved algorithm for basis path testing," the Proceedings of 2011 International Conference on Business Management and Electronic Information, vol. 3, pp. 175-178, 2011.
- [6] A. Ehrenfeucht, L. Fosdick, L. Osterweil, An algorithm for finding the elementary circuits of a directed graph, Tech. Rep. CU-CS-024-23, Department of Computer Science, University of Colorado, Boulder, 1973.
- [7] M. Evangelist, "An Analysis of Control Flow Complexity," Eighth International Computer Software and Application Conference, pp. 328-396, 1984.
- [8] F. Harary, E. Palmer, Graphical Enumeration, Academic Press, New York, 1973.
- [9] D. B. Johnson, "Finding All the Elementary Circuits of A Directed Graph.SIAM Journal on Computing," vol. 4, no.1, pp. 77-84, Mar. 1975.
- [10] B. F. Jones, H. H Sthamer, D. E. Eyres, "Automatic Structural Testing Using Genetic Algorithms," Software Engineering Journal, vol. 11, no. 5, pp. 299-306, 1996.
- [11] A. G. Li, "Automatic Generating All-Path Test Data of a Program Based on PSO," WRI World Congress on Software Engineering (WCSE '09), vol. 4, May 2009.

- [12] J. Lin, P. Yeh., "Using Genetic Algorithms for Test Case Generation in Path Testing," Proceedings of the Ninth Asian on Test Symposium, Asian, pp. 241-246, 2000.
- [13] T. McCabe, "A Complexity Measure," IEEE Transactions on Software Engineering, vol. SE-2, no. 4, pp. 308-320, Dec. 1976.
- [14] G. Myers, "An Extension to the Cyclomatic Measure of Program Complexity," SIGPLAN Notices, Oct. 1977.
- [15] T. McCabe, "Structured Testing: A Software Testing Methodology Using the Cyclomatic Complexity Measure", NBS Special Publication, pp. 99-500, Dec. 1982.
- [16] G. Myers, The Art of Software Testing. 2ed. John Wiley & Son. pp. 234, 2004
- [17] J. Poole, "A Method to Determine a Basis Set of Paths to Perform Program Testing," NISTIR 5737, Nov. 1995.
- [18] M. Prasanna, S. N. Sivanandam, R. Venkatesan, R. Sundarajan, "A Survey on Automatic Test Case Generation," Academic Open Internet Journal, 2005.
- [19] J. C. Tiernan, An efficient search algorithm to find the elementary circuits of a graph, Comm. ACM, pp. 722-726, 1970.
- [20] R. Tarjan, "Depth-first search and linear graph algorithms," SIAM Journal on Computing, vol. 1, no. 2, pp. 60-146, June 1972.
- [21] R. Tarjan, "Enumeration of the elementary circuits of a directed graph, SIAM Journal on Computing," vol. 2, no. 3, pp. 16-211, Sept. 1973.
- [22] L. J. White, B. Wiszniewski, "Path Testing of Computer Programs with Loops using a Tool for Simple Loop Patterns," Software - Practice and Experience, vol. 21, no. 10, pp. 102-1075, Oct. 1991.
- [23] J. Wegener, A. Baresel, H. Sthamer, "Evolutionary Test Environment for Automatic Structural Testing," Information and Software Technology, vol. 43, no. 14, pp. 841-854, 2001.
- [24] H. Weinblax, "A new search algorithm for finding the simple cycles of a finite directed graph," Journal of the Association for Computing Machinery, vol. 19, no. 1, pp. 43-56, 1972.
- [25] W. Xin, J. W. Cao, "Computing Strongly Connected Components Using Kosaraju algorithm," vol. 31, no. 4, pp. 05-691, Feb. 2010.
- [26] W. Xin, J. W. Cao, "Computer Engineering and Design," vol. 31, no. 4, pp. 05-691, Oct. 2007.
- [27] J. Yan, J. Zheng, "An efficient method to generate feasible paths for basis path testing," Information Processing Letters, vol. 107, no. 3-4, pp. 87-92, Jul. 2008.
- [28] G. Plugins, "Control Flow Graph Factory," <http://www.drgarbage.com/control-flow-graph-factory.html>, 2008.
- [29] S. Zhang, "Automatic Path Test Data Generation Based on GA-PSO," In 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), Oct. 2010.
- [30] A. S. Ghiduk, O. Said, S. Aljahdali, Basis Test Paths Generation Using Genetic Algorithm, Proceedings of ICCIT, 2012.
- [31] M. Papadakis, N. Malevris, "An Effective Path Selection Strategy for Mutation Testing," APSEC 2009, pp. 422-429, 2009
- [32] M. Papadakis, N. Malevris, "Mutation based test case generation via a path selection strategy," Information & Software Technology, vol. 54, no. 9, pp. 915-932, 2012.
- [33] M. Papadakis, N. Malevris, "A Symbolic Execution Tool Based on the Elimination of Infeasible Paths," ICSEA 2010, pp. 435-440, 2010.
- [34] M. N. Ngo, H. B. Tan, "Heuristics-based infeasible path detection for dynamic test data generation," Information & Software Technology, vol. 50, no. 7-8, pp. 641-655, 2008