Code-aware combinatorial interaction testing

Bestoun S. Ahmed [1, 2], Angelo Gargantini [3], Kamal Z. Zamli [4], Cemal Yilmaz [5], Miroslav Bures [2], Marek Szeles [2]

[1] Department of Mathematics and Computer Science, Karlstad University, 651 88 Karlstad, Sweden

[2] Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University, Karlovo nam. 13, Prague, Czech Republic

[3] Department of Management, Information and Production Engineering, University of Bergamo, Bergamo , Italy

[4] Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, Pahang, Malaysia

[5]Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey

ABSTRACT

Combinatorial interaction testing (CIT) is a useful testing technique to address the interaction of input parameters in software systems. CIT has been used as a systematic technique to sample the enormous test possibilities. Most of the research activities focused on the generation of CIT test suites as a computationally complex problem. Less effort has been paid for the application of CIT. To apply CIT, practitioners must identify the input parameters for the Software-under-test (SUT), feed these parameters to the CIT test generation tool, and then run those tests on the application with some pass and fail criteria for verification. Using this approach, CIT is used as a black-box testing technique without knowing the effect of the internal code. Although useful, practically, not all the parameters having the same impact on the SUT. This paper introduces a different approach to use the CIT as a gray-box testing technique by considering the internal code structure of the SUT to know the impact of each input parameter and thus use this impact in the test generation stage. The case studies results showed that this approach would help to detect new faults as compared to the equal impact parameter approach.

KEYWORDS

Program testing; Software fault tolerance; Program verification

REFERENCES

1. Kuhn, R., Kacker, R., Lei, Y., et al
   'Combinatorial software testing', Computer, 2009, **42**, (8), pp. 94–96.

2. Yilmaz, C.
   'Test case-aware combinatorial interaction testing', IEEE Trans. Softw. Eng., 2013, **39**, (5), pp. 684–706.

3. Tzoref-Brill, R.
   'Chapter two – advances in combinatorial testing', in Memon, A.M. (Ed.): 'Ser. Advances in computers', vol. **112** (Elsevier, Amsterdam, Netherlands, 2019), pp. 79–134.

4. Simos, D.E., Zivanovic, J., Leithner, M.
   'Automated combinatorial testing for detecting SQL vulnerabilities in web applications'. Proc. 14th Int. Workshop on Automation of Software Test, ser. AST '19, Montreal, Canada, 2019, pp. 55–61.

5. Ahmed, B.S., Abdulsamad, T.S., Potrus, M.Y.
   'Achievement of minimized combinatorial test suite for configuration-aware software functional testing using the cuckoo search algorithm', Inf. Softw. Technol., 2015, **66**, pp. 13–29.