ACTS - Advanced Combinatorial Testing System

Michael Wagner, Manuel Leithner
SBA Research
A-1040 Vienna, Austria
{mwagner,mleithner}@sba-research.org

Andrea Bombarda
Department of Engineering,
University of Bergamo,
Bergamo, Italy
Email: andrea.bombarda@unibg.it

***Abstract*—This document briefly describes the tool ACTS [1]. As the Competition Committee, we would like to thank Yu Lei as well as the other authors of the ACTS for allowing us to use their tool as a point of reference to compare the competitors with.**

***Index Terms*—Covering Array, Combinatorial Testing, In-Parameter-Order, Algorithms**

I. Tool overview

ACTS barely needs any introductions for CT practitioners. It is (one of) the most used and most well tested combinatorial test generation tools, implements many different algorithms, constraint solving techniques and various other features. It's algorithms are mostly based on the In-Parameter-Order strategy, which was first introduced by the tool's authors in 1998 [2]. ACTS is available upon demand at https://csrc.nist.gov/Projects/automated-combinatorial-testing-for-software/acts-library. Further, the documentation of the tool can be found online at [1]

II. IPO algorithms and constraint handling

Starting from an initial CA, the IPO strategy [2] builds a combinatorial test sets by adding columns one at a time and appending rows to cover any uncovered interactions if necessary. ACTS implements multiple different greedy IPO algorithms, such as IPOG, IPOG-F and IPOG-F2 [3]. The three different algorithms only differ in how the horizontal extension is implemented. IPOG is the fastest of the three and iterates the rows from top to bottom. IPOG-F and IPOG-F2 both further optimize the order in which rows are iterated by applying dynamic programming techniques and a heuristic respectively. In our experiments for the competition, we used the default algorithm, which is the IPOG algorithm.

III. Constraint handling

ACTS also implements various different constraint handling techniques, ranging from Minimal Forbidden Tuples (MFTs), Necessary Forbidden Tuples (NFTs) to CSP solvers, all of which are capable of handling all different instances provided in the CT competition benchmarks. This allows users to select the constraint handling method that best suits their needs. For the competition we again went with the default settings, which is the MFT approach.

[1] https://csrc.nist.gov/CSRC/media/Projects/Automated-Combinatorial-Testing-for-Software/documents/acts_user_guide_2_92.pdf

IV. Input and Output formats

Since ACTS supports the ACTS input format and can output the constructed test set in CSV form, using the tool for our competition layout required minimal effort.

References

[1] L. Yu, Y. Lei, R. N. Kacker, and D. R. Kuhn, "Acts: A combinatorial test generation tool," in *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, 2013, pp. 370–375.

[2] Y. Lei and K. C. Tai, "In-parameter-order: a test generation strategy for pairwise testing," in *Proceedings Third IEEE International High-Assurance Systems Engineering Symposium (Cat. No.98EX231)*, Nov 1998, pp. 254–261.

[3] M. Forbes, J. Lawrence, Y. Lei, R. N. Kacker, and D. R. Kuhn, "Refining the in-parameter-order strategy for constructing covering arrays," *Journal of Research of the National Institute of Standards and Technology*, vol. 113, no. 5, p. 287, 2008.