National Aeronautics and Space Administration

Combinatorial Testing Is Super-humanly Efficient And Thorough; Let's Make It Responsive And Intelligible Too (User-Friendly)

Principal Investigator: Marcel Schoppers (348); Co-Investigators: Anthony Barrett (393)

Program: FY21 R&TD Innovative Spontaneous Concepts

Objective: Beginning with a Combinatorial Testing (CT) tool, inherit its extreme efficiency and thoroughness, but address the CT field's chronic un-concern for the human interface and project complexities, e.g. using multiple testing venues, changing hardware and software, admitting dependencies between variables, injecting faults, debugging test-failures, not-repeating known failures, re-testing fixes, etc. The result will be a tool enabling non-specialists to become super-human testers.

Background: Combinatorial Testing (CT) is far more thorough, and efficient, at discovering bugs, than expert systems-engineers; more thorough even than the FAA's rigorous MC/DC standard; much more efficient than randomized testing. Perhaps you heard of the fatal crash of a self-driving Tesla, oblivious to a 4-values bug (white truck, bright sky, truck height, and truck angle); nothing but CT can guarantee to catch such bugs. However, the field's intense focus on minimums, in test-campaign size, and in generator-algorithm speed, have led to a severe lack of user-friendliness. Wrapping a CT program with machine learning algorithm C5.0 looked like it would solve those problems.

Approach and Results: *We had 12 complaints about the un-friendliness of CT tools. We now know how to solve all but 4 of those complaints!*

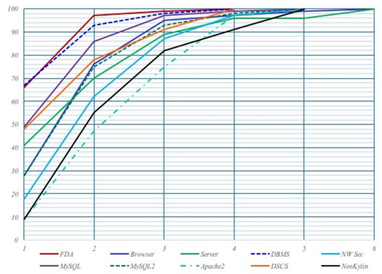
Machine Learning algorithm C5.0 couldn't be used as we proposed, for 2 unexpected reasons: #1 The "training sets" C5.0 would need – listing all the combinations yet to be tested – would be immense; #2 CT test-generators pack as many test-combinations into as few test-vectors as possible, thus making the diagnosis problem as difficult as possible, for both humans and Machine Learning. A careful mathematical analysis showed that in realistic CT domains, C5.0 will find up-to-hundreds of incorrect diagnoses.

We solved 8 of our 12 complaints, via (see the Figure at right)

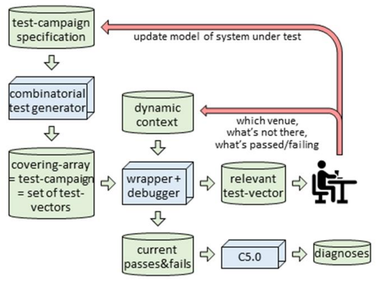
#a An "outer loop" executed infrequently, to describe the system-under-test and the test-campaign desired. The key is a clever CT-generator program that provides a small specification language for the test-campaign.

#b An "inner loop" executed on a daily basis, as shown. The wrapper is simple, the debugger is not. We investigated several debugging algorithms, and implemented one, but we're not "sold".

#c Retaining C5.0 as an on-demand capability, useful *after* debugging the failures, to produce a summary diagnosis.



(Vertical axis:) % of bugs caught as a function of (horizontal axis:) "testing strength" = number of variables interacting = combination size t. "Code coverage" is roughly comparable to t=1.
(This figure is online at NIST.)



How we expect the tool to work: An outer loop for test-campaign re-generation, an inner loop for day-to-day testing, and C5.0 *after* debugging the failures, to produce a summary diagnosis, as a decision tree or set of rules.

Significance/Benefits to JPL and NASA:

#1 Our resulting software-tool would make JPL's and NASA's testing-campaigns much more efficient and thorough than they are today [Fifo 2019] – if we can see a way to satisfy our relatively-weak requirements with these super-humanly-strong testing-campaigns!

#2 The literature has already shown that CT-campaigns are even stronger than the FAA's strong MC/DC software testing standard [Vilkomir 2017]. In future work, we hope to show that our tool can satisfy MC/DC with reduced cost. If that succeeds, this work would be important to the aviation industry, and to the public.

Publications: None yet.

References:

Miraldi Fifo, Eduard Enoiu, Wasif Afzal, "On Measuring Combinatorial Coverage of Manually Created Test Cases for Industrial Software," *IEEE International Conference on Software Testing, Verification and Validation Workshops* (22 April 2019, in Xi'an China): pp. 264–267.

D. Richard Kuhn, Dolores Wallace, Albert Gallo Jr, "Software Fault Interactions and Implications for Software Testing," *IEEE Transactions on Software Engineering* **30#6** (2004): pp. 418–421.

Sergiy Vilkomir, Aparna Alluri, D. Richard Kuhn, Raghu Kacker, "Combinatorial and MC/DC Coverage Levels of Random Testing" *IEEE International Conference on Software Quality, Reliability and Security* (25 July 2017 in Prague, Czech Republic): pp. 61–68.

National Aeronautics and Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

www.nasa.gov

PI/Task Mgr Contact
Email: Marcel.J.Schoppers@jpl.nasa.gov

Clearance Number:
RPC/JPL Task Number: R21230