COST-EFFECTIVENESS EVALUATION OF TEST CASE GENERATION VIA MODEL CHECKING

This briefing reports scientific evidence on the cost-effectiveness of Specification Patterns System (SPS) patterns/pattern scopes to generate test cases via Model Checking within the HiMoST method.

FINDINGS

→ The empirical evaluation reported here aims at assessing the cost and effectiveness of test suites generated via the following SPS patterns/pattern scopes: Absence/Global (ABS), Response Chain/Global (REC), Precedence Chain (P prec. S, T)/Global (PC1), Precedence Chain (S, T prec. P)/Global (PC2).

→ The empirical evaluation considers cost as the normalized average number of test steps per test case, and effectiveness as three measures: the average value of the proportion of transitions covered in the Statecharts, the average value of covered instructions and the average value covered branches of the code.

→ ABS presented the best cost followed by REC and in the last position is PC1.

→ Surprisingly, PC2 generated no test case for all samples. In fact, it did generate counterexamples (test cases) for all Computation Tree Logic (CTL) properties for all case studies but they were all discarded, because these counterexamples had only 1 state (the initial state) and thus less than a defined threshold.

→ ABS created a significant number of test cases which were discarded while REC showed a proportionally huge number of repeated test cases. These findings suggest that, in accordance with the HiMoST method, these patterns can contribute, albeit marginally, to increasing the cost of the testing process as a whole.

→ Still under the cost perspective, PC1 presented more satisfied (*true*) CTL formulas. Thus, it is not necessary to perform any processing/verification of repeated or discarded test cases for such formulas that were satisfied.

→ Considering the three measures of effectiveness, PC1 was the best solution followed by REC, and ABS was the worst. This is precisely the inverse order of the cost analysis. In some samples, PC1 even achieved 100% of covered transitions of the Statechart model.

→ However, even PC1 did not not present a very high average coverage of instructions (75%) and especially branch coverage (49%) of the source code. This is explained by the Statechart model. In general, such models have some normal behavior addressed and a few exception handling situations. Hence, to get higher coverage it is required more detailed situations in the Statechart models.

→ When cost and effectiveness are assessed together, PC1 was better in all situations. When cost and effectiveness contribute equally, the Euclidean distance from the optimum point, *op*, to ABS is 24.62% greater than the distance from *op* to PC1. And from *op* to REC, the distance is 25.12% greater than from *op* to PC1. There is no cost-effectiveness difference when comparing ABS with REC with a slight advantage for ABS.

→ If the effectiveness contribution is doubled, again PC1 was the best option where the Euclidean distances from *op* to ABS and to REC are 63.5% and 48.55% greater than the distance from *op* to PC1, respectively. However, here REC performed better than ABS where the distance from *op* to the latter is 10.07% greater than from *op* to the former.

→ The overall conclusion of this empirical evaluation (quasiexperiment) is that the Precedence Chain (P prec. S, T) pattern with Global scope (PC1) generates the best test suite regarding cost-effectiveness. The Response Chain pattern with Global scope (REC) is better than the Absence pattern with Global scope (ABS) only if effectiveness has a priority over cost. If they contribute equally, there is no difference between REC and ABS.

Keywords:
Software Testing
Model Checking
Quasiexperiment

Who is this briefing for?

Software engineering practitioners who want to make decisions about cost-effectiveness of test suites generated via Model Checking based on scientific evidence.

Where the findings come from?

All findings of this briefing were extracted from the Model Checking test case generation method called HiMoST presented in *Santiago Júnior and Silva*.

What is included in this briefing?

The main findings of the cost-effectiveness empirical evaluation presented in *Santiago Júnior and Silva*.

To access other evidence briefings on software engineering:

http://www.lia.ufc.br/~cbsoft2017/

Original Reference

Santiago Júnior, Valdivino Alexandre de; Silva, F. E. C. . From Statecharts into Model Checking: A Hierarchy-based Translation and Specification Patterns Properties to Generate Test Cases. In: 2nd Brazilian Symposium on Systematic and Automated Software Testing (SAST 2017), 2017, Fortaleza, CE, Brazil. Proceedings of the 2nd Brazilian Symposium on Systematic and Automated Software Testing, 2017. DOI: https://doi.org/10.1145/3128473.3128475.