International Journal on Software Tools for Technology Transfer
https://doi.org/10.1007/s10009-021-00611-5

INTRODUCTION

Special Issue: TestComp 2019

TOOLympics I: Competition on software testing

A special issue for Test-Comp 2019

Dirk Beyer[1] · Marieke Huisman[2]

Accepted: 16 March 2021

Abstract
Research competitions and challenges are a driving force in transferring theoretical results into working software tools that demonstrate the state of the art in the respective field of research. Regular comparative evaluations provide guidance to practitioners that have to select new technology and tools for their development process. In order to support competitions and challenges with an appropriate publication venue, a new theme of issues in the International Journal on Software Tools for Technology Transfer was created. This issue is the inaugural issue of the newly introduced theme on "Competitions and Challenges" (CoCha). Test-Comp, the International Competition on Software Testing, is an example of a tool competition, where the research teams submit tools for test-generation, and the competition evaluates the tools and assigns scores according to achieved coverage. Test-Comp 2019 was part of the TOOLympics event, which took place as part of the 25-year celebration of the conference TACAS. Thus, it is most natural to start the new STTT-CoCha theme with a special issue that describes the results and participating systems of Test-Comp 2019. There will be a second issue on TOOLympics with contributions from other competitions.

Keywords Competition · Challenge · TOOLympics · Software testing · Test-case generation · Coverage analysis · Software tools · Technology transfer · Comparative evaluation

1 New in STTT: Competitions and challenges

We are proud to announce a new theme in STTT: "Competitions and Challenges". This is the inaugural issue of the newly introduced theme of the journal Software Tools for Technology Transfer.

This theme is dedicated to make available overview articles about new competitions, progress reports of established research competitions, and articles that provide insights about research competitions as a scientific method. For the various research communities working on tool implementations, it is important to bring together the community and to compare the state of the art, in order to identify progress of and new challenges in the research area. Also, one of the main challenges in tool development is that it requires considerable engineering effort. In order to publish and widely disseminate the knowledge about tools that represent the state of the art according to the latest research results, we need to obtain results using scientifically valid methods, and rigorous comparative evaluations are an example for such a method.

Evaluation of scientific contributions can be done in many different ways—research competitions and challenges are suitable to evaluate tools and have been a success story so far. Community challenges, or grand challenges, are problems that cannot be solved by a single research team but by the whole community as a long-term project, potentially spanning decades. The goal of such challenges is to focus the community effort on certain topics. Competition events can serve as milestones to capture a certain status. For example, in the early 1990's, when the research area of formal methods became more mature, case studies were pro-

Open Access was funded by Projekt DEAL.

✉ Dirk Beyer
dirk.beyer@sosy-lab.org

[1] LMU Munich, Oettingenstr. 67, 80538 Munich, Germany

[2] University of Twente, P.O. Box 217, 7500 AE, Enschede, Netherlands

Published online: 06 May 2021

posed to get an overview of the strengths and weaknesses of the various modelling approaches. The first such 'competition' was probably the Production Cell case study of the KORSO project [22]. After that, there have been many more, with the VerifyThis Long-Term Challenge [16] as the most recent one. These challenges are in the tradition of evaluating approaches, instead of tool performance.

A different example of challenges are community exemplars, such as for example the Pacemaker Challenge[1], which has been used in over 50 formal-methods research papers and at least one book to illustrate and evaluate formal methods. These are examples that are provided (and perhaps enhanced over time by the community) for the purpose of show-casing and comparing techniques. Such examples are typically designed explicitly as an open-source subject for demonstrating the application of rigorous techniques, while incorporating domain realism (for example, by adapting them from real-world industry artifact), scale, and complexity.

The first formal-methods competition of tools was the SAT competition, which was founded in 1992 [18], shortly followed by the CASC competition in 1996 [27]. Since the year 2000, the number of dedicated formal-methods and verification competitions was steadily increasing. Many of these events now happen regularly, gathering researchers that would like to understand how well their research prototypes work in practice. Scientific results have to be reproducible, and powerful computers are becoming affordable; thus, these competitions are becoming an important means for advancing research progress.

The scope of the new CoCha theme is specialized on, but not limited to, the following publications:

- reports about competitions that describe the progress of technology,
- system descriptions that provide an overview of tools that participated in a competition,
- analysis articles and surveys on the topic of competitions,
- articles that focus on reproducibility and benchmarking technology,
- articles that describe benchmark sets that are used in research competitions,
- proposals and definitions of community challenges,
- progress reports on community challenges, and
- proposals for open-source system examples that are explicitly designed to stimulate community cross-assessment of different methods and demonstration of integration of methods across the system life-cycle.

The Theme Editors in Chief for the STTT theme "Competitions and Challenges" are:

- Dirk Beyer (LMU Munich, Germany)
- Marieke Huisman (University of Twente, Netherlands)

2 This special issue

TOOLympics 2019 was an event to draw attention to the achievements of the various competitions, and to understand their commonalities and differences. The event was part of the celebration of the 25^{th} anniversary of the conference TACAS and was held at ETAPS 2019 in Prague, Czechia. TOOLympics 2019 [3] included presentations of 16 competitions in the area of formal methods: CASC [26], CHC-COMP, CoCo [1], CRV [4], MCC [19], QComp [13], REC [11], RERS [14], Rodeo (planned), SAT [5], SL-COMP [25], SMT-COMP [2], SV-COMP [6], termCOMP [23], Test-Comp [7], and VerifyThis [15].

This issue is the first of two special issues on the TOOLympics 2019 event. The issue is dedicated to Test-Comp, the International Competition on Software Testing, which was held for the first time in 2019. The goals and design of the competition are described in the competition description [7]. The participating teams submitted test-generation tools, and the competition execution consists of (a) running the test-generation and (b) evaluating the produced test-suites regarding coverage. This journal issue contains articles that present the results of the competition in a report by the organizer and 7 selected competition contributions, which are briefly described in the following.

First international competition on software testing [8]

The competition report provides an overview of the competition, the definitions, technical setup, composition of the competition jury, the scoring schema and ranking calculation, and the results.

CoVeriTest: Interleaving value and predicate analysis for test-case generation [17]

CoVeriTest is a hybrid approach to test generation that combines several verification techniques. The tool interleaves a predicate analysis and a value analysis, and allows cooperation between the analyses. For the Test-Comp participation, a configuration was used in which both analyses reuse the internal data structures (abstract reachability graphs) from their previous iteration. CoVeriTest is based on the verification framework CPAchecker.

[1] See http://sqrl.mcmaster.ca/pacemaker.htm.

CPA/Tiger-MGP: Test-goal set partitioning for efficient multi-goal test-suite generation [24]

CPA/Tiger- MGP implements a test-generation technique that is based on configurable multi-goal set partitioning (MGP). The tool supports configurable partitioning strategies and processes several test goals at once in a reachability analysis. CPA/Tiger- MGP is based on a predicate-abstraction-based program analysis of the verification framework CPAchecker.

Esbmc 6.1: Automated test-case generation using bounded model checking [12]

Esbmc is a bounded model checker that uses an SMT-solver as backend. The tool participated in the theme in which the test specification was that a test should be produced that covered a certain function call. For Test-Comp 2019, Esbmc incrementally increased the bound until the specific function call is reached in the program. Once Esbmc has found an error path to the function call, it produces a test suite that contains at least one test to expose the reachability of the function call.

FairFuzz-TC: A fuzzer targeting rare branches [21]

FairFuzz is an Afl-based fuzzing tool that uses coverage-guided mutation. By targeting the mutation strategy towards rare branches, it tries to increase code coverage quickly. The tool participated in Test-Comp with a few modifications, and the competition contribution is called FairFuzz- TC.

Klee Symbolic execution engine in 2019 [9]

Klee is a tool for dynamic symbolic execution. The tool automatically explores the paths of a program, using a constraint solver to decide path feasibility. Klee integrates the solvers Stp, Boolector, Cvc4, Yices 2, and z3. In the configuration for Test-Comp, the tool uses the solver Stp for best performance and was extended such that it can better handle large numbers of symbolic variables.

Plain random test generation with PRTest [20]

PRTest is meant as a baseline tool for test-generation, which means that it uses only a 'plain' approach of random test generation in a black-box manner. PRTest executes the program for which the tests shall be generated and creates a new test value randomly whenever a value is required. The test vector is recorded and in the end, the achieved coverage is measured; the new test vector is added to the test suite only if it increases the coverage. This is executed repeatedly until the coverage criterion is satisfied or the time limit is reached. PRTest is publicly available and open source.

Symbiotic 6: Generating test-cases by slicing and symbolic execution [10]

Symbiotic is a tool for bug-finding that works in two phases: first, it preprocesses the input program by applying static analyses, instrumentation, and program slicing, and second, it executes a symbolic-execution engine to find interesting program paths. Klee is used as backend for symbolic execution.

Acknowledgements We are grateful to all the authors for their contributions and to the jury of Test-Comp 2019 for their help in evaluating the test-generation systems and selecting the papers for this special issue.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Aoto, T., Hirokawa, N., Nagele, J., Nishida, N., Zankl, H.: Confluence Competition 2015. In: Proc. CADE-25, LNCS 9195, pp. 101–104. Springer (2015). https://doi.org/10.1007/978-3-319-21401-6_5
2. Barrett, C., de Moura, L., Stump, A.: Design and results of the 1st Satisfiability Modulo Theories Competition (SMT-COMP 2005). J. Autom. Reason. **35**(4), 373–390 (2005). https://doi.org/10.1007/s10817-006-9026-1
3. Bartocci, E., Beyer, D., Black, P.E., Fedyukovich, G., Garavel, H., Hartmanns, A., Huisman, M., Kordon, F., Nagele, J., Sighireanu, M., Steffen, B., Suda, M., Sutcliffe, G., Weber, T., Yamada, A.: TOOLympics 2019: An overview of competitions in formal methods. In: Proc. TACAS (3), LNCS 11429, pp. 3–24. Springer (2019). https://doi.org/10.1007/978-3-030-17502-3_1
4. Bartocci, E., Bonakdarpour, B., Falcone, Y.: First international competition on software for runtime verification. In: Proc. RV, LNCS 8734, pp. 1–9. Springer (2014). https://doi.org/10.1007/978-3-319-11164-3_1
5. Berre, D.L., Simon, L.: The essentials of the SAT 2003 competition. In: Proc. SAT 2003, LNCS 2919, pp. 452–467. Springer (2004). https://doi.org/10.1007/978-3-540-24605-3_34
6. Beyer, D.: Competition on software verification (SV-COMP). In: Proc. TACAS, LNCS 7214, pp. 504–524. Springer (2012). https://doi.org/10.1007/978-3-642-28756-5_38

7. Beyer, D.: Competition on software testing (Test-Comp). In: Proc. TACAS (3), LNCS 11429, pp. 167–175. Springer (2019). https://doi.org/10.1007/978-3-030-17502-3_11
8. Beyer, D.: First international competition on software testing (Test-Comp 2019). Int. J. Softw. Tools Technol, Transf (2020). https://doi.org/10.1007/s10009-021-00613-3
9. Cadar, C., Nowack, M.: Klee symbolic execution engine in 2019. Int. J. Softw. Tools Technol. Transf. (2020). https://doi.org/10.1007/s10009-020-00570-3
10. Chalupa, M., Vitovska, M., Jašek, T., Šimàek, M., Strejček, J.: Symbiotic 6: Generating test-cases by slicing and symbolic execution. Int. J. Softw. Tools Technol. Transf. (2020). https://doi.org/10.1007/s10009-020-00573-0
11. Denker, G., Talcott, C.L., Rosu, G., van den Brand, M., Eker, S., Serbanuta, T.F.: Rewriting logic systems. Electron. Notes Theor. Comput. Sci. **176**(4), 233–247 (2007)
12. Gadelha, M.R., Menezes, R., Cordeiro, L.: Esbmc, : Automated test-case generation using bounded model checking. J. Softw. Tools Technol. Transf., Int (2020). https://doi.org/10.1007/s10009-020-00571-2
13. Hahn, E.M., Hartmanns, A., Hensel, C., Klauck, M., Klein, J., Křetínský, J., Parker, D., Quatmann, T., Ruijters, E., Steinmetz, M.: The 2019 comparison of tools for the analysis of quantitative formal models. In: Proc. TACAS (3), LNCS 11429, pp. 69–92. Springer (2019). https://doi.org/10.1007/978-3-030-17502-3_5
14. Howar, F., Isberner, M., Merten, M., Steffen, B., Beyer, D.: The RERS grey-box challenge 2012: Analysis of event-condition-action systems. In: Proc. ISoLA, LNCS 7609, pp. 608–614. Springer (2012). https://doi.org/10.1007/978-3-642-34026-0_45
15. Huisman, M., Klebanov, V., Monahan, R.: VerifyThis verification competition 2012: Organizer's report. Tech. Rep. 2013-01, Department of Informatics, Karlsruhe Institute of Technology (2013). Available at http://digbib.ubka.uni-karlsruhe.de/volltexte/1000034373
16. Huisman, M., Monti, R.E., Ulbrich, M., Weigl, A.: The VerifyThis collaborative long term challenge. In: Deductive Software Verification: Future Perspectives — Reflections on the Occasion of 20 Years of KeY, LNCS 12345, pp. 246–260. Springer (2020). https://doi.org/10.1007/978-3-030-64354-6_10
17. Jakobs, M.C.: CoVeriTest: Interleaving value and predicate analysis for test-case generation. Int. J. Softw. Tools Technol, Transf (2020). https://doi.org/10.1007/s10009-020-00572-1
18. Järvisalo, M., Berre, D.L., Roussel, O., Simon, L.: The international SAT solver competitions. AI Magazine **33**(1), (2012)
19. Kordon, F., Linard, A., Buchs, D., Colange, M., Evangelista, S., Lampka, K., Lohmann, N., Paviot-Adet, E., Thierry-Mieg, Y., Wimmel, H.: Report on the model checking contest at Petri nets 2011. Trans. Petri Nets Other Model. Concurr. **VI**, 169–196 (2012). https://doi.org/10.1007/978-3-642-35179-2_8
20. Lemberger, T.: Plain random test generation with PRTest. Int. J. Softw. Tools Technol, Transf (2020). https://doi.org/10.1007/s10009-020-00568-x
21. Lemieux, C., Sen, K.: FairFuzz-TC: A fuzzer targeting rare branches. Int. J. Softw. Tools Technol, Transf (2020). https://doi.org/10.1007/s10009-020-00569-w
22. Lewerentz, C., Lindner, T. (eds.): Formal Development of Reactive Systems: Case Study Production Cell. LNCS 891. Springer (1995). 3-540-58867-1. https://doi.org/10.1007/3-540-58867-1
23. Marché, C., Zantema, H.: The termination competition. In: Proc. RTA, LNCS 4533, pp. 303–313. Springer (2007). https://doi.org/10.1007/978-3-540-73449-9_23
24. Ruland, S., Lochau, M., Fehse, O., Schürr, A.: CPA/Tiger-MGP: Test-goal set partitioning for efficient multi-goal test-suite generation. Int. J. Softw. Tools Technol, Transf (2020). https://doi.org/10.1007/s10009-020-00574-z
25. Sighireanu, M., Cok, D.: Report on SL-COMP '14. J. Satisf. Boolean Model. Comput. **9**(1), 173–186 (2014)
26. Sutcliffe, G., Suttner, C.: The CADE-13 ATP system competition. J. Autom. Reason. **18**(2), 137–138 (1997). https://doi.org/10.1023/A:1005839515219
27. Suttner, C.B., Sutcliffe, G.: The design of the CADE-13 ATP system competition. J. Automat. Reason. **18**(2), 139–162 (1997). https://doi.org/10.1023/A:1005802523220