International Journal on Software Tools for Technology Transfer (2020) 22:685–687
https://doi.org/10.1007/s10009-020-00581-0

GENERAL

TACAS 2018

Tools for the construction and analysis of systems

A special issue for TACAS 2018

Dirk Beyer[1] · **Marieke Huisman**[2]

Published online: 13 July 2020

Abstract
In order to develop reliable software and systems, we depend on practical techniques for the construction and analysis of such software and systems. This special issue of Software Tools for Technology Transfer presents various tool-supported techniques that can help with the construction and analysis of such reliable software and systems. The papers in this special issue are extended versions of selected conference papers from the proceedings of the 24th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2018).

Keywords Safety analysis · Quantitative analysis · Deductive verification · Statistical model checking · Runtime analysis · Refinement · Reactive system · Cyber-physical system · Weak-memory model · Hyperproperty

1 TACAS

This special issue of the journal Software Tools for Technology Transfer (STTT) contains revised and extended versions of five papers selected out of 45 papers accepted as regular research papers at the 24th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2018) [1,2].

TACAS is a forum for researchers, developers, and users interested in rigorously based tools and algorithms for the construction and analysis of systems. The conference aims to bridge the gaps between different communities with this common interest and to support them in their quest to improve the utility, reliability, flexibility, and efficiency of tools and algorithms for building systems. Topics that are of interest to TACAS are:

- specification and verification techniques;
- software and hardware verification;
- analytical techniques for real-time, hybrid, or stochastic systems;
- analytical techniques for safety, security, or dependability;
- SAT and SMT solving;
- theorem-proving;
- model-checking;
- static and dynamic program analysis;
- testing;
- abstraction techniques for modeling and verification;
- compositional and refinement-based methodologies;
- system construction and transformation techniques;
- machine-learning techniques for synthesis and verification;
- tool environments and tool architectures;
- applications and case studies.

2 This special issue

The peer-reviewed papers collected in this special issue have been invited by the guest editors among the top papers presented at TACAS 2018 based on their relevance to STTT. They all report on tools and tool sets that advance the construction and analysis of software systems. In addition to this special issue, a companion special issue for TACAS 2018 will

✉ 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

appear in the Journal of Automated Reasoning, containing selected papers with more theoretical results.

The papers in this special issue all develop tool-supported analysis techniques. The first three papers focus on safety properties of either Simulink models, programs executed on a weak-memory model, or of a set of traces of, for example, a Verilog program. The last two papers focus on quantitative properties.

Below, we give a short summary of each paper.

2.1 Safety analysis

The refinement calculus of reactive systems tool set [4]

This paper presents a tool set for the modeling and analysis of reactive systems. The tool set is based on the refinement calculus of reactive system (RCRS), which is a contract-based refinement framework that combines the classical refinement calculus, interface automata, and the theory of relational interfaces. The goal of RCRS is to be able to model industrial-strength systems. The tool set provides a full formalization of this calculus, modeled with the Isabelle theorem prover. In addition, a suitable analysis procedure for the RCRS models is also formalized in Isabelle. For practical applicability, the tool set provides support to translate Simulink diagrams into RCRS models.

The paper outlines the tool set on a series of smaller examples. In addition, to illustrate the practical usability of the approach, the RCRS tool set has also been successfully applied to an industrial case study to model and analyze a fuel control system.

Automating deductive verification for weak-memory programs (extended version) [7]

This paper presents an encoding of weak-memory program logics using the existing deductive verification tool Viper. Over the last years, several program logics for weak-memory models, such as the C11 memory model, have been proposed. So far, these program logics have been without tool support, as automating proofs in these logics is non-trivial, due to features such as higher-order assertions, modalities, and rich-permission resources.

This paper for the first time proposes an encoding of such a program logic into an existing deductive program verifier. The key insight of this encoding is that it allows to reduce all features that make reasoning about weak-memory models complicated into a much simpler sequential logic. A major advantage of the approach is that it enables the reuse of all the verification infrastructure in the Viper framework, thus providing a combination of flexibility and automation.

The paper presents encodings into Viper for three recent C11 program logics: Relaxed Separation Logic and two forms of Fenced Separation Logic. The paper emphasizes the different encoding techniques that have been used, which could also be used for other logics. The encodings have been evaluated on various examples from existing papers as well as the Facebook open-source Folly library.

Efficient monitoring of hyperproperties using prefix trees [5]

This paper focuses on monitoring the runtime behavior of systems. In particular, it presents a new version of a tool to monitor *hyperproperties*, i.e., properties that relate multiple computation traces with each other. Typical examples of hyperproperties are non-interference and observational determinism. Classical monitoring tools that consider computations in isolation are not able to monitor such properties, and RVHyper is one of the few tools that supports the monitoring of such multiple computation properties.

The paper presents the latest version of RVHyper in detail. First, it describes the input language for the property specifications, which are given in the temporal logic HyperLTL, which is an extension of linear-time temporal logic (LTL) with trace quantifiers and trace variables. Then, it describes how RVHyper processes execution traces sequentially until a violation of the specification is detected. In this case, a counterexample, in the form of a set of traces, is returned.

Further, the paper also describes various optimizations that have been implemented for RVHyper recently. In order to reduce the amount of work that the tool has to do (and thus, the overhead of the monitoring procedure), RVHyper implements a specification analysis to avoid unnecessary work. A second optimization is a novel trace storage technique, which makes it easier to exploit partial equality between traces.

Finally, RVHyper is evaluated on several standard benchmarks from the security domain: an encoder that guarantees a Hamming distance of 2, violations of non-interference on randomly generated traces, and a symmetry property on an implementation of the Bakery protocol. In addition, to illustrate that RVHyper also can be used outside the security domain, a final benchmark is discussed to detect spurious dependencies in hardware designs.

2.2 Quantitative analysis

AMT 2.0: qualitative and quantitative trace analysis with Extended Signal Temporal Logic [6]

This paper presents the tool AMT 2.0, which is a tool for qualitative and quantitative runtime analysis of cyber-physical systems, which are modeled based on a combination of continuous and Boolean signals, using both numerical values and discrete events.

The tool uses Extended Signal Temporal Logic to specify the desired properties of the system. This Extended Signal Temporal Logic combines Timed Regular Expressions with Signal Temporal Logic. The tool can be used for qualitative monitoring, i.e., monitoring property satisfaction, as well as for measuring of quantitative features of signals. The tool provides ample support for trace diagnostics for explaining and justifying property violations.

The paper illustrates the usage of the tool on two non-trivial case studies: failure detection in an aircraft elevator control system and a timing analysis for the initialization phase of a connection protocol between a remote sensor and a microcontroller, used in the automotive industry.

An efficient statistical model checker for nondeterminism and rare events [3]

This paper presents Modes, a statistical model checker that can efficiently handle systems with nondeterminism and rare events. Statistical model checking is typically used for analysis of stochastic models, as it avoids the state-space explosion problem in verification and naturally supports complex non-Markovian formalisms. However, its runtime quickly grows in the presence of rare events, and it cannot soundly analyze nondeterministic models.

To address this issue, the paper presents the new statistical model checker Modes. Modes combines fully automated importance splitting to estimate the probabilities of rare events and a smart lightweight scheduler sampling technique to approximate optimal schedulers in nondeterministic models. Modes is part of the MODEST tool set, and as a result it supports a variety of input formalisms natively and via the JANI exchange format.

The paper describes how Modes has been used effectively on three different case studies: electric vehicle charging, low-latency wireless network, and a redundant database system. In addition, its performance has been compared favorably with several other statistical model checkers.

Acknowledgements Open Access funding provided by Projekt DEAL. We are grateful to all the authors for their contributions and to the program committee of TACAS 2018 for their help in selecting the papers for the conference program, including the papers for this issue. We are especially grateful to the TACAS 2018 program committee and external referees who reviewed the extended versions of the papers that appear in this special issue.

References

1. Beyer, D., Huisman, M. (eds.): Tools and Algorithms for the Construction and Analysis of Systems—24th International Conference, TACAS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, 14–20 April 2018, Proceedings, Part I, LNCS 10805. Springer (2018). https://doi.org/10.1007/978-3-319-89960-2
2. Beyer, D., Huisman, M. (eds.): Tools and Algorithms for the Construction and Analysis of Systems—24th International Conference, TACAS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, 14–20 April 2018, Proceedings, Part II, LNCS 10806. Springer (2018). https://doi.org/10.1007/978-3-319-89963-3
3. Budde, C.E., D'Argenio, P.R., Hartmanns, A., Sedwards, S.: An efficient statistical model checker for nondeterminism and rare events. Int. J. Softw. Tools Technol. Transf. (2020). https://doi.org/10.1007/s10009-020-00563-2
4. Dragomir, I., Preoteasa, V., Tripakis, S.: The refinement calculus of reactive systems toolset. Int. J. Softw. Tools Technol. Transf. (2020). https://doi.org/10.1007/s10009-020-00561-4
5. Finkbeiner, B., Hahn, C., Stenger, M., Tentrup, L.: Efficient monitoring of hyperproperties using prefix trees. Int. J. Softw. Tools Technol. Transf. (2020). https://doi.org/10.1007/s10009-020-00552-5
6. Ničković, D., Lebeltel, O., Maler, O., Ferrère, T., Ulus, D.: AMT 2.0: qualitative and quantitative trace analysis with extended signal temporal logic. Int. J. Softw. Tools Technol. Transf. (2020). https://doi.org/10.1007/s10009-020-00582-z
7. Summers, A.J., Müller, P.: Automating deductive verification for weak-memory programs (extended version). Int. J. Softw. Tools Technol. Transf. (2020). https://doi.org/10.1007/s10009-020-00559-y