Preface to the Special Section Issue on Improving Software Quality through Formal Methods

Yliès Falcone, Leonardo Mariani

HAL Id: hal-02548907

https://inria.hal.science/hal-02548907

Submitted on 21 Apr 2020

Preface to the Special Issue on Improving Software Quality through Formal Methods

Yliès Falcone · Leonardo Mariani

Abstract This special issue is dedicated to the presentation of novel results in the scope of formal methods for the verification and testing of software. The papers included in the special issue showcase approaches that successfully combine a rigorous and formal background with the capability to address real software applications in realistic settings.

1 Introducing the Special Issue

The special issue is devoted to the use of formal methods for improving the quality of software systems. The included articles focus on the use of runtime techniques, such as runtime verification and trace analysis; see Falcone et al. [2013]; Bartocci and Falcone [2018]; Bartocci et al. [2019]. Following an open call, this issue results in 3 papers that were selected out of 8 papers after a careful reviewing process. The papers are briefly discussed as follows.

In "Verifying Temporal Specifications of Java Programs", Spegni et al. combine satisfiability modulo theory and static analysis to approximate the behavior of Java threads. The behavior approximation is represented as a network of timed automata and the authors show that their abstraction preserves the truthfulness of two real-time temporal logics.

Y. Falcone
Univ. Grenoble Alpes, Inria, CNRS, Laboratoire d'Informatique de Grenoble,
38000 Grenoble, France
Tel.: +33 4 38 78 61 52
E-mail: ylies.falcone@univ-grenoble-alpes.fr

L. Mariani
University of Milano - Bicocca
Viale Sarca 336, IT-20126 Milan, Italy
Tel.: +39 02 6448 7870
E-mail: mariani@disco.unimib.it

In "Runtime Verification of Real-Time Event Streams Under Non-synchronized Arrival", Scheffel et al. present an approach to monitor concurrent systems with a shared clock abstracted as real-time event streams. The authors present a specification language and an online monitoring algorithm for the evaluation of the specifications.

In "Model Generation of Component-based Systems", Salva and Blot present an approach to learn the behavior of component-based systems from their execution traces using machine learning techniques. The resulting tool has been successfully experimented with network protocol traces.

The special issue was announced during the track on Software Verification and Testing (SVT) at the 33rd ACM/SIGAPP Symposium On Applied Computing. One of the papers accepted in this special issue is an extended version of a paper that originally appeared at the SVT track. Collectively, these papers tackle cutting-edge topics thereby appealing to the experts in the field.

Finally, the guest editors would like to thank the authors and the reviewers who contributed to the special issue.

References

Bartocci E, Falcone Y (eds) (2018) Lectures on Runtime Verification - Introductory and Advanced Topics, Lecture Notes in Computer Science, vol 10457. Springer, DOI 10.1007/978-3-319-75632-5

Bartocci E, Falcone Y, Bonakdarpour B, Colombo C, Decker N, Havelund K, Joshi Y, Klaedtke F, Milewicz R, Reger G, Rosu G, Signoles J, Thoma D, Zalinescu E, Zhang Y (2019) First international competition on runtime verification: rules, benchmarks, tools, and final results of CRV 2014. STTT 21(1):31–70, DOI 10.1007/s10009-017-0454-5

Falcone Y, Havelund K, Reger G (2013) A tutorial on runtime verification. In: Broy M, Peled DA, Kalus G (eds) Engineering Dependable Software Systems, NATO Science for Peace and Security Series, D: Information and Communication Security, vol 34, IOS Press, pp 141–175, DOI 10.3233/978-1-61499-207-3-141