International Journal on Software Tools for Technology Transfer (2021) 23:545–546
https://doi.org/10.1007/s10009-021-00621-3

GENERAL

Special Issue: RV 2019

Introduction to the special issue of the 19th International Conference on Runtime Verification

Bernd Finkbeiner[1] · Leonardo Mariani[2]

Accepted: 6 May 2021 / Published online: 21 May 2021

This special issue includes the extended versions of six selected papers from the refereed proceedings of the 19th International Conference on Runtime Verification (RV 2019), which was held during October 8–11, 2019, in Porto, Portugal, as part of the Third World Congress on Formal Methods (FM 2019). Runtime verification encompasses all aspects of monitoring and analysis of hardware, software, and system executions. Runtime verification techniques are crucial for system correctness, reliability, and robustness; they provide an additional level of rigor and effectiveness compared to conventional testing, and are generally more practical than exhaustive formal verification. Runtime verification can be used prior to deployment, for testing, verification, and debugging purposes, and after deployment for ensuring reliability, safety, and security and for providing fault containment and recovery as well as online system repair.

One of the most fundamental questions in runtime verification is the choice of the specification language. The article "An Extension of First-Order LTL with Rules with Application to Runtime Verification" by Klaus Havelund and Doron Peled presents a monitoring algorithm for the extension of past time first-order linear-time temporal logic with rules [3]. This logic combines two important aspects of expressiveness: first-order LTL adds quantification over data to standard LTL; the rules introduce auxiliary propositions, i.e., propositions that do not appear in the model itself. Such auxiliary propositions allow for the natural expression of ω-regular properties.

Another novel specification language is proposed in the article "Specifying and Detecting Temporal Patterns with Shape Expressions" by Dejan Nickovic, Xin Qin, Thomas Ferrère, Cristinel Mateis, and Jyotirmoy Deshmukh [5]. The authors consider sophisticated temporal patterns in data obtained from cyber-physical systems and the Internet-of-Things (IoT). Shape expressions are a declarative specification language based on regular expressions, where atomic predicates are shapes with parameters such as slope, offset, or frequency. The authors study essential properties of the language and develop a technique for the approximate matching of shape expressions.

The article "What Can We Monitor Over Unreliable Channels?" by Sean Kauffman, Klaus Havelund, and Sebastian Fischmeister studies the effect of imperfect communication on the monitorability of a specification [4]. The authors define how a verdict for a property may be trustworthy over an unreliable channel even when the property is not immune to the channel's mutation. This leads to a classification of properties that may be monitored over certain unreliable channels.

Another foundational topic is explored in the article "Comparing Controlled System Synthesis and Suppression Enforcement" by Luca Aceto, Ian Cassar, Adrian Francalanzam, and Anna Ingólfsdóttir [2]. The article investigates the interplay between static and dynamic methods by relating runtime enforcement and control system synthesis. Both techniques modify the behavior of a system to prevent erroneous executions. The paper shows that in the context of safety properties, control synthesis is the static counterpart of suppression-based runtime enforcement.

The article "Neural Predictive Monitoring and a Comparison of Frequentist and Bayesian Approaches" by Luca Bortolussi, Francesca Cairoli, Nicola Paoletti, Scott A. Smolka, and Scott D. Stoller is concerned with the predictive monitoring problem [1]. The predictive monitoring problem appears in architectures for runtime safety assurance, where the monitor needs to determine whether an unsafe state can be reached from the current system state within a given time bound. Neural State Classification is a method for predictive monitoring of hybrid automata using deep neural networks.

✉ Leonardo Mariani
leonardo.mariani@unimib.it

Bernd Finkbeiner
finkbeiner@cispa.de

[1] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

[2] University of Milano - Bicocca, viale Sarca 336, 2016 Milan, Italy

The article introduces a novel technique that complements NSC predictions with estimates of the predictive uncertainty. Such measures are useful to reject predications that are likely to be incorrect and would therefore negatively impact reliability.

The article "Quantitative Estimation of Side Channel Leaks with Neural Networks" by Saeid Tizpaz-Niari, Pavol Černý, Sriram Sankaranarayanan, and Ashutosh Trivedi presents a data-driven dynamic analysis for detecting and quantifying information leaks via side channels [6]. Confidentiality is a critical property for software that handles sensitive data such as financial or medical information. Purely qualitative methods, such as forbidding any flow of information, are often not practical, because some flow of information is needed to fulfill the functionality of the system. The article presents a two-step approach to this problem. First, a timing model of the program is learned as a neural network; then, the neural network is analyzed in order to quantify how much information is leaked.

All six articles significantly extend the corresponding papers from the refereed proceedings of the conference. They impressively demonstrate the breath and vibrancy of the current runtime verification research.

Funding Open access funding provided by Università degli Studi di Milano - Bicocca within the CRUI-CARE Agreement.

References

1. Bortolussi, L., Cairoli, F., Paoletti, N., Smolka, S.A., Stoller, S.D.: Neural predictive monitoring and a comparison of frequentist and Bayesian approaches. Int. J. Softw. Tools Technol. Transf. (2021). https://doi.org/10.1007/10009-021-00623-1
2. Cassar, I., Francalanza, A., Aceto, L., Ingólfsdóttir, A.: Comparing controlled system synthesis and suppression enforcement. Int. J. Softw. Tools Technol. Transf. (2021). https://doi.org/10.1007/s10009-021-00624-0
3. Havelund, K., Peled, D.: An extension of first-order LTL with rules with application to runtime verification. Int. J. Softw. Tools Technol. Transf. (2021). https://doi.org/10.1007/s10009-021-00626-y
4. Kauffman, S., Havelund, K., Fischmeister, S.: What can we monitor over unreliable channels? Int. J. Soft. Tools Technol. Transf. (2021). https://doi.org/10.1007/s10009-021-00625-z
5. Nickovic, D., Qin, X., Ferrère, T., Mateis, C., Deshmukh, J.: Specifying and detecting temporal patterns with shape expressions. Int. J. Softw. Tools Technol. Transf. (2021). https://doi.org/10.1007/s10009-021-00627-x
6. Tizpaz-Niari, S., Černý, P., Sankaranarayanan, S., Trivedi, A.: Quantitative estimation of side channel leaks with neural networks. Int. J. Softw. Tools Technol. Transf. (2021). https://doi.org/10.1007/s10009-021-00622-2