



# Introduction to the special issue for SPIN 2021

Alfons Laarman<sup>1</sup> · Ana Sokolova<sup>2</sup>

Accepted: 27 January 2023 / Published online: 25 February 2023  
© The Author(s) 2023

## Abstract

The 27th International Symposium on Model Checking Software, SPIN 2021, was held online, July 12, 2021. The current special issue contains extended versions of three selected works published at the symposium. This short introduction presents these selected papers and the selection process.

## 1 Introduction

This special issue contains the extended and improved versions of the best papers presented at the 27th International Symposium on Model Checking Software, SPIN 2021, held online, July 12, 2021. The symposium was co-located with ECOOP and ISSTA 2021.

The original submissions to the conference were addressing topics like formal verification techniques for automated analysis of software; formal analysis for modeling languages, such as UML/state charts; formal specification languages, temporal logic, design-by-contract; model checking, automated theorem proving, including SAT and SMT; verifying compilers; abstraction and symbolic execution techniques; static analysis and abstract interpretation; combination of verification techniques; modular and compositional verification techniques; verification of timed and probabilistic systems; automated testing using advanced analysis techniques; combination of static and dynamic analyses; derivation of specifications, test cases, or other useful material via formal analysis; case studies of interesting systems or with interesting results; engineering and implementation of software verification and analysis tools; benchmark and comparative studies for formal verification and analysis tools; formal methods education and training; insightful surveys or historical accounts on topics of relevance to the symposium;

and relevant tools and algorithms for modern hardware, e.g.: parallel, GPU, TPU, cloud, and quantum.

The symposium attracted 20 submissions that were rigorously reviewed by 34 Program Committee (PC) members. The selection process included further online discussion open to all PC members. As a result, eight papers were selected for presentation at the symposium and publication in Springer's proceedings. The program consisted of eight regular papers and three invited talks. Two of the invited speakers submitted an invited paper on the topic of their talk.

From the eight regular papers, we invited the best ones based on the reviews and the presentations in the event. The authors of three papers accepted to improve and to extend their works. The improved versions in this issue represent significant extensions of the work published at SPIN, featuring entirely new results (an extension of LTL and industrial case studies in [2]), features and correctness proofs (the support for reachability properties in [1]) and improvements in approach and empirical evaluation (support for abstraction and a substantial comparison against other tools in [3]). The extended submissions went through a new review process with at least three reviewers and two phases for each paper. The three accepted papers included in this issue addressed all the concerns by the reviewers and by the guest editors.

In the first paper, “Go2Pins: a framework for the LTL verification of Go programs (Extended Version)” [2], the authors present Go2Pins, a software model checker for Go with two backends: Spot and LTSmin. It supports a large subset of the Go runtime, including Go concurrency, and uses techniques like abstraction and reduction to handle industrial-scale projects. To check for temporal properties, the tool uses LTL, which is extended to support recursive function calls in Go. The authors demonstrate the effective-

✉ Alfons Laarman  
a.w.laarman@liacs.leidenuniv.nl

Ana Sokolova  
anas@cs.uni-salzburg.at

<sup>1</sup> Leiden University, Leiden, The Netherlands

<sup>2</sup> University of Salzburg, Salzburg, Austria

ness of Go2Pins on two industrial problems and a set of LTL formulae.

In the second paper, “Leveraging polyhedral reductions for solving Petri net reachability problems” [1], the authors present a new way to exploit structural reductions for verifying reachability properties in Petri nets. The reduction method is formalized as a Token Flow Graph (TFG), which captures the so-called ‘polyhedral abstractions.’ Polyhedral abstractions are a combination of structural reductions together with linear-arithmetic constraints between the marking of places. The authors show how the abstract system preserves the reachability of places, as well as the concurrent places in the original system. A tool called Kong implements the newly proposed algorithms, demonstrating the effectiveness of the TFG as a data structure. The authors evaluate Kong on a large collection of inputs from the 2020 Model Checking Contest. The results show that the approach works well and is efficient even when only a moderate amount of reduction is achieved.

In the third paper, “An Evaluation of Approaches to Model Checking Real-Time Task Schedulability Analysis” [3], the authors present an abstraction technique for solving the problem of real-time scheduling by model checking of timed automata. The input to this problem is a set of tasks, each of them having three components: the worst-case execution time, the deadline, and the execution period. The idea of the approach is to keep a small number of tasks (e.g., with the earliest deadlines) and abstract away the other tasks. The concrete tasks are called a spotlight, whereas the abstract tasks are called shade. If the model checking algorithm reports no violation of time constraints, more tasks are iteratively brought from the shade into the spotlight, with which the model checker can proceed. These steps are repeated until a decisive schedulability result is obtained. The authors

formally present the spotlight abstraction, as well as the exploration algorithm and an implementation. A comparison against Timestool in the analysis of the schedulability of real-time tasks shows that the spotlight approach is able to handle more tasks.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Amat, N., Zilio, S.D., Botlan, D.L.: Leveraging polyhedral reductions for solving petri net reachability problems. *Int. J. Softw. Tools Technol.* (2023)
2. Kirszenberg, A., Martin, A., Moreau, H., Renault, E.: Go2pins: a framework for the ltl verification of go programs (extended version). *Int. J. Softw. Tools Technol.* (2023)
3. Nxumalo, M., Timm, N., Gruner, S.: An evaluation of approaches to model checking real-time task schedulability analysis. *Int. J. Softw. Tools Technol.* (2023)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.