Universität Regensburg

FACULTY FOR INFORMATICS AND DATA SCIENCE
Theoretical Computer Science Group

University of Regensburg · 93040 Regensburg · Germany

Prof. Dr. Philipp Rümmer
Phone +49 941 943-68612
Room 602
Administration office:
Phone +49 941 943-68610

Bajuwarenstraße 4
93053 Regensburg
Germany

philipp.ruemmer@ur.de
https://philipp.ruemmer.org

Faculty of Mathematics and Physics
Charles University
Prague, Czech Republic

Regensburg, 3rd March 2023

Review of Martin Blicha's doctoral dissertation

The reviewed PhD thesis presents new methods for the automatic formal verification of systems and software programs. The work focuses on the analysis of safety properties ("nothing bad should happen"), and investigates the problems of automatically analysing systems on several levels: on the level of Craig interpolation, which is a basic procedure used to create invariant candidates; on the level of model checking algorithms, i.e., of the overall algorithms that determine safety of a system; on the tool implementation level, presenting a new solver for the intermediate language of constrained Horn clauses (CHC); and on the level of combining several analysis tools (or instances of analysis tools) in a parallel verifier.

The work is presented in a monograph consisting of 7 chapters, four of which providing the main research results. The work is based on five peer-reviewed publications at high-profile venues: four conference papers (2× TACAS, 1× VMCAI, 1× FMCAD) and one journal article (STTT).

[As all of the papers are authored by multiple people, it would have been useful to summarise the contributions made by the individual authors in the dissertation.]

Chapter 1: Introduction; Chapter 2: Preliminaries; Chapter 7: Conclusions

The general motivation of the work presented; an introduction to formal verification and model checking; a high-level survey of the existing model checking algorithms; the research questions or challenges considered in the work; and summary of the contributions. Chapter 2 gives a more detailed introduction to the field and the key notions used later in the thesis. Chapter 7 summarises the work, and outlines future directions of research.

Comments: The chapters are very clearly written and to the point, and provide a good background for the later chapters by putting the thesis into a bigger context. The author has worked on the model checking problem from various angles, and from the provided overview it is clear that he has an excellent, holistic understanding of the concepts used in formal verification, and of the relevance of the problem for applications.

Chapter 3: Decomposing Farkas Interpolants

Craig interpolants in linear arithmetic are typically derived by appealing to Farkas' lemma, a basic result from linear programming. This chapter presents a way to obtain logically stronger interpolants by decomposing the witness that is used by the standard interpolant construction. Such decomposition is not always possible, but when it applies it gives rise to interpolants that are conjunctions of inequalities, and that can sometimes prevent non-termination of model checkers. In experiments, it is found that the computed stronger interpolants are complementary to the standard interpolants: while they do not tend to lead to better performance in model checking (compared to the standard interpolants) when used on their own, building a portfolio of model checkers based on the standard and the decomposed interpolants leads to a significant increase in performance.

Comments: The chapter presents a basically simple (but novel) idea to compute a larger set of interpolants from a given proof of unsatisfiability in linear arithmetic. The work follows a line of research on how to control the strength of interpolants, derived from a single proof, that has been pursued for several years by different groups. The work is very carefully executed, and the technical details of how to compute decompositions are non-trivial. It is in particular encouraging to see that controlling interpolant strength can lead to significantly better model checking performance, a result that is not always easy to obtain in this context. Since the presented construction is relatively simple, and can easily be re-implemented, it can be expected that it will become one of the standard procedures (together with the standard method for computing interpolants) in model checkers in the future.

[A point that is surprisingly not discussed in the chapter is the question whether (or how) decomposed interpolants can be used to compute sequence interpolants (see Section 2.2). Model checking algorithms often require sequence interpolants, or even tree interpolants to work.]

Chapter 4: Transition Power Abstraction

This chapter presents new algorithms for bounded model checking, i.e., for the problem of showing the safety of a system up to a given depth in terms of the number of execution steps. The main idea is to apply a principle known as the "squaring trick" to consider system executions up to an exponential bound 2^n; such long executions are recursively represented as the concatenation of two executions of length up to 2^{n-1}, and over-approximated using Craig interpolants. The resulting algorithm is able to find counterexamples to safety properties that are challenging for existing algorithms, because simply too many execution steps are needed to find property violations in a system. The method can

be extended to also show correctness of safety properties by verifying that the computed transition relations are inductive, and further strengthened by combination with the k-induction principle.

Comments: In my opinion, this chapter presents the most significant contribution of the thesis: developing a fundamentally new algorithm in a field like model checking, in which many groups and researchers have been active over the last decades, is a big result. While the work is based on a known principle in Computer Science, it was not obvious how to turn this principle into an abstraction-based model checking algorithm. By combining repeated squaring of relations with Craig interpolation, the method developed in the chapter achieves this transfer in a remarkably elegant way. The method also shows very promising results in experiments (as presented in Chapter 5), outperforming existing algorithms on a relevant class of problems, and it has already caught the attention of people in the field. The method is also compatible with many optimisations, and will likely be re-implemented, generalised, and included also in other model checkers, having long-term impact both in academia and industry.

Chapter 5: The Golem Horn Solver

A new solver for the intermediate language of constrained Horn clauses is presented. The tool Golem has been developed from scratch by the candidate, and is used as a test-bed for the techniques presented in the dissertation. Golem includes five different model checking algorithms, one of them being the algorithm from Chapter 4, and it has participated at the last two competitions for CHC solvers, CHC-COMP. The chapter discusses the architecture and different components of Golem in detail, and also provides a detailed experimental comparison of the algorithms in Golem with each other, and of Golem with other CHC solvers.

Comments: The chapter presents an impressive amount of work: it is a huge effort to develop a model checker from scratch, and even more impressive to optimise the tool to a point where it can win competitions. The provided evaluation is carefully carried out, and interesting as it enables a direct comparison of different model checking algorithms within a single tool, i.e., with comparable data-structures and constraints. The evaluation contains some lessons that are not entirely expected, for instance the surprisingly good performance of the "Lazy abstraction with interpolants" algorithm.

Chapter 6: Cooperative Parallelization of Approach for Property-Directed k-Induction

In the final technical chapter, an existing model checking algorithm, PD-KIND, is re-engineered and generalised, resulting in a new framework IcE/FiRE. The framework is then used to explore different ways to parallelise model checkers, and to experiment with different ways of sharing information between multiple model checkers running in parallel. The work connects to the Craig interpolation method developed in Chapter 3, which is used as one of the options when building portfolios.

Comments: While the results presented in this chapter heavily build on the existing PD-KIND method, the presentation chosen here is rather different, technically intricate, and arguably more elegant (and more general) than the original papers introducing PD-KIND. The resulting method performed very well at the CHC-COMP competition in 2020. It is a pity, though, that the work was not better connected to the approaches and tools in Chapter 5 and 6; for instance, it would be extremely interesting how the IcE/FiRE framework compares to the five algorithms implemented in Golem (Chapter 5).

Overall Assessment and Recommendation

The thesis presents an impressive amount of research. It is well-rounded and demonstrates competency in different areas related to model checking. The thesis makes contributions both on a conceptual level and on the level of tool engineering. The idea underlying in particular the "Transition Power Abstraction" method (Chapter 4) is intriguing, and it will be interesting to see future adoption and extensions of the method. The work is presented in a coherent and problem-oriented way, in each chapter clearly motivating the problems that are addressed. Contributions are well explained and carefully worked out, and proofs are provided for all results in the thesis. The thesis is also written very well, there are hardly any language mistakes throughout the document. Overall, the thesis clearly demonstrates the ability of the candidate to perform creative scientific work at a very advanced level.

Grade recommendation: Pass

Prof. Dr. Philipp Rümmer