





Dominic Steinhöfel | Curriculum Vitae

 <https://www.dominic-steinhoefel.de>
 <https://scholar.google.de/citations?user=TshKYLEAAAAJ>
 <https://dblp.uni-trier.de/pers/hd/s/Steinh=ouml=fel:Dominic>
 <https://orcid.org/0000-0003-4439-7129>

Education

Technische Universität Darmstadt, Dr. rer. nat. (Computer Science) 2020

Thesis Abstract Execution: Automatically Proving Infinitely Many Programs [1]

Supervisors Prof. Dr. Reiner Hähnle, Prof. Gilles Barthe, PhD

Grade *Summa cum laude* (with distinction)

The dissertation presents a technique named *Abstract Execution* to prove functional and relational properties of *abstract* programs, especially of program transformation rules. Abstract Execution trades off expressiveness and automation, yielding a general framework allowing for fully automatic proofs in many cases. An application of this technique to well-known equivalence preserving program transformation rules, so-called *refactorings*, provides new preconditions for a safe application of these rules, exceeding the state documented in the literature. Conditional correctness of the rules is proven fully automatically (even for loop transformations). Furthermore, the dissertation contains fundamental contributions to symbolic execution, and proposes a trace-based framework unifying different problems in the area of program verification.

Technische Universität Darmstadt, M. Sc. (Computer Science) 2015

Thesis From Trees to Directed Acyclic Graphs: A General Lattice Model for Symbolic Execution [2]

Supervisors Prof. Dr. Reiner Hähnle, Priv.-Doz. Dr. Richard Bubel, Dr. Nathan Wasser

Grades Thesis: 1.0 (very good), Average: 1.35 (very good)

Technische Universität Darmstadt, B. Sc. (Computer Science) 2013

Thesis Enforcing Datalog Policies with Service Automata on Distributed Version Control Systems [3]

Supervisors Prof. Dr. Heiko Mantel, Dr. Richard Grewe (né Gay)

Grades Thesis: 1.7 (good), Average: 1.4 (very good)

Publications

Below are lists of my theses and peer-reviewed publications (including those still under review). References [5, 6, 8] and [10, 11] are largely included in my PhD thesis.

Two papers reporting on applications of Abstract Execution to cost analysis of program transformations [13] and correctness of transformations for parallelization [12] as well as a tool paper on the REFINITY workbench for modeling and proving program transformations with Abstract Execution [11] are currently under review, as well as a journal version [10] of the Trace Modality framework [6].

Please note that I authored [2, 3] and [9] under my former family name Scheurer.

Theses

1. Dominic Steinhöfel. *Abstract Execution: Automatically Proving Infinitely Many Programs*. PhD thesis, Technical University of Darmstadt, Department of Computer Science, Darmstadt, Germany, 2020. doi: 10.25534/tuprints-00008540. URL <http://tuprints.ulb.tu-darmstadt.de/8540/>.

2. Dominic Scheurer. From Trees to DAGs: A General Lattice Model for Symbolic Execution. Master's thesis, Technical University of Darmstadt, Department of Computer Science, Darmstadt, Germany, 2015. URL https://download.hrz.tu-darmstadt.de/media/FB20/Dekanat/Publikationen/SE/Scheurer_-_2015_-_From_Trees_to_DAGs.pdf.
3. Dominic Scheurer. Enforcing Datalog Policies with Service Automata on Distributed Version Control Systems. Bachelor's thesis, Technical University of Darmstadt, Department of Computer Science, Darmstadt, Germany, 2013.

Scientific Journals

Coauthor

4. Stijn de Gouw, Frank S. de Boer, Richard Bubel, Reiner Hähnle, Jurriaan Rot, and Dominic Steinhöfel. Verifying OpenJDK's Sort Method for Generic Collections. *J. Autom. Reasoning*, 62(1):93–126, 2019. doi: 10.1007/s10817-017-9426-4.

International Conferences

Main Author

5. Dominic Steinhöfel and Reiner Hähnle. Abstract Execution. In *Proc. Third World Congress on Formal Methods - The Next 30 Years, (FM)*, LNCS, pages 319–336, 2019. doi: 10.1007/978-3-030-30942-8_20.
6. Dominic Steinhöfel and Reiner Hähnle. The Trace Modality. In *Proc. Second Intern. Workshop on Dynamic Logic. New Trends and Applications (DaLi)*, LNCS, pages 124–140, 2019. doi: 10.1007/978-3-030-38808-9_8.
7. Dominic Steinhöfel and Reiner Hähnle. Modular, Correct Compilation with Automatic Soundness Proofs. In Tiziana Margaria and Bernhard Steffen, editors, *Proc. 8th Intern. Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA)*, LNCS, 2018. doi: 10.1007/978-3-030-03418-4_25.
8. Dominic Steinhöfel and Nathan Wasser. A New Invariant Rule for the Analysis of Loops with Non-standard Control Flows. In *Proc. 13th Intern. Conf. on Integrated Formal Methods (IFM)*, LNCS, pages 279–294, 2017. doi: 10.1007/978-3-319-66845-1_18.
9. Dominic Scheurer, Reiner Hähnle, and Richard Bubel. A General Lattice Model for Merging Symbolic Execution Branches. In Kazuhiro Ogata, Mark Lawford, and Shaoying Liu, editors, *Proc. 18th Intern. Conf. on Formal Engineering Methods (ICFEM)*, LNCS, pages 57–73, 2016. doi: 10.1007/978-3-319-47846-3_5.

Under Review

Main Author

10. Dominic Steinhöfel and Reiner Hähnle. Modal and Symbolic Trace Logic. *J. Logic and Algebraic Methods*, 2020. Under review.
11. Dominic Steinhöfel. REFINITY to Model and Prove Transformation Rules. In *Proc. 18th Asian Symposium on Programming Languages and Systems (APLAS)*, LNCS, 2020. Under review.

Coauthor

12. Reiner Hähnle, Asmae Heydari Tabar, Arya Mazaheri, Mohammad Norouzi, and Dominic Steinhöfel. Safer Parallelization. In *Proc. 10th Intern. Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA)*, 2020. Under review.
13. Elvira Albert, Alicia Merayo Corcoba, Reiner Hähnle, and Dominic Steinhöfel. Certified Abstract Cost Analysis. In *Proc. 35th Intern. Conf. on Automated Software Engineering (ASE)*, 2020. Under review.

Presentations at International Meetings

Invited Talks

Tutorial Session @ 15th Intern. Conf. on integrated Formal Methods, Bergen, Norway 2019

Title: *“How to Prove the Correctness of Refactoring Rules”*

Link: <https://ifm2019.hvl.no/refa/#pcrr>

In this tutorial session, I gave a 30-minutes talk presenting Abstract Execution and its application to proving program transformation rules. The participants used REFINITY in the second 30-minutes slot to prove the conditional correctness of two refactoring rules on their own.

International Conferences with Publication

Third World Congress on Formal Methods, Porto, Portugal 2019

Title: *“Abstract Execution”*

Link: <https://easychair.org/smart-program/FM2019/>

Second Intern. Workshop on Dynamic Logic, Porto, Portugal 2019

Title: *“The Trace Modality”*

Link: <https://workshop.dali.di.uminho.pt/>

8th ISoLA, Limassol, Cyprus 2018

Title: *“Modular, Correct Compilation with Automatic Soundness Proofs”*

Link: <http://www.isola-conference.org/isola2018/programme.html>

13th Intern. Conf. on integrated Formal Methods, Torino, Italy 2017

Title: *“A New Invariant Rule for the Analysis of Loops with Non-standard Control Flows”*

Link: <http://ifm2017.di.unito.it/program.php>

18th Intern. Conf. on Formal Engineering Methods, Tokyo, Japan 2016

Title: *“A General Lattice Model for Merging Symbolic Execution Branches”*

The website of this event is no longer accessible.

Workshops without Publication

18th Intern. KeY Symposium, Manigod, France 2019

Title: *“Abstract Execution”*

Link: <https://www.key-project.org/key-symposium-2019/>

17th Intern. KeY Symposium, Gothenburg, Sweden 2018

Title: *“Correct Compilation with Automatic Soundness Proofs”*

Link: <https://www.key-project.org/key-symposium-2018/>

PhD Symposium @ 13th Intern. Conf. on integrated Formal Methods, Torino, Italy 2017

Title: *“Assessing the Coverage of Formal Specifications”*

Link: <http://ifm2017.di.unito.it/callForPhDSymposium.php>

16th Intern. KeY Symposium, Rastatt, Germany 2017

First Talk: “*A New Invariant Rule for the Analysis of Loops with Non-standard Control Flows*”

Second Talk: “*Assessing the Coverage of Formal Specifications*”

Link: <https://www.key-project.org/key-symposium-2017/>

30th “Deduktionstreffen” of the German Informatics Society, Klagenfurt, Austria 2016

Title: “*A General Lattices Model for Merging Symbolic Execution Branches*”

The website of this event is no longer accessible.

14th Intern. KeY Symposium, Gothenburg, Sweden 2015

Title: “*A General Lattices Model for Merging Symbolic Execution Branches*”

Link: <https://www.key-project.org/keysymposium15/>

Teaching Experience

Supervised Theses

So far, I co-supervised three B. Sc. and two M. Sc. theses at TU Darmstadt.

Courses

Software Engineering Exercises

The lecture “Software Engineering” is a mandatory course for Bachelor students of computer science at Technische Universität Darmstadt. It consists of a weekly, 90-minutes lecture and a 45-minutes exercise session held directly after the lecture in the same room. In the winter terms 2016 and 2017, I was responsible for the exercises, which included preparing the exercise sheets, chairing the weekly exercise sessions, and administrating and coaching student assistants who offered weekly consultation hours and corrected exercise sheets with bonus assignments. In 2017, 500 students registered for the exam of the course.

Before I took over the exercises in 2016, the exercise sessions were presentations of solution proposals for the exercise sheets. Since individual discussions are impossible in this setting, the sessions were unpopular among students and poorly attended. I changed the format as follows: 1. The main part of an exercise session is taken by a “micro lecture” of selected highlights considered particularly difficult or interesting by either me or the attending students. 2. Instead of presenting all exercise solutions with slides only, I likewise picked difficult parts and discussed them in detail using slides, the blackboard, and live demos. 3. I added interactive elements like quizzes and (live and online) polls to make the session more attractive.

These measures helped to increase the appreciation of the Software Engineering exercises (although the course was assigned a Friday afternoon slot in 2016 and 2017). At the last official evaluation *before* 2016, the average grade students assigned to the exercises was 2.57, which improved to 2.46 in 2016 and 2.43 in 2017. The average grade in an anonymous online poll among students actively participating in the exercises was 1.96 in 2017. The detailed feedback of this poll confirms the effectiveness of the concept of “micro lectures” and focused discussions; the most frequent complaints concerned time and location of the course.

Bachelorpraktikum

The Bachelorpraktikum (english “bachelor lab”) is a mandatory course for 5-th semester Bachelor students of computer science at Technische Universität Darmstadt. Teams of four to five students realize, during a whole 6-month term, software projects offered by members of the university (mostly, but not exclusively, of the computer science department). I assisted in the organization of the Bachelorpraktikum in the summer and winter term 2015/2016 and in the winter term 2017, with responsibilities such as the assignments of students to teams or the communication with student assistants (“team leaders”) in case of problems with

the teamwork. I additionally contributed to the course by developing a grading rubric for quality assurance documents produced by the students, and by creating a complex online administration software “BP Admin” especially for the course. As of 2020, BP Admin is still being used by other groups organizing this course.

Seminars

I supervised several participants of the seminars “*Software Project Failures*” (2017), “*Symbolic Execution*” (2018) and “*Actor-Based Languages*” (2019).

Services to the Scientific Community

Organization of Events

First Intern. HacKeYthon, Karlsruhe, Germany 2018

The “HacKeYthon” is an event with the goal to bring forward the development of the KeY program verification framework, and, at the same time, unite experienced developers, new project members and associates as well as interested students. Participants work in small groups, which are mixed in terms of experience to enable knowledge transfer, on chosen programming projects related to KeY. The idea of the HacKeYthon was originally conceived by me in 2017, and I was the main organizer of the first such event in 2018.

Link: <https://www.key-project.org/1st-hackeython-2018/>

15th Intern. KeY Symposium, Manigod, France 2016

I organized the 15th KeY Symposium in Manigod, France. This included communication with participants (emails, website), assembling the program, and organizational issues regarding the venue (rooms, catering).

Link: <http://i12www.ira.uka.de/key/keysymposium16/>

Reviewing Activities

I have been reviewing articles for the journals FAOC (2017) and JLAMP (2020), and the conferences TAP (2016 and 2018), SAS (2017), CPP (2017 and 2018), ISoLA (2018), FASE (2018 and 2020), DaLí and TABLEAUX (2019), and CONCUR and SEFM (2020).

Languages

German (native), English (fully proficient), Spanish (advanced), French (intermediate).