Ömer Şakar

PhD Student · ChEOPS project

Enschede, Netherlands

(+31) 53 489 7473 | work: o.f.o.sakar@utwente.nl, personal: o.f.o.sakar@gmail.com | OmerSakar

Education

PhD Student at the University of Twente

Enschede, Netherlands

University of Twente

Jun. 2020 - Present

- ChEOPS project: Integrating Verification into GPU Program Development
- Formal Methods and Tools group
- Faculty of Electrical Engineering, Mathematics and Computer Science

Master of Science in Technical Computer Science (Technische Informatica)

Enschede, Netherlands

University of Twente

Nov. 2017 - April 2020

- Specialization: *Software Technology*
- Title of Thesis: *Extending support for axiomatic data types in VerCors*.
- Grade: 9.0

Bachelor of Science in Technical Computer Science (Technische Informatica)

Enschede, Netherlands

University of Twente

Sept. 2014 - Nov. 2017

- Title of Thesis: *Correlating the 2012 Dutch House of Representatives Elections based on Twitter mentions of Parties and their Party Leader*.
- Grade: 8.0

Work Experience

PhD Student at the University of Twente

Enschede, Netherlands

University of Twente

Jun. 2020 - Present

- ChEOPS project: Integrating Verification into GPU Program Development
- Formal Methods and Tools group
- Faculty of Electrical Engineering, Mathematics and Computer Science

Teaching Assistant (TA, Student Assistent)

Enschede, Netherlands

University of Twente

2016 - 2020

- Teaching Assistant for the Bachelor Technical Computer Science courses.
- TA during the following modules. Modules are comprised of multiple courses following a general topic.
  - Pearls of Computer Science (2016/2017, 2017/2018, 2018/2019, 2019/2020). An introduction to Computer Science.
  - Software Systems (2017/2018, 2018/2019, 2019/2020). Introduction to designing, implementing and testing software.
  - Network Systems (2017/2018, 2018/2019). A module with courses on computer networking.
  - Data & Information (2017/2018, 2018/2019). A module with courses on structured/unstructured data, databases, scripting frameworks and services such as RESTful services.

Developer Quizzard

Enschede, Netherlands

University of Twente

Feb. 2017 - May 2019

- Quizzard: A teaching quiz application.
- Mainly back end development.
- Technologies used: Java, Maven, PostgreSQL, Spring Boot, Thymeleaf, Hibernate, React.
- GitLab page: `https://git.tworem.nl/tnb/quizzard`

Back end developer Groufty

Enschede, Netherlands

University of Twente

Feb. 2016 - June 2016

- Groufty: group peer review system.
- Technologies used: Java, Maven, PostgreSQL, Spring Boot, Thymeleaf, Hibernate.
- GitHub page: `https://github.com/utwente/Groufty`

Skills

Familiar with (at least) the following technologies:

Programming languages	Most experience with Java, Python and Kotlin. Some experience with C, C++, Rust and Scala. Played around with C#, Go, Ruby, JavaScript, R and many more.
Frameworks	Spring Framework/Boot, Hibernate, JOOQ
Databases	PostgreSQL, H2
Tools	Git, GitHub, GitLab, YouTrack, Trello, Jira
DevOps	Docker, Docker Registry, GitLab CI, Travis CI
Operating Systems	Linux (Ubuntu 14.04+, Rasbian), Windows XP/Vista/7/8/10

Interests and Hobbies

In my free time I like to do the following (in no particular order); Tinkering with hardware and software. Amateur server maintainer. Reading on history, religion, ethics and Computer Science related pieces in general (either Dutch, Turkish or English). Amateur woodworker. Nostalgic gamer. Logic puzzles such as sudokus, nonograms (also called Japanese crosswords or Japanese picture puzzles) and futoshikis and cube puzzles such as the Rubik's cube. Researching things I do not understand (yet). Learning in general. Teaching others to spread knowledge.

Publications

Alpinist: an Annotation-Aware GPU Program

Munich, Germany

TACAS 2022 *April 2–7, 2022*

- Authors: Ömer Şakar, Mohsen Safari, Marieke Huisman, Anton Wijs
- Abstract: Over the last years, deductive program verifiers have substantially improved, and their applicability on non-trivial applications has been demonstrated. However, a major bottleneck is that for every new programming language, a new deductive verifier has to be built. This paper describes the first steps in a project that aims to address this problem, by language-agnostic support for deductive verification: Rather than building a deductive program verifier for every programming language, we develop deductive program verification technology for a widely-used intermediate representation language (LLVM IR), such that we eventually get verification support for any language that can be compiled into the LLVM IR format. Concretely, this paper describes the design of VCLLVM, a prototype tool that adds LLVM IR as a supported language to the VerCors verifier. We discuss the challenges that have to be addressed to develop verification support for such a low-level language. Moreover, we also sketch how we envisage to build verification support for any specified source program that can be compiled into LLVM IR on top of VCLLVM.
- DOI: 10.1007/978-3-030-99527-0_18

First Steps towards Deductive Verification of LLVM IR

Luxembourg City, Luxembourg

FASE 2024 *April 8–11, 2024*

- Authors: Dré van Oorschot, Marieke Huisman, Ömer Şakar
- Abstract: Over the last years, deductive program verifiers have substantially improved, and their applicability on non-trivial applications has been demonstrated. However, a major bottleneck is that for every new programming language, a new deductive verifier has to be built. This paper describes the first steps in a project that aims to address this problem, by language-agnostic support for deductive verification: Rather than building a deductive program verifier for every programming language, we develop deductive program verification technology for a widely-used intermediate representation language (LLVM IR), such that we eventually get verification support for any language that can be compiled into the LLVM IR format. Concretely, this paper describes the design of VCLLVM, a prototype tool that adds LLVM IR as a supported language to the VerCors verifier. We discuss the challenges that have to be addressed to develop verification support for such a low-level language. Moreover, we also sketch how we envisage to build verification support for any specified source program that can be compiled into LLVM IR on top of VCLLVM.
- DOI: 10.1007/978-3-031-57259-3_15