

# Dominik Wagner

Magdalen College – Oxford OX1 4AU, United Kingdom

✉ dominik.wagner@cs.ox.ac.uk • 🌐 domwagner.github.io

## Research Interests

---

- Probabilistic programming
- Semantics of programming languages
- Automated reasoning and verification of (higher-order) programs

## Education

---

**DPhil (PhD) in Computer Science** Oct 2018 – present

*University of Oxford, Magdalen College, Oxford*

- Supervisor: Prof. Luke Ong

**MSc in Mathematics and Foundations of Computer Science** Oct 2017 – Sep 2018

*University of Oxford, Magdalen College, Oxford, with Distinction*

- Thesis: “**Resolution for Higher-Order Constrained Horn Clauses**”  
Supervisor: Prof. Luke Ong
- Selected coursework: Lambda Calculus and Types; Computer-aided Formal Verification; Automata, Logic and Games; Categories, Proofs and Processes; Model Theory

**Preparatory phase of the Saarbrücken Graduate School of Computer Science** May – Sep 2017

*Saarland University, Saarbrücken*

**BSc in Computer Science** (minor: Mathematics) Apr 2014 – Apr 2017

*Saarland University, Saarbrücken, Grade: 1.0 (best on a scale from 1.0 to 5.0)*

- FdSI Bachelor Award for best overall performance (up to 3 recipients/semester)
- Thesis: “**Design and Implementation of a CDCL(LA) Calculus**”  
Supervisor: Prof. Christoph Weidenbach
- Selected coursework: Automated Reasoning I/II; Computational Logic I/II; Artificial Intelligence

## Research and Development Experience

---

**Student Assistant** Nov 2014 – Sep 2017

*Max Planck Institute for Informatics, Saarbrücken*

- Development of a modern CDCL-based SAT-solver used in the ground linear arithmetic solver SPASS-SATT<sup>1</sup>
- Focus on redundancy deletion and generation of unsatisfiability proofs
- Experience in writing highly efficient C-code using debugging/profiling tools like gdb, valgrind, gprof, etc.
- SPASS-SATT won the tracks “QF\_LIA” and “Best Newcomer” in the SMT Competition 2018
- Supervisor: Prof. Christoph Weidenbach

---

<sup>1</sup><https://www.mpi-inf.mpg.de/departments/automation-of-logic/software/spass-workbench/spass-satt/>

## Publications

---

C.-H. Luke Ong and **Dominik Wagner**. HoCHC: A refutationally complete and semantically invariant system of higher-order logic modulo theories. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–14, 2019.

## Presentations

---

“HoCHC: A refutationally complete and semantically invariant system of higher-order logic modulo theories”. At *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver*, June 2019.

“HoCHC: A refutationally complete and semantically invariant system of higher-order logic modulo theories.” At *6th Workshop on Horn Clauses for Verification and Synthesis, HCVS 2019, Prague*, April 2019.

Poster on unsatisfiability proofs as implemented in SPASS-SATT. At *International Summer School on Satisfiability, Satisfiability Modulo Theories, and Automated Reasoning, Lisbon, Portugal*, June 2016.

## Academic Service

---

**Reviewer:** LICS 2019

**Student volunteer:** FLoC 2018, POPL 2019, ETAPS 2019

## Awards and Funding (selection)

---

**FdSI Bachelor Award** 2017

Best overall performance in the BSc programme of Saarland University (up to 3 recipients/semester)

**Scholarship holder of the German Academic Scholarship Foundation** 2015 – 2018

Financial and academic support (e.g. summer academies) of less than 0.5% of German students

**Scholarship holder of the German Academic Exchange Service** 2017 – 2018

Full study abroad scholarship awarded to approx 1,200 German students

## Teaching

---

**Automata, Logic and Games** Michaelmas Term 2019

*University of Oxford*, Teaching Assistant

**Fundamentals of Algorithms and Data Structures** winter 2016/17

*Saarland University*, Teaching Assistant

**Mathematical Preparatory Course** Sep/Oct 2016/17

*Saarland University*, Mentor

## Programming Skills

---

**Imperative languages:** Java, C (*used in the implementation of a modern SAT solver*)

**Functional languages:** Haskell, OCaml, SML

**Interactive theorem provers:** Coq

**Tools:** git, gdb, valgrind, etc.