Samuel Gruetter: Curriculum Vitae

I am a postdoc working with Prof. Timothy (Mothy) Roscoe's group, in the Systems Group at ETH Zürich.

Education

09/2017 – 09/2024 PhD in Computer Science at MIT, working with Prof. Adam Chlipala’s Programming Languages and Verification group

02/2014 – 04/2017 MSc in Computer Science from the Swiss Federal Institute of Technology in Lausanne (EPFL), specialization in “Foundations of Software”. Participated in EPFL's MSc Research Scholars Program, working part-time as a research assistant at Prof. Martin Odersky's Programming Methods Lab (the “Scala Lab”)

10/2016 – 03/2017 MSc thesis project at Prof. Andrew Appel's lab at Princeton University

06/2014 Oregon Programming Languages Summer School on Types, Logic, Semantics, and Verification, at University of Oregon

09/2010 – 06/2013 Bachelor in Computer Science at EPFL

Research Experience

C Live Verification A framework for proving correctness of programs in a C-like language (Bedrock2). The user writes the program and the proof at the same time, aided by a real-time display of the program’s current symbolic state [PLDI'24]

Bedrock2 end-to-end I wrote a compiler from a simple C-like language to RISC-V machine code, proved it correct in the Coq proof assistant, and used it to prove end-to-end system correctness theorems covering whole software-hardware stacks [PLDI'21 & '24]

C information flow I was visiting Dr. Toby Murray at the University of Melbourne for 10 weeks to work on information flow control proofs for C [PLAS'17]

Verifying AES For a six months master thesis internship, I was working with Prof. Andrew Appel’s group at Princeton, improving the proof automation tactics of their Verified Software Toolchain, and using it to verify AES encryption

DOT During my master’s at EPFL, I was working with Prof. Martin Odersky’s Scala lab on the Dependent Object Types project, a formalization of the core of Scala’s type system, writing proofs on paper and in Twelf and Coq

Leon termination For a class project at EPFL, I contributed to the function termination checker of Leon, a tool for verification and synthesis of Scala programs by Prof. Viktor Kuncak’s group

Dotty While working at the Scala lab, I contributed to dotty, a new Scala compiler serving as a research platform to investigate new language concepts and compiler technologies for Scala

Structural Types For my bachelor thesis, I designed, explored and implemented a simple structurally typed language in PLT redex

Publications

ITP 2024 Samuel Gruetter, Thomas Bourgeat, and Adam Chlipala. Verifying Software Emulation of an Unsupported Hardware Instruction. *15th International Conference on Interactive Theorem Proving*, September 2024.

PLDI 2024 Samuel Gruetter, Viktor Fukala, and Adam Chlipala. Live Verification in an Interactive Proof Assistant. *Proceedings of the ACM on Programming Languages*, 8(PLDI), June 2024.

PLDI 2024 Andres Erbsen, Jade Philipoom, Dustin Jamner, Ashley Lin, Samuel Gruetter, Clément Pit-Claudel, and Adam Chlipala. Foundational Integration Verification of a Cryptographic Server. *Proceedings of the ACM on Programming Languages*, 8(PLDI), June 2024.

ICFP 2023 Thomas Bourgeat, Ian Clester, Andres Erbsen, Samuel Gruetter, Pratap Singh, Andrew Wright, and Adam Chlipala. Flexible Instruction-Set Semantics via Abstract Monads (Experience Report). *Proceedings of the ACM on Programming Languages*, 7(ICFP), August 2023.

TOPLAS 2023 Arthur Charguéraud, Adam Chlipala, Andres Erbsen, and Samuel Gruetter. Omnisemantics: Smooth Handling of Nondeterminism. *ACM Transactions on Programming Languages and Systems 45(1)*, March 2023.

PLDI 2021 Andres Erbsen, Samuel Gruetter, Joonwon Choi, Clark Wood, and Adam Chlipala. Integration Verification Across Software and Hardware for a Simple Embedded System. *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, June 2021.

JAR 2018 Qinxiang Cao, Lennart Beringer, Samuel Gruetter, Josiah Dodds, and Andrew W. Appel. VST-Floyd: A Separation Logic Tool to Verify Correctness of C Programs. *Journal of Automated Reasoning*, 61(1-4), June 2018.

PLAS 2017 Samuel Gruetter and Toby Murray. Short Paper: Towards Information Flow Reasoning about Real-World C Code. *Proceedings of the 2017 Workshop on Programming Languages and Analysis for Security*, October 2017.

WadlerFest 2016 Nada Amin, Samuel Gruetter, Martin Odersky, Tiark Rompf, and Sandro Stucki. The Essence of Dependent Object Types. In *WadlerFest*, Springer LNCS 9600, 2016.

Industry Internships

Google, 2021 In the Silver Oak Project, used Bedrock2 to formally verify drivers for peripherals used in the OpenTitan root of trust, and connected software correctness proofs to hardware correctness proofs

Amazon ARG, 2019 Worked with Rustan Leino at Amazon's Automated Reasoning Group on a prototype rewrite of Amazon's S3 Encryption Client in Dafny, a verification-aware programming language. Wrote and proved specifications for software interacting with real-world systems such as Amazon's S3 storage service

Netcetera, 2015 6 months Software Engineering Internship at Netcetera AG, Berne, working in a scrum team, developing an expert tool for defining and maintaining the fare zones and ticket pricing for all Swiss public transport associations, with a Java/Oracle DB/Spring backend and an AngularJS frontend in JavaScript/TypeScript

Accenture, 2012 Java Summer Internship at Accenture in Bangalore (India), developed a web interface with JSF/Enterprise JavaBeans monitoring servers and databases

Mentoring

At MIT, I mentored 12 undergraduate and MEng students on projects related to our group's research:

- Thomas Carotti, Pratap Singh (now PhD student at CMU):
  Runtime metrics bounds for the Bedrock2 compiler
- Christian Altamirano (now PhD student at Yale):
  Formally verified implementation of the Roughtime protocol in Bedrock2
- Arthur Reiner De Belen:
  Better instruction selection and dead-code elimination for the Bedrock2 compiler
- Leo Gagnon, Pratyush Venkatakrishnan, Mohit Hulse, Samuel Tian, Andrew Spears, Michelle Touma:
  Fiat2, a new high-level language for the Bedrock2 ecosystem
- Pleng Chomphoochan:
  Functional implementation and verification of crit-bit trees and Live Verification sample programs
- Viktor Fukala:
  Formally verified low-level C implementation of crit-bit trees using Live Verification
  Ranked 2nd at the PLDI'24 Student Research Competition

Service

<Programming> '22	External reviewer, outstanding reviewer awardee
Scala Symposium '22	Program Committee
OOPSLA'22	Extended Review and Artifact Evaluation Committee
CPP'20	External reviewer
GPCE'17	External reviewer

Talks

- Foundational End-to-End Verification of Systems Stacks
  - Virtual talk at Cornell Programming Languages Discussion Group, September 2024
- Verifying Software Emulation of an Unsupported Hardware Instruction
  - Virtual talk at ITP, September 2024
- Foundational Integration Verification of a Cryptographic Server
  - PLDI, June 2024
- Live Verification of C Programs in Coq
  - MPI-SWS, Saarbrücken, July 2024
  - PLDI, June 2024
  - ETH Zurich, June 2024
  - New England Systems Verification Day, April 2024
  - KU Leuven, January 2024
  - ConVeY Seminar at Ludwig-Maximilians-Universität Munich, January 2024
- Flexible Instruction-Set Semantics via Abstract Monads (Experience Report)
  - ICFP, September 2023
- Silver Oak: Hardware Software Co-Design and Co-Verification in Coq
  - Workshop on Programming Languages for Architecture (PLARCH), June 2023
- Omnisemantics: Smooth Handling of Nondeterminism
  - TU Munich, January 2024
  - OderskyFest, EPFL, September 2023
  - Keynote at CoqPL, January 2023
  - Harvard PL Seminar, October 2022
  - New England Programming Languages and Systems Symposium (NEPLS), September 2022
- E-Graphs to Help Writing Coq Proofs
  - New England Systems Verification Day, October 2022
- Semantics for Verified Software-Hardware Stacks
  - Guest lecture at Harvard class CS 152: Programming Languages, April 2022
  - Virtual guest lecture at Harvard class CS 152: Programming Languages, April 2021
- Integration Verification across Software and Hardware for a Simple Embedded System
  - Virtual talk at PLDI, June 2021
- Introduction to Proof Scripting and the Ltac Language
  - Replacement lecturer at MIT class 6.822, February 2020
- Formal Methods for Hardware-Software Integration on RISC-V Embedded Systems
  - RISC-V Summit, December 2019
- Counterexamples for Coq Conjectures
  - CoqPL, January 2019
- A Quick Hack to ask any SMT Solver if my Coq Goal is True
  - DeepSpec Workshop, June 2018
- Towards Information Flow Reasoning about Real-World C Code
  - Workshop on Programming Languages and Analysis for Security (PLAS), October 2017
- Machine checked formal reasoning about the behavior of programs
  - Guest lecture at University of Melbourne class High Integrity Systems Engineering, May 2017

Teaching Experience

MIT FRAP TA — Teaching assistant for the “Formal Reasoning about Programs” course at MIT. Designed and graded problem sets, held office hours and recitations

MOOC TA — Teaching assistant for the “Principles of Reactive Programming” course on Coursera, a massive open online course with more than 40'000 students. Developed RxScala, the library on which the programming assignments were based, helped develop and test the assignments, and answered forum questions

EPFL TA — Teaching assistant for the BSc class “Introduction to Logic Systems”, helping students with questions about the exercises

SOI lecturer — Gave lectures at workshops of the Swiss Olympiad in Informatics, teaching basic algorithms (such as graphs, scanline, dynamic programming) to high schoolers

Awards

MIT Fellowship 2017	Presidential Graduate Fellowship by MIT
hc2 2013	Ranked 3rd at Helvetic Coding Contest
SWERC 2012	Ranked 7th at Southwestern Europe Regional Contest of ACM International Collegiate Programming Contest
SOI 2010	Ranked 1st at Swiss Olympiad in Informatics
SPO 2010	Ranked 1st at Swiss Olympiad in Philosophy

Opensource Experience

RxScala — Main contributor of RxScala (Reactive Extensions for Scala), a library for composing asynchronous and event-based programs using observable sequences. RxScala is an adapter for the RxJava library by Netflix. Integrated into the Netflix repository in 2013

Other

Study Foundation — Admitted to the complementary learning program of the Swiss Study Foundation

hc2 organizer — Helped organize the Helvetic Coding Contest 2014

SOI organizer — Helped organize the Swiss Olympiad in Informatics 2011-2016, leader of the Swiss delegation to the International Olympiad in Informatics 2013

Languages

German	native
English	fluent (TOEFL: 107/120, Cambridge Certificate of Proficiency in English)
French	fluent
Latin	took 5 years of Latin in high school

Contact

Office address
ETH Zurich Systems Group
STF H 313
Stampfenbachstrasse 114
8092 Zürich
Switzerland

E-Mail samuel.gruetter@inf.ethz.ch