

Samuel Gruetter: Curriculum Vitae

Education

Since Fall 2017	PhD candidate in Computer Science at MIT, working with Prof. Adam Chlipala's Programming Languages and Verification group
April 2017	MSc in Computer Science from the Swiss Federal Institute of Technology in Lausanne (EPFL), specialization in "Foundations of Software", GPA: 5.80*
10/2016 – 03/2017	MSc thesis project at Prof. Andrew Appel's lab at Princeton University
2014 – Summer 2015	3 semesters of MSc Research Scholars Program at EPFL: Taking the standard Master's program in Computer Science and in parallel, working part-time as a research assistant at Prof. Martin Odersky's Programming Methods Lab (the "Scala Lab")
Summer 2014	Oregon Programming Languages Summer School on Types, Logic, Semantics, and Verification, at University of Oregon
2010 – 2013	Bachelor in Computer Science at EPFL, GPA: 5.51*
2006 – 2010	High school at Gymnasium Bern-Kirchenfeld, Switzerland, type Maths/Physics, GPA: 5.81*. Spring semester 2008 at Gymnase Auguste Piccard, Lausanne

Research Experience

bedrock2	Currently, I'm working on a verified compiler written in Coq from a very simple C-like language to RISC-V machine code [9]
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 [8]
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 the AES encryption implementation of mbed TLS [7]
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 using the proof assistants Twelf and Coq [3, 4, 6]
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 [5]
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 [1]

Publications

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. In <i>Journal of Automated Reasoning</i> , 61(1-4) pp 367-422, June 2018.
PLAS 2017	Samuel Gruetter and Toby Murray. Short Paper: Towards Information Flow Reasoning about Real-World C Code. In <i>Proceedings of the 2017 Workshop on Programming Languages and Analysis for Security - PLAS '17</i> , pp 43-48, Dallas, Texas, USA, 2017. ACM Press.
WadlerFest 2016	N. Amin, S. Grütter, M. Odersky, T. Rompf, and S. Stucki. The essence of dependent object types. In <i>WadlerFest</i> , 2016. Springer LNCS 9600, pp 249-272.
IOI Journal 2016	S. Grütter, D. Graf, and B. Schmid. Watch them Fight! Creativity Task Tournaments of the Swiss Olympiad in Informatics. In <i>Olympiads in Informatics</i> , 2016, Vol. 10, pp 73-85.

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

Industry Experience

Accenture, 2012	Java Summer Internship at Accenture in Bangalore (India), developed a web interface with JSF/Enterprise JavaBeans monitoring servers and databases
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 zone plans and ticket pricing for all Swiss public transport associations, with a Java/Oracle DB/Spring backend and an AngularJS frontend being migrated from JavaScript to TypeScript

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 [2] in 2013
---------	--

Teaching Experience

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

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, finished with a Latin grade of 6*

Contact

Permanent address	Mattenstrasse 19a, 3073 Gümligen, Switzerland
US address	67a Dana St Apt 2, Cambridge MA 02138, USA
E-Mail	gruetter@mit.edu

* Swiss grades: 1 = lowest, 4 = pass, 6 = best

Links

- [1] BSc semester project "Explorations of type systems", Spring 2013
<https://github.com/samuelgruetter/type-systems-spring13/blob/master/doc/report.pdf>
- [2] RxScala (Reactive Extensions for Scala)
<https://github.com/ReactiveX/RxScala>
- [3] MSc semester project "Machine-checked typesafety proofs", Spring 2014
<https://github.com/samuelgruetter/typesafety-proofs-spring14/blob/master/report.pdf>
- [4] Report "Dependent Object Types With Existential Quantification Over Objects", July 2015
<https://github.com/samuelgruetter/dot-calculus/tree/master/doc/gDOT-and-exDOT>
- [5] Report "Improving Leon's Termination Checker", June 2015
<https://documents.epfl.ch/users/k/ks/ksgruett/www/files/LeonTermination.pdf>
- [6] MSc optional semester project "Connecting Scala to DOT", Spring 2016
<https://github.com/samuelgruetter/dot-calculus/blob/master/doc/Connecting-Scala-to-DOT>
- [7] MSc thesis "Improving the Coq proof automation tactics of the Verified Software Toolchain, based on a case study on verifying a C implementation of the AES encryption algorithm", Spring 2017
<https://www.cs.princeton.edu/research/techreps/TR-999-17>
- [8] arXiv report "VST-Flow: Fine-grained low-level reasoning about real-world C code", Summer 2017
<https://arxiv.org/abs/1709.05243>
- [9] Work in progress: "Towards a low-level systems programming language with a verified compiler"
<https://github.com/mit-plv/bedrock2>