Curriculum Vitae

Martin Avanzini

Given Name	Martin
Family Name	Avanzini
Date / Place of Birth	July 30, 1982 / Hall in Tirol
Nationality	Austria
Marital Status	married
Personal Address	Schnatzenbichl 3, A-6063 Rum

Research Interests

- Functional programming
- Rewriting
- Static program analysis
- Complexity analysis
- Implicit computational complexity

Higher Education

- **PhD Degree in Computer Science.** *2013*
  Institute of Computer Science, University of Innsbruck, Austria.
  Thesis *Verifying Polytime Computability Automatically*, supervised by Georg Moser.

- **Master's Degree in Computer Science.** *2009*
  Institute of Computer Science, University of Innsbruck, Austria.
  Master Thesis *Automation of Polynomial Path Orders*, supervised by Georg Moser.

- **Bachelor's Degree in Computer Science.** *2007*
  Institute of Computer Science, University of Innsbruck, Austria.
  Bachelor Thesis *Termination Analysis for Scheme using S-Expression Rewrite Systems*, supervised by Nao Hirokawa and *Scheme Programs with Polynomially Bounded Evaluation Length* supervised by Georg Moser.

- **Diploma Höhere Technische Lehranstalt.** *2001*
  Civil Engineering, Htl Trenkwalderstraße, Innsbruck, Austria.

Awards

- **Proposed for the Heinz Zemanek Price.** *October, 2016*
  The *Heinz Zemanek price* is awarded every 3 years by the *Austrian Computer Society (OCG)* to young researchers for outstanding PhD dissertations. I was nominated by the University of Innsbruck for this price, and also passed the final selection (8 persons) from the OCG.

- **Kurt Gödel Medal.** *August, 2014*
  Our *Tyrolean Complexity tool* was distinguished with the prestigious *Kurt Gödel Medal* as best tool for the complexity analysis of term rewrite systems at the *FLoC Olympic Games*, held during the *Vienna Summer of Logic*.

- **European Summer School in Logics, Languages and Computation.** *August, 2008*
  My work received second place in *Springer best paper awards*.

Scholarships and Projects

- **Principle investigator.** *April, 2014 – May, 2017*
  Higher-Order Complexity Analysis of Rewrite Systems.
  FWF Schrödinger Fellowship (project number J-3563). *University of Bologna / University of Innsbruck*.

- **Project member.** *June, 2015 – June, 2016*
  Concurrent, Resourceful and Effectful COmputation, by Geometry of Interaction.
  Multilateral research project with members from France, Italy and Japan. *University of Bologna / INRIA Sophia-Antipolis*.

- **Research assistant.** *December, 2013 – March, 2014*
  Improving Certifiers for Termination Proofs.
  FWF standalone project (project number P22767). *University of Innsbruck*.

- **Research assistant.** *November, 2012 – May, 2013*
  Structural and Computational Proof Theory.
  Bilateral research project between ANR and FWF (project number I608-N18). *University of Innsbruck*.

- **Principal investigator.** *November, 2011 – October, 2012*
  Automated Complexity Analysis.
  Doctoral fellowship (project number NWF-2011/2/Mip7). *University of Innsbruck*.

- **Research assistant.** *October, 2008 – August, 2011*
  Derivational Complexity Analysis.
  FWF standalone project (project number P20133). *University of Innsbruck*.

Scientific Activities

- **Organiser.** *2017*
  9th Workshop on Developments in Implicit Computational Complexity, Thessaloniki, Greece.
- **PC member.** *2017*
  17th International Workshop on Logic and Computational Complexity, Reykjavik, Island.
- **PC member.** *2014*
  5th Workshop on Developments in Implicit Computational Complexity, Grenoble, France.
- **Invited speaker.** *2013*
  15th International Workshop on Logic and Computational Complexity, Torino, Italy.
- **Invited speaker.** *2013*
  3rd Workshop on Proof Theory and Rewriting, Kanazawa, Japan.

Software Development

The following gives a short list of most important software projects that I was involved in. If not mentioned otherwise, I am (among) the main developer(s). Details can be found at my software page.

- **IsaFoR/CeTA**: A formally verified tool for checking termination, confluence and complexity proofs. I have contributed the formalisation of dependency tuples.
- **GUBS**: A constraint solver for polynomial inequalities.
- **Higher Order Complexity Analysis (HOCA)**: Frontend for analysing the runtime complexity of OCaml programs through first-order tools.
- **Higher Order Sized-Type Analysis (HOSA)**: Complexity analyser of higher-order programs through sized-type analysis and program instrumentation.
- **Implicit Computational Complexity Tool (ICCT)**: Analyses the complexity of functions defined through rewrite systems.
- **Tyrolean Complexity Tool (TCT)**: Full-fledged runtime complexity analyser.