

Ross Tate — Assistant Professor in the Department of Computer Science at Cornell University

CONTACT INFORMATION 434 Gates Hall
Cornell University
Ithaca, NY 14853-7501
E-mail: ross@cs.cornell.edu
WWW: <http://www.cs.cornell.edu/~ross/>

CITIZENSHIP USA, Canada

RESEARCH INTERESTS Language Design, Type Theory, Effects, Semantics, Mathematical Foundations

EDUCATION **University of California, San Diego, CA USA**
Ph.D., Computer Science and Engineering, 2012
Equality Saturation: Using Equational Reasoning to Optimize Imperative Functions
M.S., Computer Science and Engineering, 2009
Advised by Professor Sorin Lerner
California Polytechnic State University, San Luis Obispo, CA USA
B.S., Mathematics, 2006
B.S., Computer Science, 2006
Advised by Professor Aaron Keen

ACADEMIC EMPLOYMENT Assistant Professor in Computer Science at Cornell University, 2012–present

GRADUATE FIELDS Computer Science

EXTERNAL TITLES Director of The Single Open Intermediate Language Initiative
Type-System Advisor for Kotlin
Type-System Advisor for Ceylon

AWARDS Distinguished Paper Award for Sound Gradual Typing is Nominally Alive and Well, 2017
Dahl-Nygaard Junior Prize, 2017
Distinguished Artifact Award for Java and Scala’s Type Systems are Unsound, 2016
Microsoft Research Fellowship, 2009

REFEREED CONFERENCE PUBLICATIONS Liron Cohen, Sofia Abreu Faro, and Ross Tate. The Effects of Effects on Constructivism. *MFPS '19: Conference on the Mathematical Foundations of Programming Semantics*.
Fabian Muehlboeck and Ross Tate. Empowering Union and Intersection Types with Integrated Subtyping. *OOPSLA '18: Object-Oriented Programming, Systems, Languages, and Applications*.
Andrew K. Hirsch and Ross Tate. Strict and Lazy Semantics for Effects: Layering Monads and Comonads. *ICFP '18: International Conference on Functional Programming*.
Fabian Muehlboeck and Ross Tate. Sound Gradual Typing is Nominally Alive and Well. *OOPSLA '17: Object-Oriented Programming, Systems, Languages, and Applications*.
Nada Amin and Ross Tate. Java and Scala’s Type Systems are Unsound. *OOPSLA '16: Object-Oriented Programming, Systems, Languages, and Applications*.

	<p>Stephen Longfield, Brittany Nkounkou, Rajit Manohar, and Ross Tate. Preventing Glitches and Short Circuits in High-Level Self-Timed Chip Specifications. <i>PLDI '15: Programming Language Design and Implementation</i>.</p> <p>Ben Greenman, Fabian Muehlboeck, and Ross Tate. Getting F-Bounded Polymorphism into Shape. <i>PLDI '14: Programming Language Design and Implementation</i>.</p> <p>Ross Tate. The Sequential Semantics of Producer Effect Systems. <i>POPL '13: Principles of Programming Languages</i>.</p> <p>Ross Tate, Alan Leung, and Sorin Lerner. Taming Wildcards in Java's Type System. <i>PLDI '11: Programming Language Design and Implementation</i>.</p> <p>Mike Stepp, Ross Tate, and Sorin Lerner. Equality-based Translation Validator for LLVM. <i>CAV '11: Computer Aided Verification</i>.</p> <p>Ross Tate, Juan Chen, and Chris Hawblitzel. Inferable Object-Oriented Typed Assembly Language. <i>PLDI '10: Programming Language Design and Implementation</i>.</p> <p>Ross Tate, Mike Stepp, and Sorin Lerner. Generating Compiler Optimizations from Proofs. <i>POPL '10: Principles of Programming Languages</i>.</p> <p>Ross Tate, Mike Stepp, Zachary Tatlock, and Sorin Lerner. Equality Saturation: a New Approach to Optimization. <i>POPL '09: Principles of Programming Languages</i>.</p>
JOURNAL PUBLICATIONS	<p>Ross Tate, Mike Stepp, Zachary Tatlock, and Sorin Lerner. Equality Saturation: a New Approach to Optimization. <i>LMCS-7(1:10) '11: Logical Methods in Computer Science</i>.</p>
REFEREED WORKSHOP PUBLICATIONS	<p>Andrew K. Hirsch, Pedro H. Azevedo de Amorim, Ethan Cecchetti, Ross Tate, and Owen Arden. First-Order Logic for Flow-Limited Authorization. <i>FCS '19: Workshop on Foundations of Computer Security</i>.</p> <p>Ross Tate. Mixed-Site Variance. <i>FOOL '13: Foundations of Object-Oriented Languages</i>.</p>
TECHNICAL REPORTS	<p>Ross Tate, Juan Chen, and Chris Hawblitzel. Inferable Existential Quantification. Microsoft Research, 2011.</p> <p>Ross Tate and Daan Leijen. Convenient Explicit Effects using Type Inference with Subeffects. Microsoft Research, 2010.</p> <p>Ross Tate, Juan Chen, and Chris Hawblitzel. A Flexible Framework for Type Inference with Existential Quantification. Microsoft Research, 2008.</p>
KEYNOTES	<p>Retargeting Gradual Typing. <i>ECOOP '17: European Conference on Object-Oriented Programming</i>.</p>
FEATURED ARTICLES	<p>Java is Unsound: The Industry Perspective, 2017. Featured by HackerNoon and dev.to.</p>
GRADUATED STUDENTS	<p>Fabian Muehlboeck, Ph.D. 2019: Efficient Runtimes for Gradual Typing</p> <p>Andrew K. Hirsch, Ph.D. 2019: Semantics for Secure Software</p> <p>Eric Chahin, M.Eng. 2015</p> <p>Jisha Kambo, M.Eng. 2015</p> <p>Ben Greenman, M.Eng. 2014</p> <p>Brian Toth, M.Eng. 2013</p>

CURRENT STUDENTS Brittany Nkounkou, Ph.D. Expected 2019: Trusting Computers to Help Us Design More Computers

GRANTS SHF: Small: Collaborative Research: A Rational Reconstruction of the Julia Type System, 2019 with Jan Vitek: \$252,742 and \$247,222
Mozilla Research Grant, 2017: \$44,270
CAREER: Gradual Typing for Industry Programming, 2014: \$651,829

GIFTS JetBrains, 2013: \$30,000
Microsoft Research, 2012: \$15,000

ACADEMIC SERVICE **Co-Chair:** PLDI SRC 2020, PLDI SRC 2019
Co-Organizer: PLMW 2016, PLMW 2015
Steering Committee Member: PLMW 2016–2018
Program Committee Member: OOPSLA 2019, PLDI 2018, Scala 2018, POPL 2014
External Review Committee Member: PLDI 2020, ASPLOS 2019

COURSES **CS/ENGRD-2110** Object-Oriented Programming and Data Structures: Spring 2016
CS-4120/4121/5120/5121: Introduction to Compilers: Fall 2013
CS-5152 Open-Source Software Engineering: Spring 2013, Spring 2014, Spring 2015, Spring 2017, Spring 2019
CS-6117 Category Theory for Computer Scientists: Fall 2014, Spring 2018
CS-6118 Types and Semantics: Fall 2012