James Rasmussen Wilcox

Curriculum Vitae

Paul G. Allen School of Computer Science & Engineering
University of Washington, Box 352350
Seattle, WA 98195

jrw12@cs.washington.edu
jamesrwilcox.com
Office: AC210

EDUCATION

Ph.D.	Computer Science & Engineering	University of Washington	Aug. 2021
M.S.	Computer Science & Engineering	University of Washington	Dec. 2015
B.A.	Computer Science and Mathematics	Williams College	June 2013

PRIMARY EMPLOYMENT

Paul G. Allen School of Computer Science & Engineering, University of Washington.

- Full-time Temporary Lecturer. September 2021–present
- Part-time Temporary Lecturer. March 2020–June 2021

TEACHING INTERESTS

I am interested in teaching programming, languages, and systems courses across all levels and backgrounds, and to both CS majors and non-majors. My primary teaching experience is in the "300- to 500-level" range, i.e., from undergraduate students who have just finished CS2 up through advanced undergraduate, master's, and Ph.D. students, but I am also open to teaching CS1 and CS2 with appropriate support.

CLASSES TAUGHT AT UW (7 offerings total)

CSE 331: Software Design and Implementation.

- Instructor of record: Fall 2021. 110 students. Managed 9 TAs.
- This required class comes after CS2 and teaches students how to write code of higher quality and increased complexity. We study Hoare logic, abstract data types, design patterns, testing, and advanced Java concepts.

CSE 341: Undergraduate Programming Languages.

- Instructor of record: Fall 2021, Fall 2020, and Winter 2017. 70–110 students and 5–7 TAs per offering.
- Contributed new homework and slides to pool of curriculum materials shared by all course instructors.
- Ported the course from SML to OCaml and from Ruby to object-oriented Racket.

CSE 374: Intermediate Programming Concepts and Tools.

- Instructor of record: Winter 2021. 80 students and 7 TAs.
- This is an introductory systems programming class *for non-majors*, covering C programming, the shell, and basic software engineering techniques such as version control and testing.
- Piloted contract-based assignments and grading.

CSE 490P: Advanced Programming Languages and Verification.

- Instructor of record: Spring 2020. 10 students and 1 TA.
- Pitched and designed this new advanced undergraduate seminar on the theory and practice of programming languages and verification. Students learned to do metatheory and implement type checkers and interpreters.
- Developed new course materials including ∼25 lectures and 12 homeworks (6 written and 6 programming).

CSE 505, CSEP 505: Graduate Programming Languages.

- Instructor of record: Spring 2021. Co-taught with Prof. Zachary Tatlock. 50 students and 4 TAs.
- Two different student populations: Ph.D. students and professional master's students.
- Piloted additive and transparent grading system.
- Developed new web IDE system for programming in System F.

ADDITIONAL TEACHING EXPERIENCE

- Teaching Assistant, CSEP 505, Spring 2019. Instructor: Prof. Zachary Tatlock.
- Outreach: I worked with a local high schooler weekly on learning to program in JavaScript. 2014–2017.
- Teaching Assistant, CSE 505, Autumn 2013. Instructor: Prof. Zachary Tatlock.
- Undergraduate Teaching Assistant, Williams College, Computer Science and Math departments, 2010-2013.

OTHER EMPLOYMENT

- Advisor (part-time), Certora, January 2021–present
- CTO, Certora, June 2019–December 2020
- Research Assistant, Prof. Zachary Tatlock, University of Washington, Seattle, WA, 2013–2019
- Research Intern, Dr. Jay Lorch, Microsoft Research, Redmond, WA, Summer 2017
- Research Assistant, Prof. Stephen Freund, Williams College, Williamstown, MA, 2012–2013
- Research Intern, Prof. Scott Shenker, ICSI, Berkeley, CA, Summer 2011
- Systems Administrator, Mary Bailey, Williams College, Williamstown, MA, Summer 2010

AWARDS AND FELLOWSHIPS

- Distinguished paper award, PLDI 2020
- Distinguished paper award, PLDI 2015
- National Science Foundation Graduate Research Fellowship, 2013–2018

ADVISING AND MENTORING

Undergraduate Research Supervised

- Steve Anton. Formal verification of the Raft consensus protocol. 2015-2016.
- Ryan Doenges. Verifying distributed systems with dynamic participants. 2015-2017.
  Now a Ph.D. student at Cornell.
- Miranda Edwards. Compositional verification of distributed systems. 2016-2017.
- Justin Adsuara. Verified serialization. 2017-2018.
- David Thien. Compiler testing. 2018.
- Ethan Shea. Automated verification of distributed systems. 2018.
- Taylor Blau. PL techniques for 3D printing and compositional verification of distributed systems. 2017-2018.

SIGPLAN Mentoring Program

- Karuna Grewal. 2020–present. I helped mentor Karuna through the Ph.D. application process. She is now a Ph.D. student at Cornell.
- Dani Wang. 2020–present. I helped mentor Dani through the Ph.D. application process. They are now a Ph.D. student at UT Austin.
- George Pîrlea. 2020–present. George is a second-year Ph.D. at NUS.
- Aaron Weiss. 2021–present. Aaron is a fifth-year Ph.D. student at Northeastern.
- Siddharth Bhat. 2021–present. Siddharth is a first-year Ph.D. student at the University of Edinburgh.

SOFTWARE MAINTAINED

`mypyvy`, Primary Developer and Maintainer, 2018–present

- An intermediate language and toolkit for manipulating symbolic transition systems.
- Transition systems are written in a decidable fragment of first-order logic.
- Supports verifying and synthesizing inductive invariants and several forms of bounded model checking.
- Has provided the basis for implementing tools described by several of my recent papers (CAV19, POPL22), as well as several papers by other researchers of which I am not an author.
- `mypyvy` itself is implemented in statically typed (!) Python and uses Z3 and CVC5 as underlying solvers.
- Available on GitHub: `https://github.com/wilcoxjay/mypyvy`

SERVICE

- POPL 2022 PC member
- OSDI 2021 ERC member
- ASPLOS 2021 ERC member
- VMCAI 2021 PC member
- OOPSLA 2020 external reviewer
- PLDI 2019 external reviewer
- POPL 2019 external reviewer
- OOPSLA 2018 external reviewer
- POPL 2017 external reviewer

PUBLICATIONS

Invited Articles

- Highlights in Systems Verification.
  James R. Wilcox.
  Communications of the ACM (**CACM**), February 2018.

Conference Publications

- Property-Directed Reachability as Abstract Interpretation in the Monotone Theory.
  Yotam M. Y. Feldman, Sharon Shoham, Mooly Sagiv, and **James R. Wilcox.**
  Principles of Programming Languages (**POPL**) 2022 (to appear).
- Induction Duality: Primal-Dual Search for Invariants.
  Oded Padon, **James R. Wilcox**, Jason R. Koenig, Kenneth L. McMillan, and Alex Aiken.
  Principles of Programming Languages (**POPL**) 2022 (to appear).
- Learning the Boundary of Inductive Invariants.
  Yotam M. Y. Feldman, Sharon Shoham, Mooly Sagiv, and **James R. Wilcox**.
  Principles of Programming Languages (**POPL**) 2021.
- Armada: Low-Effort Verification of High-Performance Concurrent Programs.
  Jacob R. Lorch, Yixuan Chen, Manos Kapritsos, Bryan Parno, Shaz Qadeer, Upamanyu Sharma, **James R. Wilcox**, and Xueyuan Zhao.
  Programming Languages Design and Implementation (**PLDI**) 2020.
  Distinguished Paper.
- Synthesizing Structured CAD Models with Equality Saturation and Inverse Transformations.
  Chandrakana Nandi, Max Willsey, Adam Anderson, **James R. Wilcox**, Eva Darulova, Dan Grossman, and Zachary Tatlock.
  Programming Languages Design and Implementation (**PLDI**) 2020.
- Inferring Inductive Invariants from Phase Structures.
  Yotam M. Y. Feldman, **James R. Wilcox**, Sharon Shoham, Mooly Sagiv.
  International Conference on Computer-Aided Verification (**CAV**) 2019.

- Functional Programming for Compiling and Decompiling Computer-Aided Design.
  Chandrakana Nandi, **James R. Wilcox**, Pavel Panchekha, Taylor Blau, Dan Grossman, and Zachary Tatlock.
  International Conference on Functional Programming (**ICFP**) 2018.
- Modularity for Decidability of Deductive Verification with Applications to Distributed Systems.
  Marcelo Taube, Giuliano Losa, Kenneth McMillan, Oded Padon, Mooly Sagiv, Sharon Shoham, **James R. Wilcox**, and Doug Woos.
  Programming Languages Design and Implementation (**PLDI**) 2018.
- VerifiedFT: A Verified, High-Performance Dynamic Race Detector.
  James R. Wilcox, Cormac Flanagan, and Stephen N. Freund.
  Principles and Practice of Parallel Programming (**PPoPP**) 2018.
- Programming and Proving with Distributed Protocols.
  Ilya Sergey, **James R. Wilcox**, and Zachary Tatlock.
  Principles of Programming Languages (**POPL**) 2018.
- Œuf: Minimizing the Coq Extraction TCB.
  Eric Mullen, Stuart Pernsteiner, **James R. Wilcox**, Zachary Tatlock, and Dan Grossman.
  Certified Programs and Proofs (**CPP**) 2018.
- Programming Language Abstractions for Modularly Verified Distributed Systems.
  James R. Wilcox, Ilya Sergey, and Zachary Tatlock.
  Summit on Advances in Programming Languages (**SNAPL**) 2017.
- Planning for Change in a Formal Verification of the Raft Consensus Protocol.
  Doug Woos, **James R. Wilcox**, Steve Anton, Zachary Tatlock, Michael D. Ernst, and Thomas Anderson.
  Certified Programs and Proofs (**CPP**) 2016.
- Array Shadow State Compression for Precise Dynamic Race Detection.
  James R. Wilcox, Parker Finch, Cormac Flanagan, and Stephen N. Freund.
  Automated Software Engineering (**ASE**) 2015.
- Verdi: A Framework for Formally Verifying Distributed System Implementations.
  James R. Wilcox, Doug Woos, Pavel Panchekha, Zachary Tatlock, Xi Wang, Michael D. Ernst, and Thomas Anderson.
  Programming Languages Design and Implementation (**PLDI**) 2015.
- Automatically Improving Accuracy for Floating Point Expressions.
  Pavel Panchekha, Alex Sanchez-Stern, **James R. Wilcox**, and Zachary Tatlock.
  Programming Languages Design and Implementation (**PLDI**) 2015.
  Distinguished Paper.

Workshop Publications

- Verification of Implementations of Distributed Systems Under Churn.
  Ryan Doenges, **James R. Wilcox**, Doug Woos, Zachary Tatlock, and Karl Palmskog.
  Workshop on Coq for Programming Languages (**CoqPL**) 2017.
- Information-centric networking: Seeing the forest for the trees.
  Ali Ghodsi, Scott Shenker, Teemu Koponen, Ankit Singla, Barath Raghavan, and **James Wilcox**.
  Workshop on Hot Topics in Networks (**HotNets**) 2011.
- Intelligent design enables architectural evolution.
  Ali Ghodsi, Scott Shenker, Teemu Koponen, Ankit Singla, Barath Raghavan, and **James Wilcox**.
  Workshop on Hot Topics in Networks (**HotNets**) 2011.

Journal Publications and Theses

- Compositional and Automated Verification of Distributed Systems.
  James R. Wilcox.
  Ph.D. Thesis, 2021.
- Sets Characterized by Missing Sums and Differences in Dilating Polytopes.
  Thao Do, Archit Kulkarni, Steven J. Miller, David Moon, Jake Wellens, and **James Wilcox**.
  Journal of Number Theory, 2015.

- ShrinkWrap: Efficient Dynamic Race Detection for Array-Intensive Programs.
  James Wilcox. Stephen N. Freund, Advisor.
  Williams College Undergraduate Honors Thesis, 2013.

TALKS

- Intro to Dafny.
  Guest Lecture. University of Utah. Salt Lake City, Utah. September 2021.
- Transition Systems, Verification, and Dafny.
  Guest Lecture. University of Utah. Salt Lake City, Utah. March 2020.
- Verifying Distributed Systems with `mypyvy`.
  Seminar Talk. University of Utah. Salt Lake City, Utah. March 2020.
- Compositional Verification of Distributed Systems.
  Thesis Defense. University of Washington, Seattle, WA. May 2019.
- Goldilocks the Verification Engineer.
  Galois. Portland, OR. March 2019.
- Verifying Distributed Systems.
  New England Systems Verification Day. Cambridge, MA. October 2018.
- Compositional Verification of Distributed Systems.
  - Invited talk. Tel Aviv University, Israel. March 2018.
  - PLSE retreat. Leavenworth, WA. September 2016.
- VerifiedFT: A Verified, High-Performance Dynamic Race Detector.
  Conference talk at PPoPP. Vienna, Austria. March 2018.
- Programming and Proving with Distributed Protocols.
  Conference talk at POPL. Los Angeles, CA. January 2018.
- Programming Language Abstractions for Modularly Verified Distributed Systems.
  Conference talk at SNAPL. Monterey, CA. May 2017.
- Verdi: A Framework for Formally Verifying Distributed System Implementations.
  - Invited talk. University College London. United Kingdom. June 2016.
  - Invited talk. Google. Seattle, WA. June 2016.
- Planning for Change in a Formal Verification of the Raft Consensus Protocol.
  Conference talk at CPP 2016. St. Petersburg, FL. January 2016.
- Array Shadow State Compression for Precise Dynamic Race Detection.
  Conference talk at ASE. Lincoln, NE. November 2015.