John Toman

Research Interests

I am interested in using program analysis and rigorous programming language techniques to solve practical software engineering problems. In particular, my research at Certora Inc. has focused on bringing the power of sound static analysis of low-level bytecode for the purposes of program optimization and verification.

Education

July, 2016 – March, 2019 **PhD, Computer Science**, *University of Washington*, Seattle,
Dissertation: Learning to Adapt: Program Analyses for Configurable Software.
Advisor: Dan Grossman

Sept., 2014 – June, 2016 **Master of Science, Computer Science**, *University of Washington*, Seattle.
GPA: 3.96

Sept., 2008 – Dec., 2012 **Bachelor of Science, Computer Science**, *University of Maryland*, College Park, *Cum Laude*.
GPA: 3.95

Publications

John Toman, Ren Siqi, Kohei Suenaga, Atsushi Igarashi, and Naoki Kobayashi. Consort: Context-and flow-sensitive ownership refinement types for imperative programs. In *ESOP*, pages 684–714, 2020.

John Toman and Dan Grossman. Concerto: A Framework for Combined Concrete and Abstract Interpretation. *Proceedings of the ACM on Programming Languages*, Principles of Programming Languages (POPL), January 2019. `https://homes.cs.washington.edu/~jtoman/assets/concerto-2018.pdf`.

John Toman and Dan Grossman. Legato: An At-Most-Once Analysis with Applications to Dynamic Configuration Updates. In *32nd European Conference on Object Oriented Programming (ECOOP)*, 2018. `https://homes.cs.washington.edu/~jtoman/assets/ecoop-2018.pdf`.

John Toman and Dan Grossman. Taming the Static Analysis Beast. In *2nd Summit on Advances in Programming Languages (SNAPL)*, 2017. `https://homes.cs.washington.edu/~jtoman/assets/snapl-2017.pdf`.

John Toman and Dan Grossman. Staccato: A Bug Finder for Dynamic Configuration

Updates. In *30th European Conference on Object Oriented Programming (ECOOP)*, 2016. https://homes.cs.washington.edu/~jtoman/assets/staccato-2016.pdf.

John Toman, Stuart Pernsteiner, and Emina Torlak. Crust: A Bounded Verifier for Rust. In *30th Conference on Automated Software Engineering (ASE)*, 2015. https://homes.cs.washington.edu/~jtoman/assets/crust.ase15.pdf.

Brianna M. Ren, John Toman, T. Stephen Strickland, and Jeffrey S. Foster. The Ruby Type Checker. In *Object-Oriented Program Languages and Systems (OOPS) at Symposium on Applied Computing*, 2013. https://homes.cs.washington.edu/~jtoman/assets/rtc-2013.pdf.

Undergraduate Honors Thesis

Title *Topics in Compilers*

Advisor Dr. Jeffrey Foster

Description Developed methodologies for effectively teaching advanced programming language concepts and techniques including type-inference, data flow analysis, and parsing.

Awards

July, 2016 **Staccato: A Bug-Finder for Dynamic Configuration Updates**, *Distinguished Artifact*, Artifact Track, ECOOP.

July, 2016 **Staccato: A Bug-Finder for Dynamic Configuration Updates**, *Distinguished Poster*, Poster Track, ECOOP.

Service

2020 ASPLOS External Review Committee Member

2018 PLDI AEC Program Committee

March, 2017 Department Orientation Panel

2017 ECOOP 17 Doctoral Symposium Program Committee

2017 Graduate Application Reader Committee

2016 Graduate Application Reader Committee

March, 2016 PLSE visit day activity coordinator

March, 2015 PLSE visit day activity coordinator

Work Experience

Sept. 2020 – Present **VP Research & Development**, *Certora, Inc.*, Seattle.
- Lead a team of 9+ developers in improving the performance and precision of Certora's core product
- Manage task prioritization and assignments
- Lead complete overhaul of a legacy subsystem of Certora's core product

Feb., 2020 – Sept. 2020 **Formal Verification Expert**, *Certora, Inc.*, Seattle.
- Developed multiple static analyses that enabled previously impossible simplifications
- Developed a framework for managing EVM bytecode contract instances
- Implemented a novel pattern matching framework for quickly prototyping peephole optimizations

May, 2019 – Jan. 2020 **Postdoctoral Researcher**, *Kyoto University*, Foundations of Software Lab.
- Developed a prototype automatic program verifier based on a novel refinement type system
- Formalized and proved the soundness of the type system
- Extended the prototype to support several complex program patterns, including recursive types, null types, and arrays
- Wrote a paper on this work that was accepted for publication at a top conference in the field

April 2017 – June 2017 **Software Development Intern**, *Facebook*, Seattle.
- Developed an automatic code repair framework for defects reported by a proprietary static analysis
- Contributed features and repaired bugs in a proprietary static analysis
- Extended a proprietary analysis to report a new type of code defect

July 2015 – Sept. 2015 **Software Development Intern**, *Twitter*, New York.
- Implemented ingestion pipeline for data from major 3rd party provider
- Designed an implemented a new location API for use by internal services

Feb. 2013 – July 2014 **Web Applications Program Manager**, *CATT Laboratory*, College Park.
- Oversaw a team of approximately 5 full-time faculty researchers and 6 students
- Directed development of two real-time national traffic monitoring websites used by the public and traffic operators
- Developed policies to increase code quality including unit testing and code review
- Designed, architected, and implemented a cross-platform, in-browser virtual meeting application without dependencies on proprietary browser plugins and apps
- Designed a storage scheme with HBase for warehoused data that resulted in a 4x reduction in disk usage
- Migrated an single-threaded, iterative computation to a MapReduce computation, decreasing runtimes from hours to seconds
- Improved performance of image generation framework from 500 simultaneous requests to +10,000 simultaneous requests
- Created a weighted spatial clustering algorithm that efficiently handles in-place updates to weights of objects
- Implemented a custom system for automatically managing web assets and their dependencies
- Planned and oversaw implementation of a real-time in-browser instant messaging system

June 2009 – Jan. 2013 **Web Developer**, *CATT Laboratory*, College Park.
- Maintained and added features to a real-time traffic monitoring system used by transportation agencies nationwide
- Led the development of a public version of a real-time traffic monitoring website
- Led the refactoring of legacy JavaScript code that reduced code base size by 40%
- Developed new APIs that simplified development, reduced duplicate code and lowered complexity

Jan. 2012 – Dec. 2012 **Research Assistant**, *PLUM Research Group*, College Park.
- Helped develop an opt-in type checker for the dynamic language Ruby.
- Implemented several key optimizations to improve the type-checker performance
- Created a "lazy mode" that slightly delayed detection of type errors in exchange for improved performance

Teaching and Mentoring Experience

May, 2019 – Jan., 2020 **Postdoctoral Fellow, Kyoto University**.
Co-advised a masters student on her research project and assisted in the formal development of her graduation thesis.

Jan., 2019 – March 2019 **Teaching Assistant, UW CSE341: Programming Languages**.
Teaching assistant in undergraduate programming languages course: developed and presented section material, wrote exam questions, graded homework, answered questions, held office hours.

Sept., 2016 – Dec., 2016 **Teaching Assistant, UW CSE505P: Programming Languages**.
Sole teaching assistant for professionals master course of 26 students: graded homework, answered questions, developed homeworks, met with students needing help.

Sept., 2015 – June, 2016 **New Grad Mentor**.
Advised first year graduate students on navigating advisor relationships, managing research projects, balancing coursework, etc.

April, 2015 – June, 2015 **Undergraduate CS Mentor**.
Met with two undergraduate students requiring extra help in discrete mathematics: worked through homework problems and explained concepts like induction and propositional logic proofs.