

Xuanrui (Ray) Qi

✉ me@xuanruiqi.com
<http://www.xuanruiqi.com>

Education

- May 2019 **M.S. in Computer Science**, *Tufts University*, Medford, MA, USA.
Focus: functional programming & programming languages
- May 2018 **B.S. in Computer Science**, *magna cum laude*, with honors in thesis, *Tufts University*, Medford, MA, USA.
Second major: international relations. Minor: mathematics.
- June 2014 **High School Diploma**, *Shenzhen Middle School*, Shenzhen, Guangdong, China.

Experience

- June 2019 – **Intern**, *SiFive, Inc.*, San Mateo, CA, USA.
- August 2019 Working on verification and theorem proving.
- Sep 2018 – **Graduate Researcher**, *Department of Computer Science, Tufts University*, Medford,
May 2019 MA, USA.
Doing research for my master's degree.
- Jun – Aug **Research Visitor**, *Graduate School of Mathematics, Nagoya University*, Nagoya,
2018 Japan.
Hosted by and worked with Professor Jacques Garrigue.
- Mar – May **Research Assistant**, *Department of Computer Science, Tufts University*, Medford,
2018 MA, USA.
Research assistant under Professor Sam Guyer, working in the RedLine Systems Research Group.
- Jul – Aug **Intern**, *Institute of Automation, Chinese Academy of Sciences*, Beijing, China.
2016 Interned at the State Key Laboratory of Control and Management of Complex Systems, working on computer vision.
- Sep – Dec **Teaching Assistant**, *Department of Computer Science, Tufts University*, Medford,
2017 & Sep MA, USA.
2018 – Dec Course: Concurrent Programming (Fall 2017 & Fall 2018).
2018

Technical Skills

Programming Languages

Haskell, OCaml, Erlang, Python, Ruby, Java, C/C++

Programming & Software Engineering

Performance engineering (especially GC/runtime related), dynamic program analysis, concurrent programming, low-level/systems programming

Formal Methods

Interactive theorem proving (Coq, Agda), type systems, static analysis, program logics, SMT-assisted reasoning

Security

Network analysis, packet analysis, penetration testing, web security (SQL injection, cross-site scripting, etc.), systems security

Other

Git, Linux system administration

Projects

Live Programming with Typed Holes and Polymorphism

Working on Hazel, a live programming language with structural editing, with a focus on adding sound, implicit polymorphism.

Formal verification of dynamic compact data structures

I did part of the design, implementation and modeling in Coq. A paper about our formalization has been accepted for publication.

<https://github.com/affeldt-aist/succinct>

Elephant Tracks II: high-performance, extensible GC tracing framework

I did most of the design and implementation of a prototype in C++. The resulting system's performance increased more than 10× compared to our previous systems.

<https://github.com/ElephantTracksProject/et2-java>

JumboViz: visualizing GC traces

I did most of the JVM-related hacking in C++.

<https://github.com/HeapVisCapstone>

Dynamic, Distributed File Backup System

I collaborated with two colleagues in design and implementation in Erlang.

<https://github.com/DistBackup/dbscore>

Research Publications

Research Papers

1. Reynald Affeldt, Jacques Garrigue, **Xuanrui Qi**, and Kazunari Tanaka. Proving Tree Algorithms for Succinct Data Structures. Accepted to the 10th Conference on Interactive Theorem Proving (ITP 2019).
2. **Xuanrui (Ray) Qi**. Elephant Tracks II: Practical, Extensible Memory Tracing. Senior Honors Thesis, Tufts University, 2018.

Talks & Presentations

1. Reynald Affeldt, Jacques Garrigue, **Xuanrui Qi**, and Kazunari Tanaka. Experience Report: Type-Driven Development of Certified Tree Algorithms in Coq. Submitted to the Coq Workshop 2019.
2. **Xuanrui (Ray) Qi**. From Tactics to Structure Editors for Proofs. Off the Beaten Track 2019 (OBT 2019).
3. **Xuanrui (Ray) Qi**. A Practical and Extensible Framework for Garbage Collection Tracing. SPLASH 2018 Student Research Competition.

Relevant Coursework

PL “Programming Languages” (incl. functional programming), “Concurrent Programming” (in Erlang), “Program Analysis, Verification & Synthesis”, “Foundations and Pragmatics of Dependently-Typed Systems for Interactive Proof-Assistance and Certifiably-Safe Programming”

Security “Computer Systems Security”, “Cryptography”

Theory & Algorithms “Algorithms”, “Advanced Algorithms”, “Theory of Computation”

Others “Working with Corpora”