

Yixuan Chen

curriculum vitae

+1 (734) 789 0357
✉ yixuan.chen@yale.edu
📄 me.xlk.me
🌐 LukeXuan

Education

- 2019– **Ph.D.**, *Computer Science*, Yale University.
2017–2019 **Bachelor**, *Computer Science Engineering*, University of Michigan, 3.8.
2015–2019 **Bachelor**, *Electrical and Computer Engineering*, Shanghai Jiaotong University, 3.6.

Research Interests

Formal verification, programming languages, operating systems and applying formal verification in large and concurrent software systems

Research

- May. – Aug. **Research intern**, *prof. Andrew Appel*, Princeton University.
2018 Designing and verifying high performance KV-database implemented in C
 - Using Verified Software Toolchain and Verifiable C to prove the functional correctness of C programs
 - Designing and verifying a high performance in-memory KV-database based on MassTree and SQLite cursor
 - Exercising the modular verification approach and finding the correct abstractions
 - Expecting results and methodologies to be present at POPL student research competition and other publications in 2019

2017–2019 **Research assistant**, *prof. Manos Kapritsos*, University of Michigan.
Verification tool on concurrent programs
 - Understanding and modifying the Dafny language and program verifier
 - Designing algorithms for automatic derivation of abstract syntax tree mappings to generate refinement proofs between transformation of programs
 - Expecting results to be present at OSDI in 2019

Academic Experience

- 2020 Winter **Teaching assistant**, *CPSC 421 Compilers and Interpreters*, Yale University.
 - duties including holding office hours and grading

2019 Winter **Teaching assistant**, *EECS 482 Operating Systems*, University of Michigan.
 - 250-student upper level technical elective course
 - duties including holding office hours and lab teaching

2018 Fall **Teaching assistant**, *EECS 482 Operating Systems*, University of Michigan.
 - 250-student upper level technical elective course
 - duties including holding office hours and lab teaching

- 2018 Jul. **Student volunteer**, *DSSS 2018*, Princeton, NJ.
DeepSpec Summer School 2018
◦ Assisting participants on Coq proofs with Verified Software Toolchain
- 2018 Jan. **Participant**, *POPL 2018*, Los Angeles, CA.
Principles of Programming Languages 2018

Publications

- 2020 *Low-Effort Verification of High-Performance Concurrent Programs*, by Jacob R. Lorch, Yixuan Chen, Manos Kapritsos, Bryan Parno, Shaz Qadeer, Upamanyu Sharma, James R. Wilcox, Xueyuan Zhao, 41st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '20), London, United Kindom
- 2019 *Verification of a Cache-optimized Data Structure*, by Yixuan Chen, Aurèle Barrière, Lennart Beringer, and Andrew W. Appel, accepted to appear in poster session at POPL 2019 Student Research Competition (POPL '19 SRC), Lisbon, Portugal

Industrial Experience

- 2016–2017 **Intern**, *Apple Inc.*, Shanghai.
◦ Developed concurrent software systems used for audit and version control of test stations
◦ Wrote detailed code documents and deployment instructions for the systems to be maintained after leaving

Projects

- 2018 Winter **Course Project**, *Fault-tolerant distributed chat server*, EECS591 (Graduate Distributed Systems).
Implemented chat server with replication that tolerates benign failures and unreliable channels
◦ Understood **Paxos** and **multi-Paxos** algorithms and implemented in Python
◦ Dealt with non-deterministic bugs introduced by unreliable channels
- 2018 Winter **Course research project**, *On investigating mixer's impact on database performance*, EECS 591 (Graduate Distributed Systems).
Identifying application-aware mixer's impact on the performance of database
◦ Benchmarked the database performance with TPC-W datasets
◦ Reasoned about the database performance in terms of network, concurrency, database and application logic
- 2018 Winter **Course project**, *Compiler for COOL*, EECS 483 (Undergraduate Compilers).
Implemented a compiler for COOL (subset of Java) programming language, written in OCaml and compiles COOL program into x86_64 assembly program
◦ Implemented a full-stack from lexical analyzing to code generation
◦ Used intermediate language to achieve extensibility
◦ Generated assembly file that can be assembled into ELF file and directly executable on Linux

Programming Languages

Functional OCaml, Coq

Imperative C, C++, C#, Pascal

Scripting Python, Javascript, Shell (and variants)

Others Dafny, LaTeX, HTML/CSS, Verilog, Matlab, Mathematica