Website: https://effect.systems **GitHub:** @yihozhang
E-mail: yz489@cs.washington.edu **Twitter:** @yihongz_bot

Research Interests

E-graphs and Equality Saturation. Staging and Program Manipulation. Program Synthesis. Datalog.

Education Background

Master, University of Washington Sept 2021-June 2022

- M.S. in Computer Science

Bachelor, University of Washington Sept 2018-June 2021

- B.S. with Departmental Honors in Computer Science, *Magna Cum Laude*. GPA 3.90/4.00.
- Thesis: *Faster and Worst-Case Optimal E-Matching*, advised by Zachary Tatlock.
- Selected Coursework:
  - Graduate: Principles of Programming Languages, Principles of Database Systems, Computer-Aided Reasoning for Software, Software Engineering, Database Internals, Artificial Intelligence.
  - Undergrad: Advanced PL & Verification, Design and Implementation of DSLs, Distributed Systems, Operating Systems.

Publications

- **Yihong Zhang**, Remy Wang, Max Willsey, Zachary Tatlock. *Relational E-matching*. Principles of Programming Languages (POPL) 2022.
- **Yihong Zhang**. *Faster and Worst-Case Optimal E-matching via Reduction to Conjunctive Queries*. Programming Language Design and Implementation (PLDI) 2021 Student Research Competition. **Awarded first place at the undergraduate division.**
- Maximilian Schleich, Zixuan Geng, **Yihong Zhang**, and Dan Suciu. *GeCo: Quality Counterfactual Explanations in Real Time*. PVLDB, 14(9): 1681-1693, 2021. doi:10.14778/3461535.3461555.

Research Experience

Relational E-Matching, advised by Zachary Tatlock Jan 2021–Current

- Performed simulations and benchmarking at the early stage of research and developed an efficient implementation for relational e-matching.
- Narrate relational e-matching in my undergrad thesis and collaborate with coauthors on our POPL paper.
- Presented the work at PLDI Student Research Competition and won first place.

Practical Deforestation for Haskell Programs, advised by Lionel Parreaux June 2021–Sept 2021

- Devised a practical deforestation algorithm for functional programs with Prof. Parreaux. Unlike prior work on short-cut deforestation or supercompilation, our algorithm reasons the structure of producer and consumer functions with surgery precision and avoids unproductive fusions.
- Implemented the prototype as a Haskell plugin.
- Presented the algorithm and preliminary result at the end of internship.

A Genetic Algorithm for Counterfactual Explanations, advised by Dan Suciu Aug 2020–Mar 2021

- Researched and implemented the efficient compilation and evaluation of partial models, including random forests and multi-layer perceptions.

- Proposed two optimizations that are critical in making the genetic algorithm real-time.

Employment

Teaching Assistant, University of Washington Sept 2020-June 2021

- Responsibility includes grading assignments, hosting office hours, teaching sections, and designing homework.

Software Engineer Intern, Oracle June 2020-Sept 2021

- Worked with the Order-to-Cash team on several features and issues for NetSuite, an enterprise-resource planning platform.

Honors

First Place, PLDI Student Research Competition (Undergraduate Division) June 2021
First Place, International Collegiate Programming Contest (ICPC), UW Qualifier Oct 2019
Fifth Place, International Collegiate Programming Contest (ICPC), Pacific Northwest Region Nov. 2018
Dean's List, University of Washington Dec 2018–Current

Grad Course Projects

Sdl: A Staged Datalog Compiler using Lightweight Modular Staging (LMS)

- I build a staged Datalog compiler using Lightweight Modular Staging (LMS). Early experiments show that it achieves up to 10× speedup compared to Souffle Datalog compiler.

MaxDuet: Reconciling Statistical, Top-down, and Bottom-up Search for Programming by Example

- As an attempt to unify enumerative, deductive, and stochastic search into a framework for program synthesis, I add support for enumerative search to MaxFlash, a recent PBE project. The enumerated programs are used to directly solve the synthesis problem and guide the witness functions during probabilistic deductions.

Cornelius: Killing Equivalent and Redundant Mutants with E-graphs

- With Ben Kushigian, Ishan Chatterjee, and Gabrielle Strandquist. We propose to utilize e-graphs to eliminate equivalent and redundant mutants for mutation testing. Experiments on a pure Java subset show that it discovers much more equivalent and redundant mutants than Trivial Compiler Equivalence in less time.

Sager: A Demonic Graph Synthesizer for Worst-Case Performance

- With Mike He. We propose to use symbolic evaluations to synthesize input instances that manifest the worst-case complexity of a given algorithm. We build a prototype implementation with Rosette and show it makes Shortest-Path Faster Algorithm (SPFA) and its variants asymptotically worse.

Teaching

Teaching Assistant, CSE 341: Programming Languages Autumn 2020
Teaching Assistant, CSE 374: Intermediate Programming Concepts and Tools Winter 2021
Teaching Assistant, CSE 505: Principles of Programming Languages (Graduate) Spring 2021

Programming for Fun

- Hatafun: A prototype embedding of the type system of Datafun (ICFP 2016) in Haskell.
- egraph-sql: A minimal implementation of e-graphs as relational databases.