

Matias Scharager

matias.scharager@gmail.com | pi314mm.github.io

WORK EXPERIENCE

Software Engineer *AWS, Seattle* Summer 2023

- Worked on optimizing LLVM code produced by a database query

Software Engineer *Google, Sunnyvale* Summer 2022

- Statically analyzed JavaScript using multiple intermediate representations (JSIR) in the MLIR LLVM framework
- Modified compilation passes from a high level JSIR to a low level JSIR for an explicit control flow graph structure
- Created a Dead Code Analysis pass via encoding control flow mechanisms in the low level JSIR
- Created a Constant Folding pass via creating an interface between JSIR and the v8 engine for expression execution
- Working on an additional project: experimental common IR for malware analysis, initial results seem promising

Software Engineer *Meta, Menlo Park* Summer 2020

- Analyzed the dependency graph structure of C/C++ Buck builds along with the objects being passed into the linking process to determine potential code bloat in large binaries
- Implemented, documented, and successfully used a tool to help identify poorly utilized libraries
- Improved the compilation time and diminished the final size of important C/C++ binaries
- Experimented with an automated script to remove unused #include headers

Applied Research Mathematician and Software Engineer *National Security Agency* Summer 2019

- Granted a Top Secret/SI (Special Intelligence) security clearance with full scope polygraph
- Optimized algorithms in a custom assembly language for a high-performance SIMD computer
- Developed skills in python programming, assembly language programming, machine level architecture, parallel computing, and algorithm analysis

Cybersecurity Developer *Northrop Grumman Xetron* Summer 2018

- Designed and solved cybersecurity challenges including reverse engineering and buffer exploits with IDA Pro
- Synchronized a web interface and database with automated test execution on multiple virtual machines in parallel

Machine Shop Engineer *Max Planck Florida Institute for Neuroscience* Summer 2015

- Using SolidWorks, designed specialized equipment for microscopes lens tracks used in neuroscience research
- Programmed a five-axis milling machine and operated several machines for constructing aluminum lens holders

EDUCATION

PhD: Carnegie Mellon University (CMU) August 2021 - Expected: 2026

- Computer Science Department: Type Theory
- Advisor: Karl Crary

Bachelors: Carnegie Mellon University (CMU) 3.67 GPA August 2017 - May 2021

- Bachelor of Computer Science - School of Computer Science (SCS) College
- Minor in Logic and Computation and SCS Concentration in Programming Language Theory
- University Honors and SCS College Honors

PUBLICATIONS

Verified Quadratic Virtual Substitution for Real Arithmetic

Matias Scharager, Katherine Cordwell, Stefan Mitsch and André Platzer

- Formal Methods (FM) 2021. (doi | arXiv | AFP)
- Formally verified Virtual Substitution algorithm in the Isabelle theorem prover language
- Implemented efficient and verified simplification of quantified first order real arithmetic formulas

WORKS IN PROGRESS

Full Abstraction Correctness of Compilation using Type-Oriented Merging	2020-Present
<i>Matias Scharager, Karl Crary, Matthew McKay (Submission planned Summer 2024)</i>	
Compactness via Pattern Stepping Bisimulation	2023
<i>Matias Scharager (Submission planned Spring 2024)</i>	

TEACHING EXPERIENCE

Types and Programming Languages (15-814) <i>TA under Professor Jan Hoffmann</i>	Fall 2023
Constructive Logic (15-317) <i>TA under Professor Karl Crary</i>	Fall 2021
Programming Language Theory (15-312) <i>TA under Professor Robert Harper</i>	Fall 2020
Student Taught Course: Anime (98-038) <i>Co-Instructor</i>	Fall 2020 – Spring 2021
Algorithm Design and Analysis (15-451) <i>TA under Professors Daniel Sleator and Gary Miller</i>	Fall 2019

RELEVANT COURSEWORK

80-713 Category Theory	Fall 2022
15-819 Advanced Topics in Programming Languages	Spring 2022
15-780 Graduate AI	Spring 2022
15-857 Analytical Performance Modeling & Design of Computer Systems <i>Queueing theory</i>	Fall 2021
15-414 Bug Catching <i>Writing formally verified programs</i>	Spring 2021
07-599 Undergraduate Research Thesis	Fall 2020 - Spring 2021
15-411 Compilers <i>Implemented a compiler from a subset of C to assembly</i>	Fall 2020
15-417 HOT Compilation <i>Implemented a type-directed compiler from SML to C</i>	Spring 2020
15-819 Advanced Topics in PL: Computational Higher Type Theory	Spring 2020
15-317 Constructive Logic <i>Theorem proving in Prolog and SML</i>	Spring 2020
15-312 Programming Language Theory <i>Statics and dynamics of various languages</i>	Fall 2018
80-411 Proof Theory <i>Various topics in formal proofs and computability</i>	Fall 2019
15-451 Algorithm Design and Analysis <i>Various topics in computer science theory</i>	Spring 2019
80-419 Interactive Theorem Proving <i>Formal verifications in the Lean language</i>	Spring 2019