Litao Zhou

tonyzhou0608@gmail.com • ltzhou.com

Final year undergraduate applying for PhD in Computer Science, with research interest in Programming Languages, Program Verification and Functional Programming.

Education

Bachelor of Science with Honors — Shanghai, China
Shanghai Jiao Tong University — *Sept 2018 - June 2022 (Expected)*
GPA: 4.06/4.3; Rank:2/134 (first three year)

Non-graduating Non-exchange Student — Singapore
National University of Singapore, School of Computing — *Aug 2021 - Dec 2021 (Expected)*

Research Experience

Imperative Program Verification in Two Phases
Supervised by Qinxiang Cao, John Hopcroft Center for Computer Science, SJTU — *2019 Fall - 2021 Spring*

- Proposed a flexible two-phase approach to verify imperative programs, based on the idea of decomposing the verification into algorithm proof and implementation proof.
- Proved the recursion based imperative C implementation of Tarjan's Strongly Connected Components Algorithm correct in VST w.r.t. an iterative algorithm description in Coq.
- The proposed solution reduces the implementation proof to 800 LoCs and enables modular verification.

Foundationally Sound Annotation Verifier for Sequential C programs
Supervised by Qinxiang Cao, John Hopcroft Center for Computer Science, SJTU — *2020 Fall - 2021 Summer*

- Build and prove the soundness of a control-flow splitting tool based on VST to simplify verification proof of assertion-annotated C programs.
- Identify conjunction rule as a key requirement for the framework. To this end, we supplement VST program logic with a model-level theorem about the preciseness of load/store operation.

Higher-Order Abstract Syntax Approach to Verified Functional Program Compilation
Supervised by Yuting Wang, John Hopcroft Center for Computer Science, SJTU — *2020 Summer - now*

- Implemented several functional compilation phase, namely, higher-order uncurrying, lambda shrinking and lambda lifting in λProlog, a high-order logic programming language.
- With HOAS, the binding structure in the object functional language can be embedded into the meta-theory of the system, thereby greatly simplifying the overall implementation and reasoning processes.

Extend HIP/SLEEK Verifier to Support Higher-order Functions
Supervised by Chin Wei Ngan, School of Computing, NUS — *Ongoing*

Awards & Scholarships

- 2019-2020 **National Scholarship** (Ministry of Education of P.R. China) for top 2% students.
- 2018/2019/2020 **Zhiyuan Scholarship** (Zhiyuan College, SJTU) for top 5% students majoring in Engineering.
- Awarded the **Second Prize** of **Software Service Innovation Competition** at NCSC 2020 (National Conference on Services Computing) hosted by CCF (China Computer Federation)
- Awarded **First Prize** of **National Mathematics Competition** for College Students in 2019

Practical Experiences

SJTU-Go: A Navigation System for Shared Vehicles
Software Engineering Course Project at SJTU — *2020 Spring - 2020 Summer*

- Used JavaScript and Java to create WeChat mini-program and a backend system that allow users to check the real-time distribution of various shared vehicles and navigate through SJTU campus.
- The software has won the second prize of 2020 CCF NCSC Software Service Innovation Competition

Efficiency Testing and Debugging Tool
Software Engineer Intern, Intel Asia Pacific Research & Development Center, Shanghai — *2020 Fall - 2021 Winter*

- Used Django (Python) and Vue (JavaScript) to develop a web-based automation tool to smooth the testing workflow of graphics drivers.
- The tool has been put into internal use in the Visual and Parallel Computing Group

Technical Skills

- **Verification:** Coq, Abella
- **Functional:** OCaml
- **Logic:** Teyjus (λProlog)
- **Imperative:** C, C++, Java
- **Scripting:** Python, JavaScript, etc.
- **Others:** LaTeX, HTML/CSS, PHP, MySQL

Other Experiences

- OPLSS 2021 **Participant**
- ICFP 2021 **Student Volunteer**