roife

[redacted] | roifewu@gmail.com | roife | roife.github.io | Shanghai / Beijing / Hangzhou / Hong Kong

Education

Nanjing University 2023.09 - 2026.06 (expected)
Master's Degree in *Computer Science and Technology* | Pascal Lab. Tutor: [redacted] Li | Focus on PL, program analysis and HDL

Beihang University 2019.09 - 2023.06
Bachelor's Degree in *Computer Science and Technology* | GPA 3.84/4.00

Work Experience

Rust Foundation Fellowship Program 2024.09 - 2025.06 (expected)

- **Contributing to rust-analyzer** (the official Rust IDE)
  - Ranked **19/970** in contributors; resolved **over 50 issues** and contributing to multiple modules (e.g. **semantic analysis**, etc.), improving robustness of the project. Also introduced several new features (e.g. **control flow navigation**, etc.);
  - Implemented a **SIMD** version of the line-breaking algorithm, leading to a **6.5x** speedup for this module on ARM;
  - **Resolved a P0 incident in v0.3.1992**. 4 hours after the release, the community encountered a critical bug that drained resources and blocked processes. I identified the issue **in 3 hours** and designed a new algorithm as fix. This emergency fix mitigated the incident, preventing widespread disruptions for Rust users and improving project stability.
- **Open-source community maintenance**: Including meeting discussions, bug fixes, PR reviews, and more.

Awards

- 2022 **National Scholarship** (Ranked 1/195 in the major), **Outstanding Graduate** of Beihang University
- **1st Prize** in the NSCSCC Compilation System Design Competition (Huawei Bisheng Cup) 2021, ranking 2nd overall.
- Additionally awarded over ten provincial and university-level awards and scholarships

Projects

Vizsla, a modern Verilog/SV IDE for hardware development (Rust / SystemVerilog) (In development)

- (**Project Leader**) Designed the core architecture of the IDE, the incremental computation processes, intermediate representation, semantic analysis module, etc. Also implemented most of the IDE functionalities.
- Aimed to equip chip design with modern IDE features (e.g. **code navigaiton**, **completion**, etc.) to enhance productivity.
- Based on the LSP, built an **incremental semantic analysis framework**.

LLVM-Lite, a lightweight edge-side compiler for neural network operators (C++ / LLVM / ARM) roife/llvm-lite

- (**Independently Developed**) Huawei research project, received an excellent evaluation as my undergraduate thesis.
- Utilizing shape information of neural networks to perform optimizations on operators, reducing its runtime cost.
- Successfully **reduced runtime by 6%** and **target file size by 38%** of the neural network operators in test cases.

Ayame, project of the Huawei Bisheng Cup, a compiler of a C-subset (Java / LLVM / ARMv7) No-SF-Work/ayame

- (**Co-author**) Implemented the graph-coloring register-allocation, arch-specific optimizations, the local evaluator and CI;
- The project ranked **1st in nearly half of the testcases** and outperformed `gcc -O3` and `clang -O3` on 1/3 of the examples.

Open-source contributions

- **Rust-lang Member**, rust-analyzer contributors team. Mainly worked on rust-lang/rust-analyzer. Also contributed to rust-lang/rust-clippy, rust-lang/rustup, rust-lang/rust-mode
- llvm/llvm-project, clangd/vscode-clangd, google/autocxx, yuin/goldmark and more on my GitHub.

Skills

- *Programming Languages:* Not limited to specific language. Proficient in C, C++, Java, Rust, Python, Verilog/SystemVerilog. Comfortable with Ruby, Swift, JavaScript, OCaml, Coq, Haskell, etc.
- *PL Theory:* Familiar with **type systems**, formal semantics, formal verification and theory of computation.
- *Compilers / IDE:* **4 YoE**. Proficient in compilation optimizations and various IR (like SSA, CPS, etc.). Knowledgeable about LLVM. Familiar with IDE based on LSP and **incremental computation**.
- *Program Analysis:* **2 YoE**. Familiar with static analysis algorithms (e.g. dataflow analysis, pointer analysis, etc.).
- *System Programming:* Familiar with Arch and OS, capable of low-level development and debugging.
- *Tools:* Proficient in Emacs; comfortable working in macOS and Linux; skilled in leveraging AI.

Misc

- **Languages**: Chinese (native), English (fluent)
- **Teaching Assistant**: *Programming in Practice* (Fall 2020), *Object-oriented Design and Construction* (Fall 2021, Spring 2022), *Principles and Techniques of Compilers* (Spring 2024).