Roife

roifewu@gmail.com | roife | roife.github.io

Education

Nanjing University 2023.09 - 2026.06 (expected)

Master's Degree in Computer Science and Technology

- Member of Pascal Research Group, focus on PL and program analysis.
- TA for *Principles and Techniques of Compilers* (Spring 2024)

Beihang University 2019.09 - 2023.06

Bachelor's Degree in Computer Science and Technology

- GPA 3.84/4.00. **Outstanding Graduate of BUAA**. Awarded the **National Scholarship** 2022 (top 1.5% of the major) and multiple scholarships in university, including academic scholarships, competition scholarships, etc.
- Won **the First Prize** in the NSCSCC Compilation System Design Competition 2021, ranking 2nd overall.
- TA for *Programming in Practice* (Fall 2020), *Object-oriented Design and Construction* (Fall 2021, Spring 2022 / S.T.A.R).

Work Experience

Rust Foundation Fellowship Program (Project Fellow) 2024.09 - 2025.06 (expected)

- **Contributing to rust-analyzer**: Submitted 50+ PRs; Worked on semantic analysis, type inference and so on, reducing crashes and improving robustness. Developed new features, like code navigation on control-flow keywords, etc.
- **Community Maintenance**: Including bug fixes, user support, PR reviews, discussions in steering meetings, etc.

Projects

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

- (**Project Leader**) Designed the core architecture of the IDE, incremental computation processes, intermediate representation, semantic analysis module, etc. Also implemented most of the IDE functionalities.
- Implemented incremental semantic analysis, providing features like real-time completion, navigation, refactoring, etc.

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

- (**Independently Developed**) Huawei research project, which is also my undergraduate thesis project.
- Utilizing shape information of neural networks to perform secondary optimizations on operators, reducing runtime cost.
- Included a lightweight edge-side compiler for IR optimizations, and a trimmed LLVM for assembly code generation.
- Successfully **reduced runtime by 6**% and **target file size by 38**% of the neural network operators in test cases.

Open-source contributions

- **Rust-lang Member** (rust-analyzer contributors team). Contributed to rust-lang/rust-analyzer, rust-lang/rust-clippy, rust-lang/rustup, rust-lang/rust-mode
- llvm/llvm-project, clangd/vscode-clangd, google/autocxx, moonbitlang/tree-sitter-moonbit, yuin/goldmark, llvm/clangd-www, doomemacs/doomemacs and more on my GitHub.

Skills

- **Programming Languages**: Not limited to specific language. Especially proficient in C, C++, Java, Rust, Python, Verilog/SystemVerilog. Comfortable with Ruby, Swift, JavaScript, OCaml, Coq, Haskell, etc. (no certain order)
- **PL Theory**: Familiar with type systems, formal semantics, formal verification and theory of computation.
- **Compilers & Program Analysis**: Proficient in compilation optimizations, static analysis (like dataflow analysis, pointer analysis, etc.) and various IR (like SSA, CPS, etc.). Knowledgeable about LLVM. Capable of independently designing and implementing a complete compiler from source code to RISC-assembly.
- **IDE**: Knowledgeable in IDE based on LSP and incremental computation, especially familiar with rust-analyzer and clangd.
- **Web & Mobile**: Full-stack. Skilled in Vue, RoR, Docker, PostgreSQL, Redis; familiar with SwiftUI.
- **Tools**: Proficient in Emacs; comfortable working in macOS and Linux; skilled in leveraging AI.

Misc

- **Languages**: Chinese (native), English (fluent)