

教育背景

南京大学 2023.09 - 2026.06 (预计)  
硕士, 计算机科学与技术 | Pascal Lab, 导师: 李 | 研究方向为程序语言与程序分析  
北京航空航天大学 2019.09 - 2023.06  
本科, 计算机科学与技术 | GPA: 3.84/4.00

工作经历

Rust Foundation Fellowship Program 2024.09 至今

- 维护 rust-analyzer (Rust 语言官方 IDE)
  - 社区贡献排名 21/972, 解决 60+ issues; 参与社区日常维护工作、会议讨论、PR 审核等; 维护了语义分析、类型检查等多个模块; 为项目添加控制流导航、测试更新等多项功能; 为断行模块编写了 NEON 下的 SIMD 实现, 该模块在 ARM 平台上提速 6.5 倍;
  - 解决 v0.3.1992 P0 事故: 社区在发版 4 小时后发现会导致资源耗尽且无法结束进程的恶性 BUG。本人在 3 小时内定位到错误算法, 并设计新算法解决了问题。该紧急修复控制了事故影响范围, 避免影响全球 Rust 开发者的工作。

奖项荣誉

- 2022 年国家奖学金 (该学年专业排名 1/195); 北京航空航天大学优秀毕业生;
- 2021 年全国大学生计算机系统能力大赛·编译系统设计赛 (华为毕昇杯) 一等奖, 排名第二;
- 蓝桥杯 C++ 程序设计竞赛北京赛区一等奖, 国赛三等奖; 另获其他各类省、校级奖项与奖学金十余次。

项目开发

Vizsla · 面向芯片设计的现代化 Verilog/SystemVerilog IDE (Rust / SystemVerilog) roife/vizsla (WIP)

- 基于增量计算架构, 实现了面向可综合 SystemVerilog 的语义分析框架, 旨在为芯片设计配备现代 IDE 功能;
- 项目在功能、性能与可用性等指标上均达到业界先进水平: 已完成代码导航、重构、代码补全等数十项现代 IDE 特性, 并能够做到毫秒级延迟; 项目已通过语言服务器协议适配 VS Code、Emacs 等主流编辑器。

Ailurus · 实验性编程语言及工具链的设计探索 (Rust / MLTT) roife/ailurus (WIP)

- 基于 Martin-Löf 类型论; 实现了双向类型检查、dependent type、inductive types、pattern matching、module system 等特性;
- 项目包含解释器和 REPL; 使用 NbE 进行等价检查, 并实现了 propositional equality, 支持定理证明。

LLVM-Lite · 面向深度学习算子的轻量级端侧编译器 / 华为课题兼本科毕设 (C++ / LLVM) roife/llvm-lite

- 课题旨在利用端侧推理设备已知的形状信息对深度学习算子进行二次优化, 以减少算子运行时的时间与空间开销;
- 项目包含一个使用 LLVM-IR 的轻量级端侧编译器优化利用形状信息算子, 并使用经裁剪的 LLVM codegen 模块进行代码生成;
- 毕设获优秀评价; 成功将测试算子的推理时间降低 6%, 目标文件减小 38%; 将解析耗时降低 60%, 将解析内存占用降低 60%。

Ayame · C 子集到 ARMv7 的编译器 / 毕昇杯比赛项目, 合作 (Java / LLVM / ARM) No-SF-Work/ayame

- 主要负责完成了基于图着色的迭代寄存器合并算法, 以及指令调度等 Machine IR 上的优化; 并利用 docker 搭建了本地评测 CI;
- 项目在 SSA IR 与 Machine IR 上进行优化, 在近一半样例上排名第一, 并在 1/3 的样例上优化效果超越 gcc -O3 与 clang -O3。

开源社区贡献

- Rust-lang Member (rust-analyzer contributors team): 工作集中在 rust-lang/rust-analyzer 上, 同时也贡献过 rust-lang/rust, rust-lang/rust-clippy, rust-lang/rustup, rust-lang/rust-mode ;
- llvm/llvm-project, clangd/vscode-clangd, google/autocxx, yuin/goldmark, 更多项目见 GitHub。

专业技能

- 编程语言: 不局限于特定语言。熟悉 C, C++, Rust, Java, Python, JS/TS, (System)Verilog; 了解 Ruby, Swift, OCaml, Haskell, Coq 等。
- 程序语言理论: 了解形式语义、形式验证与计算理论相关知识, 了解 Hindley-Milner、Dependent Type 等类型系统的理论和实现。
- 编译器/虚拟机: 4 年经验。熟悉编译器的全流程开发, 熟悉多种中间代码表示 (如 SSA / LLVM-IR, DBI, ANF, MLIR 等) 和编译优化; 对 LLVM 的架构和实现有一定了解; 了解 Mark-Sweep、Copying 等常见的 GC 算法。
- IDE: 2 年经验, 熟悉基于增量计算的 IDE 架构和编译器架构, 尤其熟悉 rust-analyzer 和 clangd 的架构和实现; 了解 VS Code、Emacs 等主流编辑器的插件开发, 熟悉语言服务器协议 (Language Server Protocol) 的规范和实现。
- 程序分析: 熟悉常见静态分析算法, 如数据流分析、区间分析、IFDS 等; 熟悉基于各类敏感性的过程间指针分析算法。
- 系统编程: 了解体系结构和操作系统相关知识, 能进行汇编级的开发调试工作, 了解 Docker, GDB, CMake 等常用工具。
- 开发环境: 熟悉 Emacs; 习惯在 macOS / Linux 下工作; 能熟练使用 AI 提高工作效率。

其他

- 助教: 程序设计基础 (2020 秋), 面向对象设计与构建 (2021 秋, 2022 春), 编译原理 (2024 春)。
- 外语: 英语 (CET-6);