Ruqing Yang

yangrq.lambda@gmail.com github.com/waterlens

Research Interests

By studying how compiler technology can shape the features of programming languages, I aim to contribute to the development of new programming languages that offer strong static guarantees, intuitive error handling, and seamless cross-platform portability.

Education

Hong Kong University of Science & Technology — **Sept. 2023 - Aug. 2025 (expected)**
M. Phil. in *Computer Science and Engineering*. Supervised by Lionel Parreaux. — *Hong Kong S.A.R., China*

Zhejiang University — **Sept. 2019 - June 2023**
B. Eng. in *Computer Science and Technology*. GPA: 3.84/4.0 — *Hangzhou, China*

Projects

MLScript — **Autumn 2023 - Now**

- This is an ongoing project in HKUST TACO Lab.
- Designed an ANF-based IR with join points support and integrated it into MLsript compiler.
- Implemented a code rewriter that contains a non-duplicate partial inliner leveraging function splitting.
- Implemented a C++ backend. Using a universal object representation, and reference counting for memory management.

Calocom — **Spring 2022**

- A coursework for the course *Compilation Principle*.
- Designed and implemented a programming language with functional features like algebraic data type, closure, and pattern matching.
- Topics include type checking, closure conversion, LLVM-based code generation

SyOC — **Spring 2022 - Summer 2022**

- This is a compiler for SysY (a subset of C) language.
- Typical dataflow analysis: immediate dominator analysis, iterated domination frontier analysis for SSA construction.
- Constant propagation, CFG simplification, and dead code elimination.

MMM — **Autumn 2024 - Now**

- A small compiler for the functional MiniMoonBit language.
- Do selective CPS transformation and thunking on function calls to avoid stack overflow in the JavaScript backend.
- Implemented an efficient native backend with tree-pattern covering instruction selector and chordal graph coloring register allocator.
- Lambda lifting, loop invariant code motion, local value numbering, and guaranteed tail recursion elimination.

Publications

Smart Inlining through Function Splitting, *PLDI SRC 2025* — **April 2025**

Experience

Intern for Programming Language Tool Development, *at IDEA* — **Mar. 2025 - June 2025 (expected)**

Student Volunteer, *ICFP 2024* — **Sept. 2024**

Teaching Assistant, *Programming with C++* — **Jan. 2024 - June 2024**

Remote Research Intern, *hosted by Yizhou Zhang* — **Sept. 2022 - Jan. 2023**

Undergraduate Teaching Assistant, *Principles of Programming Languages* — **Sept. 2022 - Jan. 2023**

Skills

Programming Languages: OCaml, Rust, C/C++, Scala, Java, Python, etc.
Proof Assistant: Coq

Last Updated on April 2025