Ruqing Yang

✉ yangrq.lambda@gmail.com waterlens

RESEARCH INTEREST

I'm interested in the **design** and **implementation** of programming languages and typing systems.

EDUCATION

Zhejiang University **Sep. 2019 - June 2023**
B. Eng. *Computer Science and Technology* *Hangzhou, Zhejiang, China*
GPA: 89/100

COURSES

Data Structures and Algorithm Analysis (97)
Principles of Programming Languages (97)
Theory of Computation (90)
Operating Systems (98)
Compilation Principle (96)
Computer Architecture (93)
Computer Network (96)
Introduction to Computer System (95)

PROJECTS

Calocom | ***Rust, Compiler, Type system*** **Spring 2022**

- This is a course project where I and my teammates designed and implemented a programming language with functional features like algebraic data type, closure, and pattern matching.
- I was involved in designing the type system, the typed AST, and the middle IR.
- I wrote the type checker, transformer from typed AST to middle IR, the code generator targeting LLVM IR, the run-time library, and the standard library.

SyOC | ***C++, Compiler, Optimization*** **Spring 2022 - Summer 2022**

- An optimizing compiler for SysY (a subset of C) language.
- Used technique: Iterated domination frontier analysis for SSA -form IR construction, sparse conditional constant propagation, etc.

Rmatch | ***C++, JIT, Regex*** **Autumn 2021**

- A simple regular expression matcher with JIT support.

EXPERIENCE

Research Intern **Summer 2022 - Now**

- On the topic of bidirectional tunneling algebraic effects and co-worked.
- We aim to implement the most frequently used tail-resumptive and abortive effect handlers without heap allocation and dynamic stack inspection and at a lower cost compared with normal context-save-resume schemes
- We assure this improvement by a set of formalized transformations that limits the lifetime of partial effect handlers.
- Supervised by *Prof. Zhang Yizhou*.

Graduation Project **Autumn 2022 - Now**

- Parallelize the SMT solver in a typical CDLT(T) architecture.
- Supervised by *Prof. Yao Peisen*.

Teaching Assistant **Autumn 2022 - Now**

- In the course *Principles of Programming Languages*.
- I prepared a lab that requires students to implement Hindley–Milner type inference in a simple lambda calculus with let expression.
- I designed and wrote the auto judgement system of labs.

SKILLS

Proof Assistants: Coq
Programming Languages: C/C++, Rust, OCaml, Python, Java