Ruqing Yang

+86-13083098668 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: 3.83/4, 88/100

COURSES

Fundamental Data Structures and Algorithm Analysis (97)
Advanced Data Structures and Algorithm Analysis (88)
Operating Systems (98)
Computer Architecture (93)
Principle of Programming Language (97)

In addition to the coursework listed above, I also have been learning the following content on my own, considering my interest.

Software Foundations (Volume 1 & 2)
Types and Programming Languages
Advanced Compiler Design and Implementation
Compiling With Continuations

SKILLS

English: GRE: 155 + 169 + 3.5, CET6: 525
Programming Languages: C/C++, Rust, Coq, Haskell, Python

PROJECTS

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

- A optimized compiler for SysY (subset of C) language.
- It's designed to achieve similar performance to the normal C/C++ compiler with the -O3 level (e.g. GCC, Clang).
- Used technique: Iterated domination frontier analysis for SSA -form IR construction, sparse conditional constant propagation, etc.
- It's also a large-scale project of software engineering from a personal perspective.

Rsmix | ***Rust, Parser Combinator*** **Spring 2022**

- A ***mixfix*** parser.
- The origin idea comes from the user-defined operator syntax of the Agda language, and I'm interested to use it in my project as a flexible parser.

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

- A simple regular expression matcher with JIT support.
- The purpose of this project is to examine the possibility of compiling at the runtime. Though only supporting simple regular expression now, it is a successful prototype for me helping me explore many concepts and designs.

Oeobia | ***C++, Meta-Programming, λ Calculus*** **Spring 2022**

- Functional programming with `C++` templates.
- I developed it to examine the ability of template meta-programming. The partial template specialization serves as a way for pattern matching, and type parameter substitution as evaluation.