

# 杨汝清

[✉\\_yangrq.lambda@gmail.com](mailto:yangrq.lambda@gmail.com) [🌐github.com/waterlens](https://github.com/waterlens)

## 研究兴趣

致力于改进编程语言，提升性能并为用户提供更强的保障。设计和实现编程语言编译器的优化是我毕生的追求。

## 教育背景

香港科技大学2023 年 9 月 - 2025 年 11 月（预计）  
哲学硕士，计算机科学与工程，导师：Lionel Parreaux中国香港特别行政区  
研究方向：函数式编程语言的优化。

浙江大学2019 年 9 月 - 2023 年 6 月  
工学学士，计算机科学与技术，GPA：3.84/4.0中国杭州

## 项目经历

**Calocom**  *Rust*2022 年春季

- 《编译原理》课程的团队项目。
- 开发了一种功能丰富的编程语言，支持代数数据类型、高阶函数和模式匹配。
- 负责设计类型系统、类型化抽象语法树（AST）、对象内存表示、名称修饰风格，以及用于去糖化和必要转换的中间表示（IR）。
- 领导项目开发，实现了除词法分析和语法分析外的几乎所有组件，包括语法去糖化、语义检查、闭包转换、基于 LLVM 的代码生成（使用 Rust 库 *inkwell*）。同时，使用不安全 Rust 编写了标准库（字符串、向量等）和运行时（对象分配和程序入口）。
- 获得班级最佳课程项目。

**SyOC**  *C++*, *Python*, *ARM*2022 年春季 - 2022 年夏季

- 与朋友合作的学习项目，旨在学习编译器优化技术并参加毕昇杯编译器大赛。从零开始开发，代表浙江大学首个进入决赛的两人团队。
- 领导项目开发，设计了基于 C++ 模板的编译器框架，用于优化转换，以及基于 SSA 的中间表示，支持定义-使用和使用-定义链。
- 实现了词法分析器、递归下降解析器、mem2reg 优化（包括立即支配者分析和迭代支配边界分析以构建 SSA）、死代码消除和常量传播。
- 编写 Python 脚本，比较优化后程序与 gcc 或其他编译器的性能。

**MLscript**  *Scala*, *C++*2023 年秋季 - 至今

- 实验室联合项目。
- 设计了基于 ANF 的中间表示，扩展了连接点（join points）。
- 实现了基于控制流分析的智能内联器，用于决定内联时机，并利用函数分割技术减少内联导致的代码重复。同时负责 C++ 后端的实现，包括对象的通用内存表示、无装箱值的优化算术运算和基于引用计数的内存管理。

**QuicKaml**  *C*2023 年秋季

- 一个个人兴趣项目，为一单态语言实现了基于寄存器的虚拟机解释器，并进行了许多低级优化。
- 使用补丁版本的 LLVM，为解释器中的 VM 指令处理生成保证尾调用的代码。
- 尝试了多种技术来提高解释器的性能，包括：操作数重排，允许更高效的符号扩展；部分寄存器解码，减少 x86-64 架构上的无用移位；基于非对齐内存访问或移位和掩码的指令获取；预解码，在指令分派之前利用 CPU 流水线的并发能力进行解码掩盖延迟。

**MMM**  *MoonBit*, *RISC-V*, *WebAssembly*2024 年秋季

- 与朋友合作的 MGPIIC 大赛项目，获得第一名，遥遥领先第二名。
- 领导开发并设计了基于 MoonBit 的优化编译器框架，支持 *Mini MoonBit* 语言，包含 JavaScript、RISC-V 和 WASM 后端。
- 实现了大赛所需的所有核心优化，包括保证尾递归消除、基于寄存器压力的选择性  $\lambda$  提升、基本块拉直、死代码消除、局部值编号、公共子表达式消除、循环不变量代码移动、跳转表优化、标量替换和快速堆分配。
- 编写了 JavaScript 和 RISC-V 后端的代码生成器。为避免栈溢出，在 JavaScript 后端设计了选择性 CPS 转换和自动闭包化；在 RISC-V 后端，移植了 Cranelift 的树模式覆盖指令选择器，并实现了弦图着色寄存器分配器。
- 扩展语言功能，支持参数多态（泛型）、特设多态（通过字典传递实现的类型类）和用户定义操作符。

**RMatch**  *C++*2021 年秋季

- 个人兴趣项目，解析正则表达式并生成基于 NEA 的虚拟机字节码，随后使用 C++ 库 *xbyak* 将字节码即时编译为 x86-64 本地机器指令。

**Apple  $\mu$ Arch Bench**  *C*2024 年春季

- 兴趣项目，探索 Apple Silicon 的微架构特性，使用硬件性能计数器进行分析。

**OCaml 的 SIB 优化**  *OCaml*2025 年春季

- 共享不可变块优化。函数式编程语言常对现有数据结构进行模式匹配，即使新对象与旧对象相同，也常会分配新对象。我实现了一种可靠的优化，若对象被证明为不可变，则消除不必要的分配。
- 该优化已在 MoonBit 编译器内部使用。

单子哈希  *C*, *AArch64*2025 年春季

- 增量计算研究项目的性能关键部分。
- 使用 ARMv8 的 pmull 指令扩展了 fast-crc32 实现，加速单子组合。具体为加速在  $GF(2^{32})$  域上两个位反转多项式的乘法运算。

## 学术成果

通过函数分割实现智能内联，*PLDI SRC 2025*2025 年 4 月

## 工作与实践经历

编程语言工具开发实习，*IDEA*2025 年 3 月 - 2025 年 9 月（预计）

- 实现了一项 OCaml 优化，提升了 MoonBit 编译器的性能。
- 通过使用 tcc 编译器编译生成的 C 源代码，优化了 MoonBit 测试的本地后端编译速度，包括重构构建系统、修复 tcc 编译器中的错误，以及支持 Linux、macOS 和 Windows 的运行时库跨平台兼容性。
- 为工具链添加了基准测试功能，支持统计分析和可视化。

**ICFP 2024** 学生志愿者2024 年 9 月

**C++ 编程助教**2024 年 1 月 - 2024 年 6 月

- 设计了帮助学生理解 C++ 中指针和引用的实验课程。

远程研究实习，导师：张轶舟2022 年 9 月 - 2023 年 1 月

- 研究了词法代数效应的实现与语义，这是编程语言研究的热门话题。

本科助教，编程语言原理2022 年 9 月 - 2023 年 1 月

- 设计了基于 OCaml 的实验，帮助学生理解 Hindley-Milner 类型推断算法。
- 设置作业，帮助学生学习和使用基于评估上下文的操作语义和定界续体。
- 设计并实现了课程的在线评测系统，利用公共 GitHub 仓库和免费 CI（GitHub Actions）配额。为保护学生代码隐私，要求学生使用公钥加密代码后以 GitHub issue 形式提交。

## 技能

编程语言：精通多种编程语言，包括但不限于：

- 最常用：OCaml、Rust、C/C++、Scala
- 熟悉：Java、Python
- 有使用经验：TypeScript、JavaScript、Ruby、Haskell、Lua、Verilog、Scheme 等

编程语言理论：

- 使用 Coq 进行形式化验证。
- 熟悉基于约束的类型推断、双向类型推断等，具备丰富的类型系统知识。

编译器：

- 熟练使用和修改常见编译器框架，如 LLVM、Cranelift 等。
- 熟悉多种编程语言范式的编译，包括命令式、函数式、面向对象和动态语言。
- 熟练使用性能分析工具（如 perf、VTune、flamegraph）进行微架构级性能调优。
- 熟悉多种寄存器分配算法（迭代寄存器合并、线性扫描等）和垃圾回收算法（标记-清除、标记-压缩、三色增量、分代回收等）。
- 深入了解解释器和运行时系统设计与实现，包括各种 threading 技术、栈式 VM 和寄存器式 VM、内存管理、运行时对象表示、上下文切换等。

语言能力：

- 中文（母语），英语（良好的工作沟通能力）

最后更新：2025 年 6 月