

概要介绍

1. 项目目标

SYSY 编译器是专为 C 语言子集 SYSY 设计的编译器，它以 SYSY 源代码作为输入，经过前端翻译、中端优化、后端生成，最后得到 RISC-V 汇编代码输出。

2. 创新概述

本 SYSY 编译器是一个将 C 语言子集源代码转换为 RISC-V 指令集的编译工具。它采用 Flex 和 Bison++ 进行词法和语法分析，通过遍历抽象语法树生成类 LLVM 中间代码并划分基本块转化为控制流图。接着，编译器使用 Mem2reg 过程将内存操作转化为寄存器操作，生成静态单赋值形式的中间表示，并进行活跃性分析、插入 phi 函数、死代码消除、部分冗余消除、CFG 基本块的简化、循环优化、内联优化、伪递归消除、消除不重要边、数组相关的优化等。最终，将中端传递的 IR 进行模式匹配，并通过寄存器分配和其他后端优化，生成最终的 RISC-V 汇编代码。

3. 功能简介

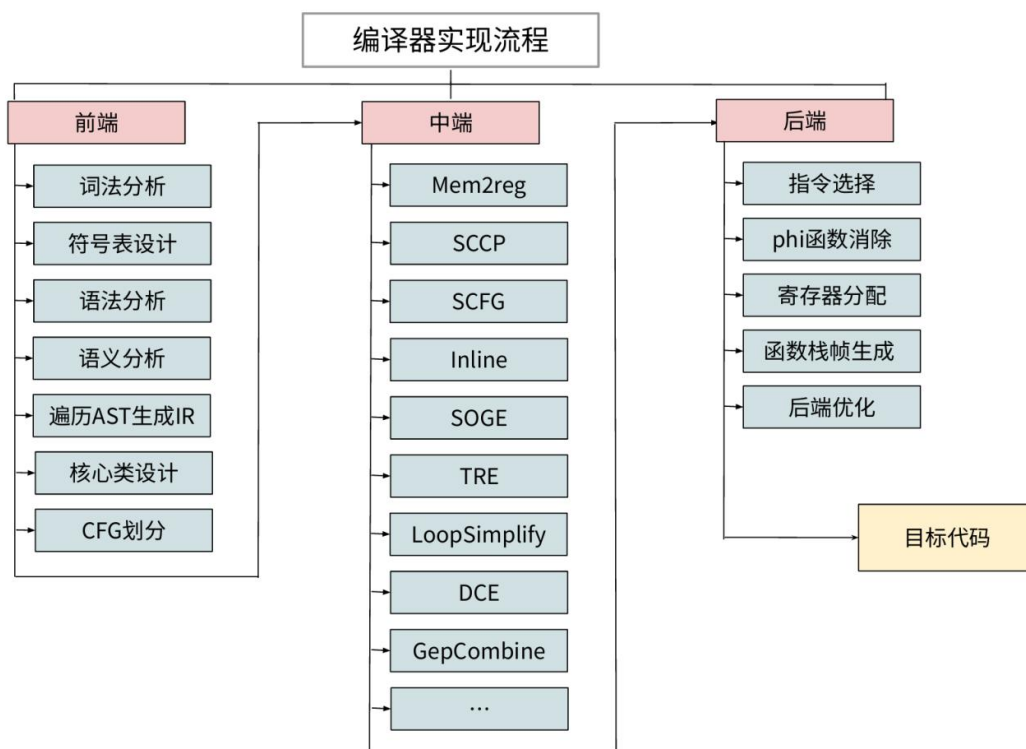


图 3.1 “SYSY 编译器实现流程”

前端翻译：该前端编译器首先使用 Flex 进行词法分析，将源代码转换为 Token 序列，

并构建符号表存储标识符信息。接着通过 Bison++ 进行语法分析，生成抽象语法树，并进行语义检查以确保程序正确性。随后基于 AST 和符号表生成 LLVM 风格的中间代码，并利用核心类管理数据流和依赖关系。最后划分基本块并构建控制流图，为中端优化奠定基础，确保后续编译阶段高效执行。

中端优化：在编译器中端阶段，系统通过控制流分析和支配树构建优化程序结构，并利用 mem2reg 技术转换为 SSA 形式，便于数据流分析。随后进行标量优化和基本块简化，提升执行效率。针对循环结构，实施循环分析、旋转、展开、并行等优化，减少开销，提高性能，为后端代码生成奠定基础。

后端生成：编译器后端将优化后的中间代码转换为目标机器指令。首先进行指令选择，匹配最优指令序列，再通过图着色算法分配物理寄存器，减少内存访问。随后优化处理包括消除 Phi 函数、简化代码，最终生成高效机器码，确保程序正确且高性能执行。

4. 结语

SYSY 编译器作为一款教学型编译工具，成功构建了从 SYSY 源代码到目标平台可执行代码的完整编译链路。该系统采用模块化设计，不仅实现了词法分析、语法解析、语义检查等基础功能，更在中间代码优化和目标代码生成环节展现出优异的性能表现。凭借良好的跨平台兼容性，可在 Linux、Windows 等主流操作系统稳定运行，为编译原理教学提供了可靠实践平台。其可扩展的架构设计为后续集成高级优化技术和语言特性预留了充足空间，既能满足当前教学需求，又能适应未来技术发展，是理论联系实践的优秀教学案例。