# Lab05-IR Optimization

> 实验名称：NJU-编译原理(2022秋)-Lab05-中间代码优化(Intermediate Representation Generation)
> 实验人员：201220096 钟亚晨
> 使用框架：贺柄毓同学提供的框架代码(Framework)
> 完成日期：2023/01/09
> 更多实现细节可见个人博客文章：(pawx2's Blog)【密码： `chcp65001` 】

## 功能实现

- 补充了**框架代码中所有TODO部分**；
- **通过了OJ平台上的绝大部分测试用例**；

## 实现思路

- 后向解释器部分：依照Tai-e实验手册中的说明，结合前向解释器实现的已有代码进行仿照实现即可；
- 可用表达式分析、常量传播、复制传播、活跃变量分析部分：依照本次实验说明中各个对应部分的传递函数、控制流约束函数、迭代算法伪代码以及框架代码注释进行选择填空即可；

## 实验感想

- 本次实验主要靠贺柄毓同学的框架进行完成，在阅读框架代码的过程中，了解并认识到了很多新的C语法，以及C中宏的使用技巧，对于函数封装、命名，以及注释如何进行书写、分类、使用有了新的接触。这也是上大学以来不多的阅读源代码的经历，阅读代码的确是学习语言如何使用、学习技巧、语法的很好的方法。今后应当加强对代码阅读能力的锻炼，多阅读优秀的源代码，写代码固然重要，但读代码同样很重要。
- 中间代码优化，尤其是全局代码优化部分，确实非常依靠数据流分析技术，在实现和阅读实验指导的过程中，尽管各类数据流分析在学习完后感受到”就那么回事“，但是一想到如何从0→1想到这样的分析方法便顿感头大。中间代码优化和数据流分析的内容庞杂且多，自己甚至感觉到其像一门”小课程“，也借此实验对其进行了一遍梳理，感受到计算机前辈们经年累月构建出这样的一套体系/方法确实非常巧妙。并且通过对比**前四个实验**中使用的内容和**本次中间代码优化**所使用的技术，从感觉上而言，中间代码优化使用到的各种技术确实更加”数学“、更加繁杂，也更加深刻体会到冯老师在授课时所讲的”**在做编译器研究时，如果你仅仅只是做了前端的基本部分，而没有做优化相关的内容，那么实际上你什么也没做**“，是的，编译器从开始一直到目标代码生成部分，如果不考虑优化，事实上已经具备了成熟的流程，而优化部分则是百花齐放，多种多样的优化技术、方法，从**广度**上确实是远大于从源代码→目标代码的这条流程。

## 致谢

最后，本次实验对贺柄毓同学提供的框架代码表示由衷的感谢！👍