

1. 支配边界插入Phi函数 (placePhi)

1. 构造数据流信息：
 - 创建 DataFlowInfo 对象，提取函数中的所有变量
 - 计算变量的活跃性信息 (liveness)，包括每个语句的 livein 和 liveout 集合
2. 初始化Phi函数记录：
 - 为每个基本块初始化一个集合 A_phi，用于记录该块中已插入的 Phi 函数变量
3. 遍历变量：
 - 对于每个变量 a：
 - 如果变量没有定义（如函数形参），则跳过
 - 初始化工作集 W，包含所有定义变量 a 的语句
4. 插入Phi函数：
 - 遍历工作集 W：
 - 对于每个基本块 n_block，获取其支配边界集合
 - 对于支配边界中的每个块 Y：
 - 如果变量 a 的 Phi 函数尚未插入，并且变量 a 在块 Y 的 liveout 集合中：
 - 创建一个新的 Phi 函数，将其插入到块 Y 的语句列表中
 - 将变量 a 添加到 A_phi[Y] 中

2. 重命名变量 (renameVariables)

1. 构造数据流信息：
 - 创建 DataFlowInfo 对象，提取函数中的所有变量
2. 初始化计数器和栈：
 - 为每个变量初始化一个计数器 Count，用于生成唯一的变量名
 - 为每个变量初始化一个栈 Stack，用于记录当前作用域内的变量版本
3. 递归重命名：
 - 从入口基本块开始，递归处理每个基本块：
 - 替换变量的使用：
 - 对于每条语句，使用栈顶的变量版本替换 use 集合中的变量
 - 重命名变量的定义：
 - 对于每条语句，生成新的变量版本，更新 def 集合，并将新版本压入栈
 - 处理后继块：
 - 对于每个后继块中的 Phi 函数，添加当前块的变量版本作为参数
 - 递归处理子块：
 - 根据支配树递归处理子块
 - 回溯：
 - 在回溯时，弹出当前块中定义的变量版本

Git Graph

GRAPH

Auto

master

25/5/8 17:34 Ethan Cao

25/5/8 17:26 Ethan Cao

添加注释 Ethan Cao

HW7 pass Ethan Cao

25/5/8 17:14 Ethan Cao

25/5/8 16:36 Ethan Cao

25/5/8 16:30 Ethan Cao

25/5/8 16:23 Ethan Cao

25/5/8 16:04 Ethan Cao

25/5/7 22:02 Ethan Cao

25/5/7 21:32 Ethan Cao

25/5/7 19:16 Ethan Cao

25/5/7 17:16 Ethan Cao

25/5/7 17:03 Ethan Cao

HW6 pass Ethan Cao

25/4/24 19:51 Ethan Cao

finish HW5 Ethan Cao

HW5 pass Ethan Cao

全局类表 Ethan Cao

每个类单独一个类表 Ethan Cao

pass test8 Ethan Cao

pass test6 Ethan Cao

25/4/16 16:13 Ethan Cao

25/4/16 15:42 Ethan Cao

pass test5 Ethan Cao

25/4/16 13:46 Ethan Cao

25/4/16 13:27 Ethan Cao

25/4/16 13:18 Ethan Cao

WSL: Ubuntu-24.04

Build