编译原理实验五报告

周羽萱 211220074 1813156367@qq.com

一. 实现功能

在词法分析、语法分析、语义分析和中间代码生成程序的基础上，使用数据流分析算法等消除效率低下和无法被轻易转化为机器代码的中间代码，从而将 C--源代码翻译成的中间代码转化为语义等价但是更加简洁高效的版本。

二. 程序编译

使用 makefile。

三. 代码实现

使用框架代码，完成框架代码中的 todo 部分即可。

- src/IR_optimize/available_expressions_analysis.c
  对可用表达式进行分析。
  a. OutFact[Entry] = **空集**。
  b. InitFact = **全集**。
  c. 交汇运算是**交集**运算
- src/IR_optimize/constant_propagation.c
  对常量传播进行分析。
  a. 控制流约束：根据文档内容。

  控制流约束函数与传递函数略有不同：

  UNDEF∪c=c　　NAC∪c=NAC

  c∪c=c　　c1∪c2=NAC

  b. 计算二元运算结果的 CPValue 值
  考虑 v1 和 v2 都是 NAC 的情况下，结果也是 NAC。
  考虑 v1 和 v2 都是常数，且 v2=0,运算符是除号，结果是 UNDEF。
  c. **前向**传播
  d. 函数参数初始化为 **NAC**
  e. **传播过程：**
  处理赋值语句 (IR_ASSIGN_STMT):
  函数提取赋值语句的左值(def)和右值(use_val),使用 Fact_update_value 函数更新 fact，将左值变量映射到右值的常量值。
  处理操作语句 (IR_OP_STMT):
  提取操作语句中定义的变量（def）和操作数（rs1_val, rs2_val)。
  使用 calculateValue 函数计算操作结果，并使用 Fact_update_value 更新 fact，将定义的变量映射到计算出的值。
  处理其他语句：

对于不属于上述两类的其他语句，如果语句定义了新的变量（def），则将其映射到 NAC。

- src/IR_optimize/copy_propagation.c
  类似，不再赘述。
- src/IR_optimize/live_variable_analysis.c
  对活跃变量进行分析
  a. **后向**传播
  b. 交汇运算是**并集**运算
  c. 先执行 **kill**,后执行 **gen**
  d. 死代码消除：如果在后续代码中没有被使用，就可以视为死代码。
- src/IR_optimize/solver.c
  仿照前向求解器的实现，完成后向求解器即可。

四． 总结与感悟

本次实验在借助框架代码的情况下比较简单，相当于带我把优化部分的主要内容复习了一遍。

感谢助教哥哥批改！