

实验报告

181850236 张子辰 (基础班) 181850236@smail.nju.edu.cn

1. 程序运行和本地环境:

进入"/Lab/Code"目录,直接在终端运行 make 命令,即可编译生成 parser 文件,运行 parser *.cmm *.ir 后,即可得到虚拟机使用的 ir 文件。

我的本地运行环境为 Ubuntu amd64 18.04, flex 2.6.4, bison 3.0.4, gcc 7.5.0。

2. 程序功能

我的程序实现了本次实验的必做任务和所有选做任务。

本次实验中我的主要代码在 intercode.c/h 文件中,其主要框架和 L2 中类似,主要集中于修改 Stmt 和 Exp 的相关函数,符号表的建立和维护和 L2 中几乎完全相同。

为了配合 L3 的实验,我在 L2 中定义的 HashNode 结构中增加了 id 和 is_param 域,前者用于将所有变量名替换成 vi (i 为一个整数) 的格式,防止出现下划线开头的变量名或以 ti 格式 (程序中临时变量的格式) 命名的变量;后者用于标记该变量是否为函数参数,在选择输出变量还是输出地址时有帮助。

我的程序实现了数组的赋值 copyArr(), 是以 4 个字节为单位,逐个进行赋值,直到抵达较小数组的末尾。

Stmt 和 Exp 的修改主要参考了讲义中给出的方法,对于数组和结构体,为了解决嵌套问题,我使用递归调用的方式来生成代码,这样虽然使生成的代码行数增多,但是增加了程序的鲁棒性。

3. 程序亮点

我使用了上一级拔尖班同学维护的[测试库](#)进行了测试,可以通过 272/274 的测试样例,其中错误的样例为 impossible.cmm 和 ZMpro.cmm,由于测试文件太长,我几乎无法进行 debug。两个测试文件都集中于函数调用的测试,由于选做任务中结构体和数组也可用于函数调用,使我对什么时候要输出地址,什么时候要输出变量产生了不小的困惑,导致了未知的 bug。等下发新的测试样例后我将重新排查。