编译原理实验 3 实验报告

--151220097 孙旭东

0.实验环境

本次的实验环境具体为：

系统：ubuntu 16.04　　GCC：version 5.4.0

Flex：version 2.6.0　　Bison：version 3.0.4

我的邮箱地址是 **248381185@qq.com**，如果检查中出现问题请助教联系我。

1. 实现功能

本次实验实现了中间代码生成，也就是 IRSIM 可以接受的代码。在实验二生产的符号表的基础上，程序再次遍历语法树，根据各个节点对应的语义生成对应的代码，并将代码存入代码链表中。生成代码的思路和实验二类似，通过深度优先遍历语法树来生成。实现了 3.2，即程序**支持高维数组**，**支持数组作为参数**。在生成代码之后会进行一定的优化，优化有时可以达到**减少 1/3 甚至更多**指令的效果。

2.编译方法

由于涉及的文件较多，所以编写了 makefile 来帮助进行程序编译。

make compile　　--编译生成 parser 程序

make lab　　--测试 test/lab3 文件夹中的.cmm 文件

make run　　--测试 test/1.cmm

make clean　　--删除程序

具体的，如果有一个文件 test.cmm 希望进行编译，那么步骤为：

（1）**make compile**　//删除旧程序，编译生成新程序

（2）**./compiler test.cmm** //对 test.cmm 进行分析

或者可以 make test 批量测试。

对 test.cmm 分析完毕之后生成的中间代码会保存为文件 test.cmm.ir 中，也就是说**生成的代码文件和测试文件会在同一文件夹中**。

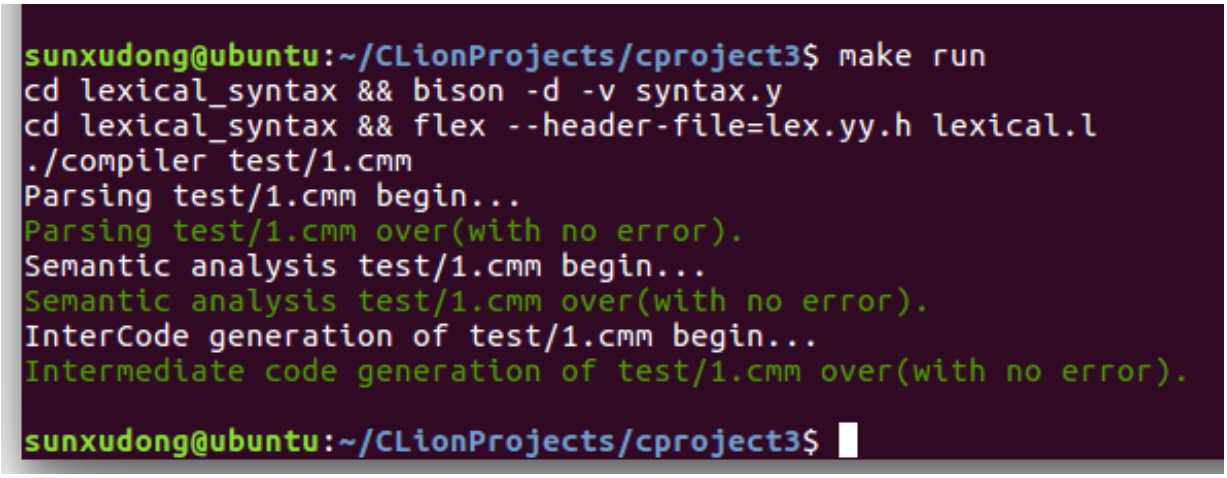
3.额外完成

完成了选做 3.2，即高维数组。

红色字体提示错误，绿色字体提示正确无误。

4.运行演示

make run，输出信息：



5.完成度与亮点

- 完成了必做的所有内容
- 完成了选做 3.2（高维数组）
- 代码优化（删除冗余赋值，表达式化简，fall）