

编译原理实验报告

中间代码生成

王梓轩 121220100

完成的功能点

在词法分析、语法分析和语义分析程序的基础上，将 C--源代码翻译为中间代码。该程序将中间代码输出成线性结构，从而可以使用虚拟机小程序来测试中间代码的运行结果。

具体包括：基本功能 1~7，完成了代码的简单优化等等。

测试样例 1、2 已通过。

编译及运行方法

依次输入：

```
make
./parser XXX1 XXX2
//在 irism 中使用该.ir 文件
//make clean 可以删除编译过程中生成的一些文件
```

(XXX1 为测试用的 C 文件，XXX2 为生成的.ir 文件，与 parser 在同一文件夹下)

实现方法

本实验的目的是实现中间代码，所以仍然需要遍历一次语法树。针对该树中的每个节点，有相对应的处理方式。考虑到该节点的语义，选择是否生成中间代码，并决定所生成中间代码的样式。

具体来说，是按照实验介绍中提供的翻译模式，并利用一个结构体链表实现三地址代码的打印语句。每次需要生成三地址代码的时候，则插入在链表尾部。最后打印的时候遍历整个链表即可。

亮点

1. 有全局的 var 和 label 技术器，可以按照顺序生成变量和标签。
2. 对于 Exp 中的条件判断语句和 if、while 中的条件判断有不同的中间代码生成形式。
3. 对于 read 和 write 两个函数，在 Exp 对应的语句处理的时候，识别这两个函数名，然后做出特殊的处理（即类型设置为 READ 及 WRITE）。
4. 优化一：对于连续的两个 LABEL，其实有一个是冗余的，所以会遍历代码打印链表，对于生成的连续两个标签，利用链表的先向指针，往前遍历，找到使用冗余标签的所在语句，换成另一个，最后从链表中删除冗余标签即可。
5. 优化二：对于一些跳转指令而言，若出现了“if *** goto label1”，“label1: goto label2”，这样的语句，可以删除第二条语句，直接将第一句改为“if *** goto label2”。
6. 优化三：尝试加入一些 ifFalse，以便减少三地址代码个数。

其他说明

我还试图完成高维数组的实现，但实现后经过大量调试，还是有一些错误，因此没有加入已完成的功能点中。