stage-3 实验报告

姓名：郑友捷　学号：2021010771　班级：计14

工作内容

step 7

step7部分引入了作用域与块语句，需要对作用域与符号表的划分进行修改。

词法分析部分

新增了`block`节点用于识别块语句

语义分析部分

程序设置有一个作用域栈，存储当前不同层级的作用域信息。当语义分析扫描到块语句时，在作用域栈上新开一个局部作用域，用于存储当前作用域的符号表。当块语句分析结束，则将该作用域从栈中弹出。

在声明部分，检查变量是否重名时，需要在当前作用域的符号表进行检查。

在检查到一个变量被使用时，为了获取其符号信息，应当从当前作用域的符号表中开始寻找。若找不到则寻找父作用域的符号表，逐层上传。

中间代码生成部分

只需要正确获取变量对应的符号即可，已在语义分析部分完成。

目标代码生成部分

这部分需要使用数据流分析的方法判断某一个基本块是否可达。若不可达，则不为其分配寄存器。具体修改包括：

在控制流图`cfg.py`部分，当建立起控制流图之后，对其采取类似于拓扑遍历的操作，获取所有可达的基本块的编号，并为其添加判断函数，判断当前编号为i的基本块是否可达。

在为基本块分配寄存器的`brutergalloc.py`部分，先判断当前基本块是否可达，若不可达则不为其分配寄存器。

step 8

step 8部分引入了循环语句

词法语法分析

为`for`、`while`、`do while`语句各添加了一个AST节点，并完成词法分析补全。

对`for`语句进行词法分析时需要注意，其初始化语句、更新语句与终止条件均可能为空，因此需要额外判断。

语义分析

1. 对于`for`循环，其初始化语句、更新语句与终止条件部分自成一个作用域，因此在分析到初始化语句时需要先开一个局部作用域，在分析完终止条件后要关闭该局部作用域，从而防止该作用域的变量与for循环的正文冲突。
2. 当进入`while`、`for`、`do while`部分时，需要让当前点的循环次数增加1。在遇到`break`与`continue`语句时需要判断当前的循环次数是否为0，为0则报错。

中间代码生成

中间代码较为重要的便是确定不同标签与语句的生成顺序。

对于`for`循环，考虑`continue`与`break`，一种可行的tac顺序为：

```
init_statement
begin_label
cond_statement
body_statement
continue_label # continue对应的label
update_statement
j beginlabel # 跳转回开始标签
breaklabel # break对应的label
```

对于`while`循环，一种可行的顺序为：

```
beginlabel
cond_statement
body_statement
looplabel
j beginlabel # 跳转回开始标签
breaklabel
```

对于`do while`循环，一种可行的顺序为：

```
beginlabel
body_statement
looplabel
cond_statement
j beginlabel # 跳转回开始标签
breaklabel
```

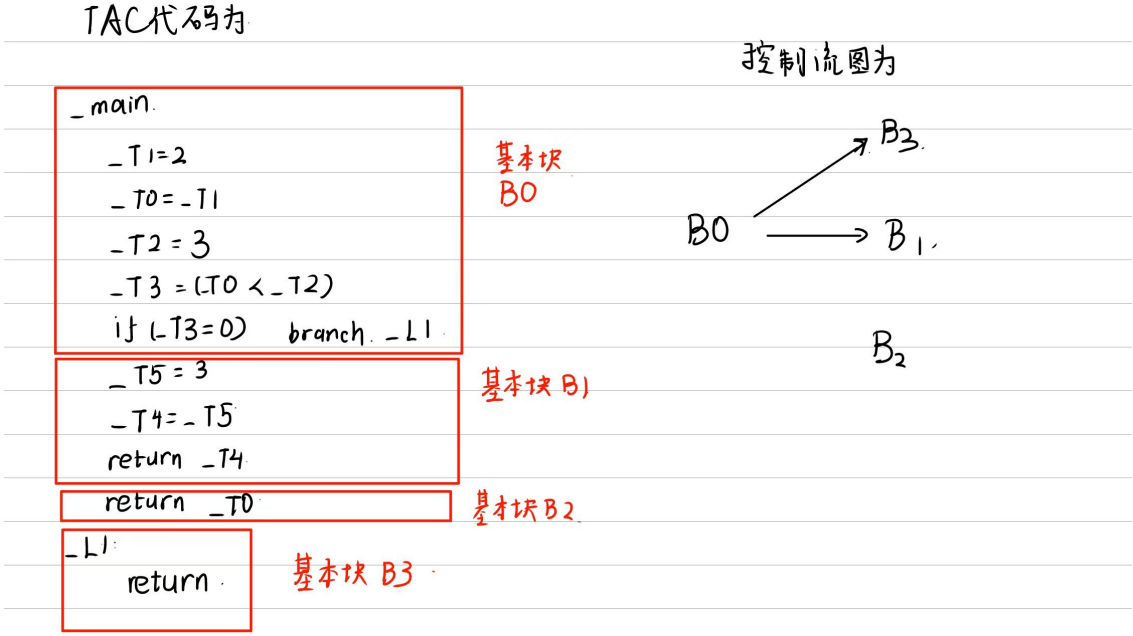
注意：当`do while`执行`continue`时，会先执行条件判断语句。

目标代码生成

只需要按照TAC的标签，额外加入`break`与`continue`对应的跳转语句即可。

思考题

1.



2. 第二种翻译方式更好。若触发了了continue，则
   - 第一种翻译方式会在触发continue语句之后跳转到 `continue_label`，之后跳转到 `beginloop_label`，再执行 `cond` 语句，进行条件跳转，共执行四条指令。
   - 第二种翻译方式会在触发continue语句之后跳转到 `continue_label`，之后执行 `cond` 语句，再进行条件跳转，共执行三条指令。
   - 对于其他正常执行的语句与更新，两者执行条数相同。

   因此第二条执行的指令条数更少。