Stage-3 Report

计83 李天勤 2018080106

Experiment Background

The purpose of this experiment is to complete stage-3 tasks, which include step7 作用域和块语句 and step8 循环语句, while supporting break and continue within loops.

Step7

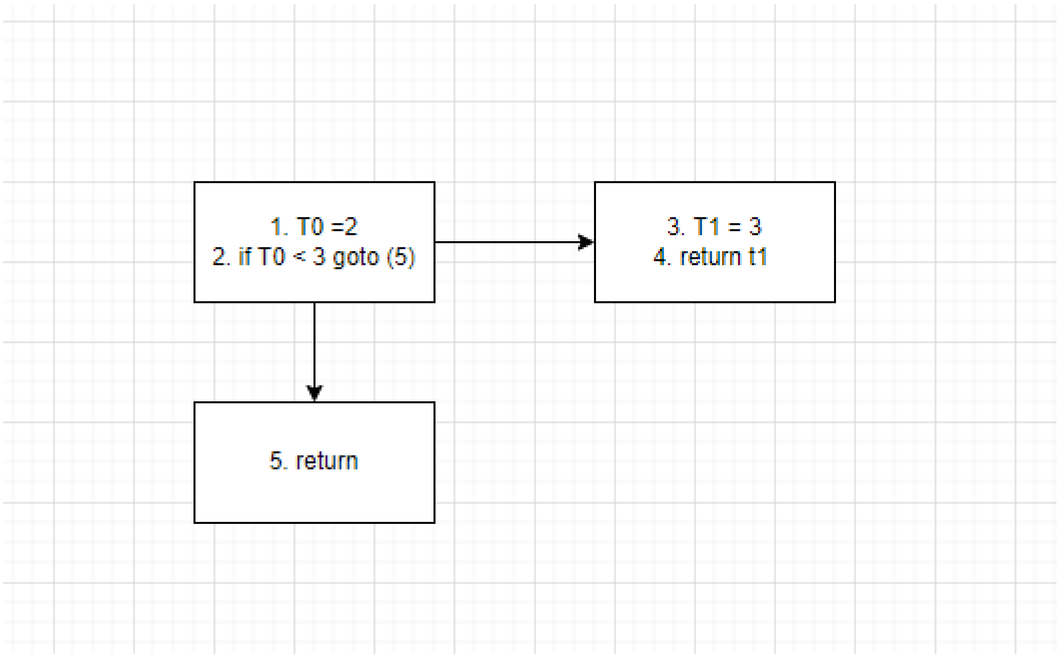
Here, we want to expand the support for block statements. The most important key implementation is making sure that when the same variable name appears (but in different places), we must be able to identify which variable and variable name is for each block statement. We, also need to make sure that in the same block, the same variable cannot be declared more than once.

In this experiment, we have to update `CompStmt` functionality, introducing the scope stack, handling both global scope and local scope implementations.

Thinking Questions

1. Please draw the control flow diagram of the MiniDecaf code below

```
int main(){
 int a = 2;
 if (a < 3) {
     {
         int a = 3;
         return a;
     }
     return a;
 }
}
```



Step 8

Here, we need add support for loop statements, and break/continue. We have to handle all for loop implementations that can occur in C, support do while loop, and make sure that break/continue statements cannot appear outside the loop.

We add three new ast nodes located in `ast_for_stmt.cpp`, `ast_do_while_stmt.cpp`, `ast_break_stmt.cpp` and `ast_cont_stmt.cpp`.

We also have to update the following files just like the previous steps. First we need to declare the functions in the hpp files, including

```
translation.hpp
ast.hpp
visitor.hpp
```

Then we have define our access functions in `build_sym.cpp and type_check.cpp`. We also have to do our intermediate code generation located in `translation.cpp`. Here, we have to slightly adjust the previous while loop functionality to handle continue and break statements as well as implement for loops and do while loops.

Thinking Questions

1. 将循环语句翻译成 IR 有许多可行的翻译方法，例如 while 循环可以有以下两种翻译方式：

第一种（即实验指导中的翻译方式）：

1. `label BEGINLOOP_LABEL`：开始下一轮迭代
2. `cond 的 IR`
3. `beqz BREAK_LABEL`：条件不满足就终止循环
4. `body 的 IR`
5. `label CONTINUE_LABEL`：continue 跳到这
6. `br BEGINLOOP_LABEL`：本轮迭代完成
7. `label BREAK_LABEL`：条件不满足，或者 break 语句都会跳到这儿

第二种：

1. `cond 的 IR`
2. `beqz BREAK_LABEL`：条件不满足就终止循环
3. `label BEGINLOOP_LABEL`：开始下一轮迭代
4. `body 的 IR`
5. `label CONTINUE_LABEL`：continue 跳到这
6. `cond 的 IR`
7. `bnez BEGINLOOP_LABEL`：本轮迭代完成，条件满足时进行下一次迭代
8. `label BREAK_LABEL`：条件不满足，或者 break 语句都会跳到这儿

从执行的指令的条数这个角度（`label` 指令不计算在内，假设循环体至少执行了一次），请评价这两种翻译方式哪一种更好？

【答】The second of the two methods are better.

Let us assume that the amount of instructions of cond is n_c and the amount of instructions taken by the body is n_b, and that the loop runs exactly $m \geq 1$ times, while continue or break will decrease the number of instructions by d, the the number of instructions made by the two loops above can be calculated. Where the total number of instruction by the first one is

$((n_c + 1) + (n_d + 1)) * m + n_c + 1 - d$ and the second $n_c + 1 + (n_d + n_c + 1) * k - m$