

stage-3 实验报告

秦若愚 2019011115

实验内容

step-7

实验目标

- 增加块语句的支持

实验内容

在符号表构建的阶段，需要在每次开启一个代码块时，新建一个作用域并压栈；而当退出代码块时，要弹栈关闭此作用域。具体则是使用 `ctx.open(Scope(ScopeKind.LOCAL))` 和 `ctx.close()` 这两条语句来完成。

此外，在寄存器分配过程中，需要对基本块是否可达做判断，若基本块不可达，则无需分配寄存器。

step-8

实验目标

- 增加对循环语句的支持
- 增加对 break/continue 的支持

实验内容

由于实验框架中已经实现了 while 语句和 break 语句，因此只需实现 continue、for、do while 这三种语句。

首先，在 frontend/lexer/lex.py 中添加 `continue`、`for`、`do` 这三个保留字。

其次，在 frontend/ast/tree.py 中添加 For、DoWhile、Continue 三种抽象语法树节点类，并在 frontend/ast/visitor.py 中实现对应的 visit 函数。

接着，在 frontend/lexer/ply_lexer.py 中添加 continue、for、do while 这三种语句的解析规则。

然后，在 frontend/typecheck/namer.py 中添加对 For、DoWhile、Continue 三种节点的解析函数。

最后，在 frontend/tacgen/tacgen.py 中完成中间代码生成阶段对 For、DoWhile、Continue 这三种节点的处理函数。

思考题

step-7

1. 答：

MiniDecaf 代码的 TAC 代码如下（基本块用空格分开）：

```
main:
    _T1 = 2
    _T0 = _T1
```

```

_T2 = 3
_T3 = (_T0 < _T2)
if (_T3 == 0) branch _L1

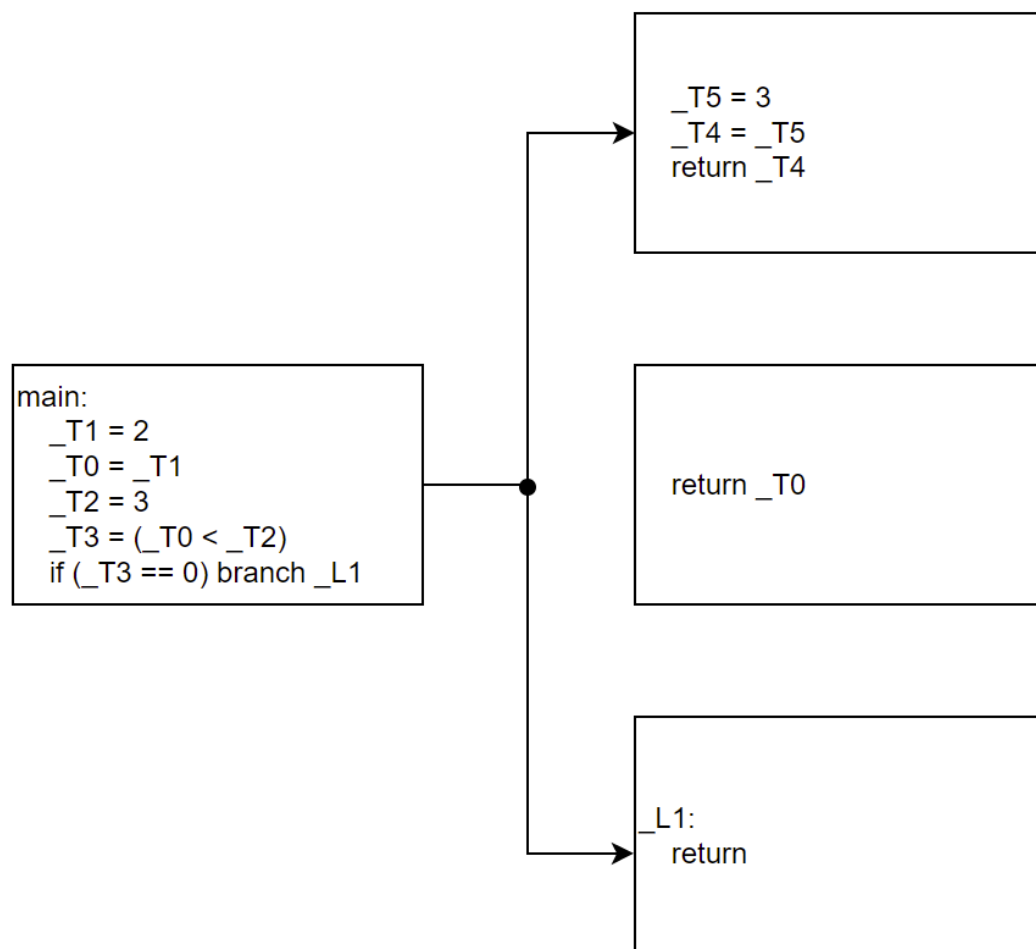
_T5 = 3
_T4 = _T5
return _T4

return _T0

_L1:
return

```

控制流图如下：



step-8

1. 答：

假设 cond 和 body 中执行的指令数分别为 m_c 和 m_b ，循环执行了 n 次，则

第一种翻译方式执行的指令条数为：

$$N_1 = (m_c + 1 + m_b + 1) \times n + m_c + 1 = (n + 1)m_c + nm_b + 2n + 1$$

第二种翻译方式执行的指令条数为：

$$N_2 = m_c + 1 + (m_b + m_c + 1) \times n = (n + 1)m_c + nm_b + n + 1$$

因此第二种翻译方式更好。