

# stage-2 实验报告

计 93 王哲凡 2019011200

## 实验内容

### step5: 局部变量和赋值

首先在符号表建立时，添加对于 Declaration（声明）、Assignment（赋值）、Identifier（符号）的访问，具体修改在 frontend/typecheck/namer.py 中：

- 对于 Declaration，需要注意对于重复声明的报错即 DecafDeclConflictError；在 VarSymbol 的新建时，需要注意其标识符名应该为 decl.ident.value 而非 decl.ident.name。
- 对于 Assignment，主要需要左侧不为左值的错误（测例中实际未涉及）。
- 对于 Identifier，需要注意标识符未定义的问题，同样应该使用 ident.value 而非 ident.name。

而对于中间代码生成，同样可以按照代码中的注释提示，对于上述三种进行访问，具体修改在 frontend/tacgen/tacgen.py 中。

而后端则只需在 backend/riscv/riscvasmmmitter.py 中新添加 visitAssign() 函数即可：

```
1 def visitAssign(self, instr: Assign) -> None:
2     self.seq.append(Riscv.Move(instr.dst, instr.src))
```

### step6: if 语句和条件表达式

框架中给出的部分，在 step6 中实际上已经完成了 if 语句的部分，所以只需要在 frontend/tacgen/tacgen.py 中按照类似 if 语句的方式修改 visitCondExpr() 函数即可。

具体而言，我们不需要考虑 otherwise 为空的情况（A ? B : C 的三部分必须完整），且需要为 ConditionExpression 设置正确的临时值。

我实现的思路是，在 otherwise 分支中，添加将结果赋值给 then 分支结果的 Assign 语句，最后再把 then 的结果设置为整体 ConditionExpression 的结果：

```
1 mv.visitAssignment(expr.then.getattr("val"), expr.otherwise.getattr("val"))
2 mv.visitLabel(exitLabel)
3 expr.setattr("val", expr.then.getattr("val"))
```

## 实验思考题

### step5: 局部变量和赋值

```
1 | addi sp, sp, -16
```

2. 首先，符号表中除了存储变量，还应该存储变量的版本数（即当前已经经过了几次声明）。

其次，变量定义时，在符号表中查询变量冲突的过程，对于在符号表中未出现过的变量，需要初始化版本数；对于出现过的变量，则不报错，而是更新版本数（加一）。

然后，对于符号表的查询，应该将不同版本的相同名称变量视为不同的符号，并且返回的是最新版本的变量。

最后，对于题目中出现的带初始化的声明，需要首先访问初始化部分保证其中涉及的变量不会被指向刚构造的符号。

具体实现时，类似于将第 `cnt` 次声明的变量 `var` 重新命名为 `var_{cnt}`，且在变量冲突时，选择为其命名为 `var_{cnt + 1}`，并及时计数 (`cnt <- cnt + 1`)。

### step6: if 语句和条件表达式

1. Python 框架中，使用了产生式优先级的方式，即设置优先匹配 `if else` 而非 `if` 的方式，具体来说，在 `frontend/parser/ply_parser.py`，定义 `p_if_else()` 函数先于 `p_if()` 函数，使得解析时，优先考虑完整 `if else`。
2. 首先应该按照顺序访问 `cond`、`then`、`otherwise`（不带条件），之后的条件跳转修改为 `BNE` 语句，即条件正确时跳过，跳过部分只须添加将 `otherwise` 的结果赋值给 `then` 分支结果的 `Assign` 语句即可：

```
1 | expr.cond.accept(self, mv)
2 | expr.then.accept(self, mv)
3 | expr.otherwise.accept(self, mv)
4 | skipLabel = mv.freshLabel()
5 | mv.visitCondBranch(
6 |     tacop.CondBranchOp.BNE, expr.cond.getattr("val"), skipLabel
7 | )
8 | mv.visitAssignment(expr.then.getattr("val"), expr.otherwise.getattr("val"))
9 | mv.visitLabel(skipLabel)
10 | expr.setattr("val", expr.then.getattr("val"))
```