

Stage 2 报告

程思翔 2021010761

1 实验内容

1.1 step 5

语义分析: 实现了声明语句、赋值语句、标识符的语义分析函数.

在 `visitDeclaration` 中检查定义域中是否存在定义冲突, 在 `visitIdentifier` 中需要检查变量是否存在.

```
1  # frontend/typecheck/namer.py:class Namer
2  def visitDeclaration(self, decl: Declaration, ctx: Scope) ->
    None:
3      if ctx.lookup(decl.ident.value):
4          raise DecafDeclConflictError(decl.ident.value)
5      # 构造 VarSymbol 对象, 将其加入符号表, 并设置 decl 的 symbol 属性
6      symbol = VarSymbol(decl.ident.value, decl.var_t.type)
7      ctx.declare(symbol)
8      decl.setattr("symbol", symbol)
9      if decl.init_expr:
10         decl.init_expr.accept(self, ctx)
11
12 def visitAssignment(self, expr: Assignment, ctx: Scope) -> None:
13     # 参考 `visitBinary` 的实现
14     expr.lhs.accept(self, ctx)
15     expr.rhs.accept(self, ctx)
16
17 def visitIdentifier(self, ident: Identifier, ctx: Scope) ->
    None:
18     symbol = ctx.lookup(ident.value)
19     if not symbol:
20         raise DecafUndefinedVarError(ident.value)
21     # 设置 ident 的 symbol 属性
```


目标代码生成: 为赋值语句实现了 `visitAssign` 函数进行目标代码生成.

```
1 # backend/riscv/riscvasmemitter.py:class RiscvAsmEmitter
2 def visitAssign(self, instr: Assign) -> None:
3     self.seq.append(Riscv.Move(instr.dst, instr.src))
```

2 思考题

2.1 step 5

1. 我们假定当前栈帧的栈顶地址存储在 `sp` 寄存器中, 请写出一段 **risc-v 汇编代码**, 将栈帧空间扩大 16 字节 (提示1: 栈帧由高地址向低地址延伸; 提示2: risc-v 汇编中 `addi reg0, reg1, <立即数>` 表示将 `reg1` 的值加上立即数存储到 `reg0` 中).

答: 汇编代码为:

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

2. 有些语言允许在同一个作用域中多次定义同名的变量, 例如这是一段合法的 Rust 代码 (你不需要精确了解它的含义, 大致理解即可):

```
1 fn main() {
2     let a = 0;
3     let a = f(a);
4     let a = g(a);
5 }
```

其中 `f(a)` 中的 `a` 是上一行的 `let a = 0;` 定义的, `g(a)` 中的 `a` 是上一行的 `let a = f(a);`.

如果 MiniDecaf 也允许多次定义同名变量, 并规定新的定义会覆盖之前的同名定义, 请问在你的实现中, 需要对定义变量和查找变量的逻辑做怎样的修改 (提示: 如何区分一个作用域中**不同位置**的变量定义?).

答: 在语义分析部分 `frontend/typecheck/name.py` 中:

- `visitDeclaration` 定义变量时, 不查询是否有同名变量并抛出同名异常, 因为顺序执行的程序中, 新的定义会覆盖之前的定义. 先访问初始化语句, 再访问变量声明, 并覆盖原始变量. 这是因为如果存在重名变量定义, 可以先根据变量的初始值计算出其新定义值.
- `visitIdentifier` 查找变量无需修改, 变量被新定义的变量覆盖后, 只需寻找当前作用域中的符号, 即是最新定义的变量.