

# 实验报告—step9

2018011340 计 83 郭峥岩

## 一、 实验内容

### 1. 整体框架

本次实验中 step9 的整体架构与之前的框架相比没有发生变化。只是在名称解析的时候添加了对于函数声明和定义的检查，并将相关信息存入符号表中，同时根据 RISC-V 的函数调用规则进行调用。保存寄存器等行为也是根据实验指导书的指导进行。后续的 IR 生成和汇编生成在后边会进行介绍。

### 2. Step9 的功能实现

#### (1) 实现程序的语法规则的改动：

支持函数的声明和定义，支持函数调用。

#### (2) IR 类的更改和产生：

添加了 irFunction 和 irCall 两个类别，用于函数调用的时候进行使用。同时完善之前的有关函数的栈帧分配的 ir 类。

#### (3) 汇编代码的产生：

ret 的对应的汇编代码不发生变化，增加新增的 IR 类对应的汇编代码，有多少个参数执行几次 pop，之后将返回值（在 a0 寄存器中）保存到栈帧之中。

### 3. 实现思想和细节

名称解析的时候添加对函数的支持，添加了相应的 visit 函数，标记当前的解析函数，同时支持了记录参数地址，对函数的变量表进行了解析，并且对声明和定义进行了一致性判断。

同时在名称解析的时候添加了重定义等的判断

## 二、 思考题

1. MiniDecaf 的函数调用时参数求值的顺序是未定义行为。试写出一段 MiniDecaf 代码, 使得不同的参数求值顺序会导致不同的返回结果。

解答:

给出的代码如下所示:

```
int fun(int x,int y){
    return (x>y)
}

int main() {
    int a = 0;
    fun(a=a+1,a=a+1);
    return b;
}
```

## 三、 参考资料

助教所写的代码: md-dzy branch

## 四、 总结

本次的实验过程即增加几条语义分析, 对于函数的解析由于之前有了对变量解析的基础, 因此实现起来较为容易, 需要注意函数调用时的规定, 按照实验指导书上的正常顺序进行并未遇到什么太大的问题。