

# 编译原理 stage 2 报告

---

傅子轩 2020010742 计01

## 主要改动

---

1. 修复了 `parser` 关于 `IfExpr` 的错误优先级
2. 在 `build_sym` 里为所有的函数，变量声明，左值正确建立并设置了符号
3. 在 `type_check` 里检查类型，和 `IfStmt` 的 `false_branch` 和 `false_branch` 的类型
4. 在 `translation` 里为存在 `false_branch` 和不存在 `false_branch` 的 `IfStmt` 分别进行了翻译。
5. 在 `translation` 里为 `IfExpr` 做了短路翻译
6. 在 `riscv_md` 里为 `Tac::ASSIGN` 做了翻译

## 思考题

---

### step 5

1.

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

2.

同一个作用域中出现同名定义时，从符号表中取消对旧符号的引用，并添加一个新的作为新的符号。但不删除旧符号，因为它仍可能被之前的语句所引用。查找逻辑无需改变。

或者不在符号表中检查同名定义，而是在查找时寻找最后定义的符号。

### step 6

1.

`then` 和 `else` 作为运算符，同级，右结合。产生无 `else` 的 `if` 语句的产生式优先级等同于 `then` 运算符。如此一来，在 `shift-reduce` 时会优先选择 `shift`。

2.

先把两个表达式的计算翻译成代码，再使用分支语句从中选择一个作为条件表达式的值。