编译原理 MiniDecaf 编译器实验
Step 3 实验报告

李祥泽
2018011331
lixiangz18@mails.tsinghua.edu.cn

说明: 本 Step 是与 Step 4 一并实现的, 因此没有单独的 Commit.

实验内容

实现二元数学运算符 `+`, `-`, `*`, `/`, `%`

依照实验指导书中提示的二元运算语法, 编写 g4 文件, 并在对应的 Visitor 中实现 IR 生成和 IR 到汇编的生成.

指导书中给出的上下文无关语法已经很好地考虑了运算符优先级与结合性, 因此直接采用了 (修改了一些名字以符合个人代码习惯).

思考题

请给出将寄存器 `t0` 中的数值压入栈中所需的 RISC-V 汇编指令序列; 请给出将栈顶的数值弹出到寄存器 `t0` 中所需的 RISC-V 汇编指令序列.

```
addi sp,sp,-4
sw t0,0(sp)
```

```
lw t0,0(sp)
addi sp,sp,4
```

语义规范中规定 "除以零, 模零都是未定义行为", 但是即使除法的右操作数不是 0, 仍然可能存在未定义行为. 请问这时除法的左操作数和右操作数分别是什么? 请将这时除法的左操作数和右操作数填入下面的代码中, 分别在你的电脑中和 RISCV-32 的 qemu 模拟器中编译运行下面的代码, 并给出运行结果.

```
#include <stdio.h>

int main() {
  int a = 左操作数;
  int b = 右操作数;
  printf("%d\n", a / b);
  return 0;
}
```

左操作数是 -2^{31} (`0x80000000`), 右操作数是 -1.

在 WSL 下用 gcc 9.3.0, 加参数 `-O0` 的运行结果是 `Floating point exception (core dumped)`; 加参数 `-Og` (或更高优化等级) 的结果是 `-2147483648`.

在 WSL 下用本实验所提供的交叉编译工具链编译并在 qemu 运行的结果是 (所有优化等级均如此) `-2147483648`.

Honor Code

主要参考实验指导书实现.