

# 第四次实验报告

201250044 王星云

## 一、程序实现功能、实现方式、精巧设计

### (一) 程序实现功能

程序将使用SysY语言书写的源代码翻译为LLVM IR，本次实验翻译

#### 1. main函数

- main函数无参数且返回值为int类型
- 仅包含return语句

#### 2. 表达式

- 在return时不仅仅会返回一个单独的数字，而是返回一个常量表达式
- 常量表达式仅包含整形面值常量及其运算

### (二) 实现方式

#### 1. 遍历语法树

- 设计一个继承SysYParserBaseVisitor的MyVisitor类来遍历语法树并生成LLVM IR，这个类的返回值类型是LLVMValueRef。

#### 2. 引入并初始化LLVM

- 在MyVisitor类中import LLVM并且初始化LLVM。

#### 3. 创建模块和公共变量

- 为需要翻译的文件创建模块，定义了module与builder与i32Type等的成员变量，以及zero, negative1与positive1等LLVMValueRef类型的常量，用于之后的表达式求值。

#### 4. 添加函数、基本块

- 访问到函数定义时为module添加function（函数）并为function添加basicBlock（基本块）

#### 5. 获得表达式的值

- 访问到return语句时，需要添加return中间指令。
- return语句的返回值通过访问表达式子节点得到。

#### 6. 输出中间代码

- 参考助教给出的文档，输出到指定文件中。

### (三) 精巧设计

1. 使用switch语句判断单目运算符并进行相应处理

## 二、有趣的现象和印象深刻的bug

### (一) 有趣的现象

在LLVM中提供多种除法，其中UDIV表示无符号除法，SDIV表示有符号除法，本次实验应该使用SDIV。

### (二) 印象深刻的bug

我一开始写完程序，程序能输出函数定义等中间代码，但是无法输出return语句等的中间代码。

经过与示例代码的逐行比对，发现是缺少这行代码：

```
1 //选择要在哪个基本块后追加指令
2 LLVMPositionBuilderAtEnd(builder, block); //后续生成的指令将追加在
  block1的后面
```

在创建块后，需要选择在哪个块后追加指令，这些之后追加的指令才能输出