MyLang

Scrum2

编译原理课程作业第二次迭代阐述及后期设计文档

黄志强 5120809022

陈嘉南 5130379006

石嘉昊 5130379072

# 前端设计

## 词法分析

使用正则表达式表示词法，对于词法所对应的规则，主要包含两部分操作：

1. return相应的token来进行语法规则匹配
2. 使用yylval变量来存储相应的值。

## 语法分析

使用文法来表示语法规则，该部分主要包含以下操作：

1. 声明union来与词法分析建立联系，存储token的值
2. 写出语法规则，注意conflict和优先级的问题
3. 对于匹配的语法，建立相应的树的节点。

## 语义分析

在建立好语法树之后，一部分不匹配的错误已经被检测，另外一部分数组越界等错误在语义分析中进行，维护一个symbol table（详细设计在后端设计中描述），根据节点类型执行相应的操作，符号表包含符号的各种信息，比如类型，值，范围等等，这样一边进行语义分析，一边发现错误。

由于语法树本身是根据程序的顺序进行构建的，因此在做语义分析的时候，只需要先遍历左节点，后遍历右节点，递归调用就可以了。

# 后端设计

## 输出格式

MyLang的输出格式为LLVM IR，QuickSort的预计IR输出见qsort.ll文件(可正常执行)。

对于IR的部分基本格式进行了一些修改以便后端IR输出，例如取消了双重/多重判断/循环时的跳转优化，将所有判断/循环统一为一种方式；将一些输出函数所需要的符号首先自动生成为全局变量(例如换行符、格式符)，则减少在使用时大量临时变量生成；将数组参数的声明修改为固定格式等。

## 预期设计方案

#### 变量设计

首先维护一个map保存所有全局变量以及函数，对于每一个函数维护一个map保存其local变量以及一个数组(或map)保存临时寄存器%n的位置和使用情况(0代表不能使用, 1代表i32…….;如果太多了考虑使用map)

#### 错误检测

要处理的代码将由前端词法语法分析保证部分正确性。类型错误和数组越界错误(由于无指针所以单纯的检测上下界)将会在翻译过程中通过与维护的相关变量数据的对比同步完成

#### 翻译设计

对于每一个固定的语法结构采用固定的结构化方式进行翻译。例如对while() endwhile结构，首先设置一个br，然后翻译判断跳转语句，然后设置一个br，然后翻译结构内部成分，到endwhile设置一个br(在此之前，每新创一个临时寄存器，都将跳转语句中的寄存器值++；这个设计可能对实现不友好，在后期实现如果存在严重问题可能考虑修改)完成结束语句，并将相应local变量的状态设置为0(不能使用)。

#### 预期效果

预期的输出必须能够通过llvm/clang运行，尽量与设计的预期输出相同并保证正规性。