

# 编译原理实验报告一

151220135 许丽军 xulj.cs@gmail.com

## 零、实验进度描述

完成了所有的必做内容和选做要求

## 一、实验内容

### 实现的功能

利用GNU工具Flex和Bison，对c--语言编写的源代码的进行词法分析和语法分析，并在标准输出打印分析结果。

### 如何实现功能

#### 1. 编写Flex源代码 *lexical.l*

- 过滤空白符DELIM（空格，制表符，回车）
- 实现选做要求1.3，过滤"/"和"/..."形式的注释
- 利用RE和相应函数识别词素，返回词法单元（在syntax.y中定义）。因为Flex依次尝试每一个规则，RE的排序需满足条件：1)识别float的规则在识别int,DOT(.)的规则之前；2)识别RELOP(!,==,...)的规则在识别ASSIGNOP(=),NOT(!)；3)识别id的规则放在最后

```
digit  [0-9]
letter [_a-zA-Z]
RE(float) = ({digit}+\.{digit}+)|((({digit}*\.{digit}+)|({digit}+\.{digit}*)))[eE][+-]?{digit}+
RE(int) = (0|[1-9]{digit}*)|(0[0-7]+)|(0[Xx][0-9a-fA-F]+)
RE(ID) = {letter}({letter}|{digit})*
```

- 对未定义的字符报告词法错误(Error type A)
- 对于不匹配的"/"注释报告语法错误(Error type B)

#### 2. 编写Bison源代码 *syntax.y*

- 定义词法单元以及声明其属性，在lexical中使用
- 定义语法规则和对应的语义动作
- 解决二义性和冲突：1) 规定词法单元的优先级和结合性；2) 利用%prec解决悬空else问题；3) 利用%prec解决MINUS(-)的不同含义（减号OR取负）的不同优先级问题。
- 在报告语法错误（Error type B）的同时，利用error产生式进行错误恢复。error主要放在全局变量定义、局部变量定义、函数定义、语句的产生式中。

#### 3. 定义语法分析树的结点类型，编写构建和输出语法树的函数接口 *node.c node.h*

#### 4. 定义main函数读取文件输入，进入分析过程并输出结果 *main.c*

## 二、编译和运行

- 编译并生成可执行目标文件：make
- 运行可执行目标文件：./parser your\_file\_name 或 make test [TESTFILE = your\_file\_name]
- 以pretest/目录下的所有文件为输入，批量运行：make test-all
- 清除中间生成文件：make clean-temp
- 清除所有生成文件：make clean