

# 实验报告

151220037 何知涵

## 1. 功能实现

### 1.1 词法分析

#### 1.1.1 c 语言下基本的词法

基本同附录。

#### 1.1.2 空白符号的处理

为了得到的词法单元更加简洁，加入了空白符号的判断，对应空处理。

### 1.2 语法分析

#### 1.2.1 c 语言下基本的文法

基本同附录

#### 1.2.1 if-else 的二义性处理

用讲义中所讲借助算符优先级的方法处理

### 1.3 语法树的构造和打印

#### 1.3.1 构造

定义结构体 Nod，在词法和语法分析的过程中，维护当前节点并将其插入语法树，这里由于 yylval 等过程中的变量均为临时或局部变量，所以为了完整地保留整个结构体，在插入语法树的过程中并不是单纯地插入当前节点，而是另外分配内存，将当前节点保存到内存中，再将复制出来的节点插入到语法树。

#### 1.3.2 打印

在整个语法规约的过程成功执行完之后，即 yyparse() 函数成功返回，便会调用递归函数 print\_tree 打印整个树，这里根节点保存为全局变量。

#### 1.3.3 空串规约的处理

空串一开始没有加处理函数，发现会出不可控的重复 bug，后来空串也单独构造节点加入到树中，只是打印语法树的时候会忽略。

### 1.4 错误处理

#### 1.4.1 错误报告

词法错误在 . 的处理函数中打印出来，而语法分析则在 syntax.y 中对 yyerror 函数重定义，至于出错行数的输出，词法分析直接维护 yylineno，语法分析中将 yylineno 作为外部全局变量引用。

#### 1.4.2 错误恢复

暂时除了讲义提供的对声明(stmt)，语句块(compst)和表达式(exp)的 error 产生式外，还加入了全局定义(ExDef)的 error 产生式。

## 2. 编译方法

直接用 Makefile 编译生成可执行文件 parser，同时定义 make run 指令直接测试前 4 个必做内容样例。