

实验一报告

1. 实现的功能

1. 识别C--词法中未定义字符或者不符合定义的字符：**lexical.l**文件中对词法进行定义
2. 识别八进制、十进制、十六进制整数，指数形式的浮点数，`"/"/`和`"/ * ... */`形式的注释：利用正则表达式识别

```
// lexical.l
digit [0-9]
letter [a-zA-Z]
dec [1-9]{digit}*
oct 0+[0-7]*
hec 0[xX][0-9A-Fa-f]+
INT {dec}|{oct}|{hec}
FLOAT (([1-9]{digit}*)|0)(\.{digit}+)?([eE][+-]?{digit}+)?

NOTELINE "//"
NOTEALL "\\*(\\/(\\*)*(((\\^*\\/(\\/(\\*)*)))*(\\*\\/))
```

3. 对包含词法或语法错误的输入文件报错，对正确的输入文件打印语法树：

```
// 错误打印
// syntax.y
int yyerror(char* msg) {
    get_error = 1;
    if(error_lineno != yylineno){
        fprintf(stderr, "Error type B at line %d: %s\n", yylineno,
msg);
        error_lineno = yylineno;
    }
}

// 语法树打印在 multitree.c
// 语法树函数结构体声明 multitree_head.c
// token识别 syntax.y
思路：识别对应token时将其相关信息传递给trans_tree函数并生成节点、插入节点，最后识别完成打印树。
```

4. 对二义性文法的处理：显式指定算符的优先级和结合性

```
// syntax.y
%right ASSIGNOP
%left OR
%left AND
%left RELOP
```

```
%left PLUS SUB
%left STAR DIV
%right MINUS NOT
%left LP RP LB RB DOT

%nonassoc LOWER_THAN_ELSE
%nonassoc ELSE
```

5. 错误恢复：

利用bison中错误处理的机制,将error符号和某些语句进行匹配,从而完成再同步;
如通过Stmt : error SEMI;
将出现错误后的符号丢弃直至SEMI符号,从而完成恢复

2. 编译

在Code文件夹下 make