

lab1

211275009 陈铭浩

211275009@smail.nju.edu.cn

实现功能

1. 可以识别C--程序中的词法错误--错误类型A（出现 C--词法中未定义的字符以及任何不符合 C--词法单元定义的字符）
 - 实现思路：主要在lexical.l文件中进行编码，包括：
 - a. 依据附录A中给出的C--的词法规则（**token**），定义各种词法的模式（正则表达式）和动作。
 - b. 对于未被规则匹配的文本（如非法的字符或数字），打印出错信息，指出错误的类型（“**Error type A**”）、发生的位置（行号）以及错误的内容。
 - c. 使用`%option yylineno`来自动维护当前行号`yylineno`，这样可以在报告错误时使用。
 - d. 通过`YY_USER_ACTION`宏定义来更新每个词法单元的位置信息（起始行、起始列、结束列），便于后续的语法分析阶段使用。
2. 可以识别C--程序中的语法错误--错误类型B（出现不符合 C--语法中定义语法规则的语句）
 - 实现思路：主要在syntax.y文件中进行编码，包括：
 - a. 依据附录A中给出的C--的词法单元和语法单元定义终结符和非终结符，并用产生式构建语法规则。
 - b. 优先级和结合性定义：对于二元运算符和其他需要考虑优先级和结合性的情况，使用`%left`, `%right`, `%nonassoc`等声明来指定运算符的优先级和结合性，以确保表达式被正确解析。

c. 通过在产生式中使用**error**标记来捕获和处理语法错误。当分析器遇到不符合任何语法规则的输入时，将尝试匹配包含**error**的产生式，例如
ExtDef : error SEMI;。这样可以在遇到错误时提供一个同步点，从而跳过一部分输入直到遇到预期的符号（如;），继续进行语法分析。

d. 在与**error**符号关联的动作中调用**yyerror**函数，打印出错误信息。例如，
{error_num++; yyerror("Syntax error in high-level definitions");}。这里，**yyerror**函数被用来输出具体的错误信息，并增加错误计数。

3. 完成选做2.1要求（识别八进制数和十六进制数，并实现若有不符合词法定义的八进制数和十六进制数，给出输入文件有词法错误（即错误类型 A）的提示信息）

- 实现思路：利用正则表达式额外定义八进制和十六进制数的模式，并额外新增不符合词法定义的八进制、十六进制数的错误处理规则

```
0[xX][0-9a-zA-Z]* {printf("Error type A at Line %d: Illegal hexadecimal number \'%s\'.\n", yylineno, yytext);error_num++;}  
0[0-9a-zA-Z]+ {printf("Error type A at Line %d: Illegal octal number \'%s\'.\n", yylineno, yytext);error_num++;}  
. {printf("Error type A at Line %d: Mysterious characters \'%s\'.\n", yylineno, yytext);error_num++;}
```

4. 构造输入文件的语法树，对于没有任何词法或语法错误的输入文件，将构造好的语法树按照先序遍历的方式打印每一个节点的信息

- 实现思路：

a. 在**tree.h**中定义**struct**类型**TreeNode**（语法树的基本单元），包含节点的基本信息。

```
// define tree node  
typedef struct TreeNode {  
    char* name;  
    char* value;  
    //NodeType type;  
    int lineNo;  
    struct TreeNode** children;  
    int childrenCount;  
} TreeNode;
```

b. 在**tree.c**中主要实现三个函数，创建节点 (**createNode**)，添加节点 (**addNode**)，打印树 (**printTree**)，其中在添加子节点时采用动态分配内存的方式 (**realloc**)。

c. 在**lexical.l**和**syntax.y**中分别调用定义好的数据结构或函数，从而生成语法树，并在需要时打印。

编译方式

- 1 在Lab1/Code文件夹下
- 2 `$ make parser`
- 3 然后用Code/parser替换Lab1下的parser（当然，在提交的版本中笔者已经替换过了）
- 4 在Lab1文件夹下执行
- 5 `$./parser <测试文件名>`