《算法设计与分析》软件设计文档

### 项目名称及成员

一个简易的递归下降语法分析器

张顺&&唐翼腾

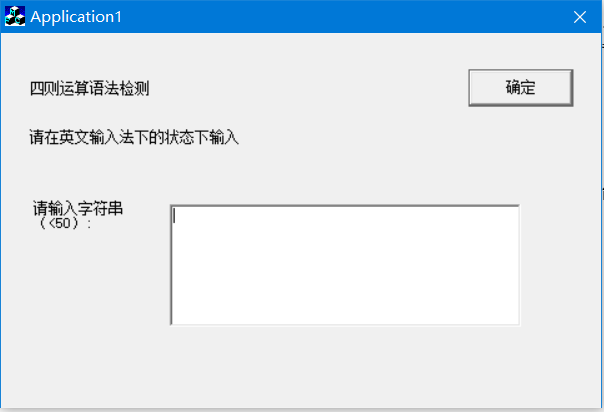
### 项目背景

递归下降（recursive descent）法，这是一种适合手写语法编译器的方法，且非常简单。递归下降法对语言所用的文法有一些限制，但递归下降是现阶段主流的语法分析方法，因为它可以由开发人员高度控制，在提供错误信息方面也很有优势。就连微软C#官方的编译器也是手写而成的递归下降语法分析器。

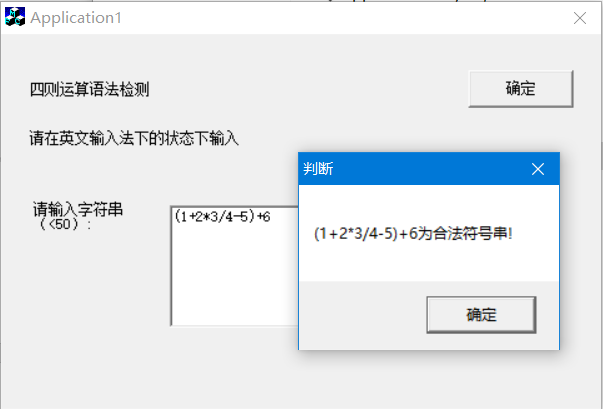
### 功能说明

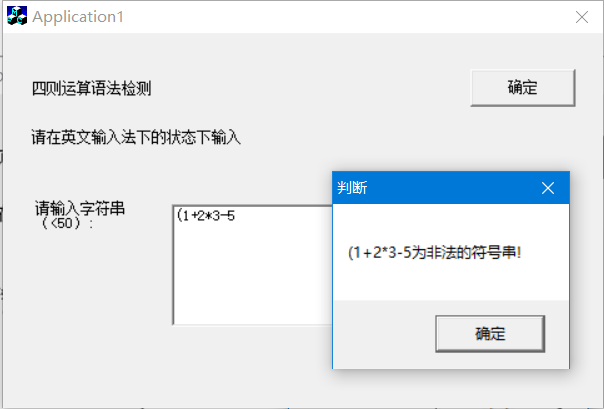
1. 该语法分析器能够分析词法分析器的结果，即单词二元式。在输入单词二元式后，能输出分析的结果。

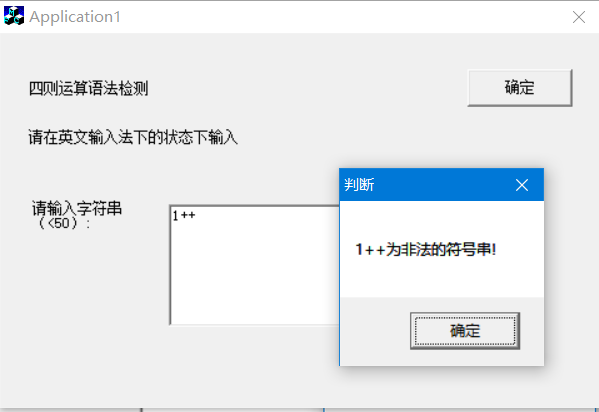
输入式子：



输入完后点击确定：





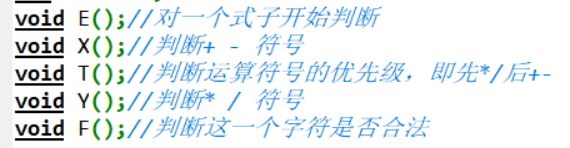


以上为显示的结果。

### 核心算法设计

#### 算法思想

本程序的实现方法主要就是函数的递归调用，在程序前面先对文法中所涉及到的对应的函数进行声明，以免在用到的时候提示函数无声明的错误提示：

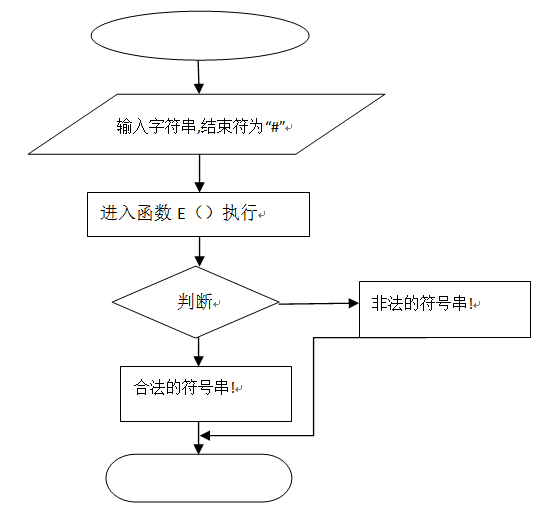


遇到一个式子，先判断是否有左括号，如果有则调用E函数对括号里的式子开始一次新的判断。

开始判断式子里一个符号后面的一个字符（或者括号里的第一个字符）时，看它是否属a~z /A~Z(作为未知数),0~9的范围，如果还是运算符号，则是非法的符号串，如果符合，则开始判断运算符号（即F函数）。

判断运算符号时，先判断 \*或/（Y函数），如果符合，则判断 /或\* 的下一个字符，即在Y函数中调用F函数和Y函数。

直到遇到右括号或者 +或- ，结束Y函数递归进入下一个式子的判断（E函数）或者+和- 的判断（X函数），X函数里面同样调用F函数和X函数进行递归操作。如果递归完成而且前面有左括号却没有遇到右括号，同样为非法字符串。



#### 算法复杂的分析

因为要对每个字符进行分析，所以复杂度至少为O（n）.

#### 与其他算法的比较（可选）

如果用暴力判断优先级，就会产生大量的if else 函数，复杂度可想而知。