构造能处理完整程序的递归下降翻译器

17计二许红凯 320170941570

任务介绍

对递归下降分析器技能型改造，使其能够一边处理，同时完成语法分析和中间代码翻译

输入

一个完整的源程序

输出

与输入对应的一个四元式序列

题目

对实验六程序进行升级改造，使得程序对于输入的一个完成的源程序，在对其做递归下降分析的同时，生成等价的四元式序列，一遍完成

程序功能说明

1. **词法分析**

   扫描源程序，根据词法规则，识别单词。如果产生词法错误，则显示错误信息、位置，并试图从错误中恢复（简答的恢复方法是忽略该字符或单词）继续扫描

2. **语法分析**

   对源程序作语法分析，确定是否属于LittleC文法，同时揭示出程序的内在结构

3. **语法错误检查**

   根据LittleC的文法规则设置检测手段，通过查错子程序或一些查错语句，报告源程序出错位置、性质等，直至整个程序结束为止

4. **语义分析与生成目标代码**

   在语法分析的基础上，进行语义分析，生成输入源程序的目标代码。输入源程序的目标代码可以生成四元式序列

代码思路

1. 对保留字、运算符界符、自定义关键字进行编码，便于在语法分析及语义分析时识别、判断与定位

2. 对于语义分析，可拆分为识别当前扫描的单词是否为自定义关键符号以及是哪一个关键符，从而确定这一语句类型，确定类型后调用相关函数再去识别该语句的组成成员来进行具体的语句解析

3. 对于有的赋值语句和普通算术表达式，都以标识符开头，在识别关键符来确定语句类型时可以暂时归为一类，然后再识别语句中的运算符可以区分两者。对于算数表达式，等号右部分算式内的运算符前后的算子均存在；对于标识符开头的赋值，用等号连接左右两部分后，右部仅有一个算子且没有运算符

4. 对代码块或源程序文件的语义分析，以分号（；）、花括号（}）为语句断点，要注意后序语句的类型分类

代码测试

文本输入：

```
main()
{ int a,b,max;
a=10;  b=1;
if (a>b) max=a;
else max=b;
while(a>0)
do{
b=a+b*a;
a=a-1;
}
}
```