语法分析器报告

学号：71Y16123 姓名：刘钊伟

# 一、目的

了解掌握语法分析器的基本内容，提高程序设计能力，掌握自顶向下和自底向上的语法分析

# 二、内容

使用 LL(1)方法 设计一个语法分析器， 对于指定的CFG，输入为字符串流，要求输出它的推导产生式

# 三、实现方法

消除二义性、消除左递归，提取公共子表达式、计算该文法的first、follow表，构建分析表，然后根据分析表实现分析器

# 四、设定

相关文法如下：

1、S🡪AB

2、B 🡪 +AB

3、B 🡪 -AB

4、B 🡪 ε

5、A 🡪 CD

6、D 🡪 \*CD

7、D 🡪 /CD

8、D 🡪 ε

9、C 🡪(S)

10、C 🡪 i

# 五、计算步骤

First和Follow集合

|  |  |  |
| --- | --- | --- |
| 非终结符 | First | Follow |
| S | ( i | $ ) |
| A | ( i | + - ε $ ) |
| B | + - ε | $ ) |
| C | ( i | \* / + - ε $ ) |
| D | \* / ε | + - ε $ ) |

分析表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 非终结符 | 输入符号 | | | | | | | |
| ( | ) | I | + | - | \* | / | $ |
| S | 1 |  | 1 |  |  |  |  |  |
| A | 5 |  | 5 |  |  |  |  |  |
| B |  | 4 |  | 2 | 3 |  |  | 4 |
| C | 9 |  | 10 |  |  |  |  |  |
| D |  |  |  | 8 | 8 | 6 | 7 | 8 |

# 六、重要数据结构描述

栈 S：存储文法符合序列

字典 table：存储分析表

数组P:存储产生式

数组T:存储终结符

数组NT：存储非终结符

# 七、核心算法描述

利用上面的预测分析表，构建表驱动的预测语法分析

输入：一个串w, 预测分析表

输出：如果w符合文法，则输出w的一个最左推导，否则给出一个错误提示

核心方法：输入缓冲区中是字符串 w$， 文法的开始符号S位于栈顶，它的下面是$

*#设置p使它指向w的第一个符号，p为指针*

*#令X = 栈顶符号*

while( X != $) *#栈非空*

{

if ( X 等于p所指向的符号 a) 栈弹出， p向前移动一个位置

else if ( X 是一个终结符号) error()

else if ( 预测表[X, a] 为空) error()

else if ( 预测表[X, a] = X--> Y1Y2...Yk)

{

输出产生式 X--> Y1Y2...Yk

弹出栈顶符号

将 Yk, Yk-1, ...Y1压入栈中， Y1位于栈顶

}

令X = 栈顶符号

}

核心算法的具体实现如下：

while(x != '$'):

if(x == w[i]):

S.pop()

i += 1

elif(x in T):

print("Error")

break

elif(-1 == table.get((x, w[i]), -1)):

print("Error")

break

elif(table.get((x, w[i]))):

p\_num = table.get((x, w[i]))

run\_s.append((P[p\_num-1][0], '-->', P[p\_num-1][1]))

d\_p = P[p\_num-1][1]

S.pop()

temp\_s = Stack()

for s in d\_p:

temp\_s.push(s)

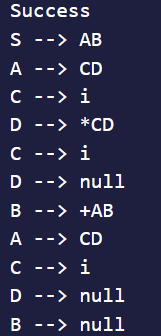
while not temp\_s.is\_empty():

S.push(temp\_s.pop())

x = S.peek()

# 八、运行时情况

对于字符串 i\*i+i, 结果为：



# 九、问题与解决的办法

最初对于不带有$的字符串，分析器运行时会出错，所以在最开始加了一个手动检测机制。

# 十、感想和建议

本次实验通过使用LL（1）算法，实现了一个对简单字符串进行分析的语法分析器，对自顶向下的LL（1）算法有了更好的掌握。同时，本次实验的缺陷是并没有实现和语法分析的联动，而是使用单个终结字符来代替对应的token，同时对于出错的字符串，只能指出该字符串是错误的，并不能指出具体错在哪里。这两点都是值得改进的地方。