

一．LR1语法分析器的算法描述：

1. **构造项目：**

LR(1)项目的一般形式为了[A→α•β, a] ，其中A→α•β 是LR(0) 项目，a是终结符或者#。a称为展望符。

归约项目[A→α•, a]意味着：当它所属的状态呈现在栈顶且后续输入符号为a，才可以把栈顶的α归约为A。

1. **First集**

对每一个文法符号X∈VT UVN 构造FIRST(X): 应用下列规则,直到每 个集合FIRST不再增大为止.

1. 如果X∈VT,则FIRST(X)={X}.
2. 如果X∈VN,且有产生式Xàa…,则把a加入到FIRST(X)中;若Xàe也是一个产生式, 则把e加入到FIRST(X)中.

(3)如果XàY…是一个产生式且Y∈VN, 则把FIRST(Y)\{e}加到FIRST(X)中; 如果XàY1Y2…Yk是一个产生式, Y1,...,Yi-1∈VN,而且对任何j, j∈[1, i-1], e∈FIRST (Yj ), (即Y1Y2…Yi-1=>e) , 则把FIRST(Yi )\{e}加到FIRST(X)中; 特别是,若所有的FIRST(Yj )均含有e , j=1,2,…,k, 则把e加到FIRST(X)中.

1. **Closure运算，构造项目集规范族**

假定I是文法G’的任一项目集，构造I的闭包 CLOSURE(I)的方法是：

• I的任何项目都属于CLOSURE(I)；

• 若[A→α•Bβ, a]属于CLOSURE(I)， B→η是一个

产生式，那么，对FIRST(βa)中的每个终结符b,

将[B→ • η,b]加入CLOSURE(I)；

• 重复上述两步骤直至CLOSURE(I)不再增大为止

1. **Go函数**

函数值GO(I,X)定义为GO(I,X)=CLOSURE(J)，其中

J={任何形如[A→αX•β, a]的项目 | [A→α•Xβ, a]属于I}

1. **构造LR（1）分析表**

• 步骤 ：

• 构造拓广文法G’的LR(1)项目集规范族C={I0 ,I1 ,…,In}

• 从Ik构造语法分析器的状态k，状态k的分析动作如下：

• 如果[A→α·aβ, b]在Ik中，且GO(Ik , a)= Ij，则置action[k, a]为sj ，即“移动(j, a)进栈”，

这里要求a必须是终结符

• 如果[A→α·, a]在Ik中，则置action[k, a]为rj ，即按照rj归约，其中j是产生式A→α的序号

• 如果[S’→S·, #]在Ik中，则置action[k, #]为acc，表示接受

• 状态k的转移按照下面的方法确定：如果GO(Ik , A)= Ij,那么goto[k, A]=j

• 其余表项设为出错

• 初始状态是包含[S’→·S, #]的项目集构造出的状态。

1. **根据LR（1）分析表，移进规约输入串**

分析栈中的串和等待输入的符号串构成如下形式的三元组: (S0S1S2…Sm, #X1X2…Xm, aiai+1 …an#)

1. 其初态为: (S0 , #, a1a2 ...aiai+1…an#)
2. 假定当前分析栈的栈顶为状态Sm,下一个输入符号为ai ,分析器的下一个动作

• 如果action[Sm, ai ]=移进且S=GOTO[Sm,ai ],则分析器执行移进, 三元组变成 (S0S1S2…SmS, #X1X2…Xmai , ai+1…an#)即分析器将输入符号ai和状态S移进栈, ai+1变成下一个符号；

• 如果action[Sm, ai ]=归约A→**β**,则分析器执行归约, 三元组变成 (S0S1S2…Sm-rS, #X1X2…Xm-rA, aiai+1 …an#), 此处S= GOTO[Sm-r , A], r为**β**的长度且**β**=Xm-r+1Xm-r+2…Xm；

• 若action[Sm,ai ]= acc,则接收输入符号串,语法分析完成；

• 若action[Sm,ai ]= err,则发现语法错误,调用错误恢复（直接返回）子程序进行处理。

1. **输出分析结果（UI）**

输出First集、项目集规范族、分析表、分析过程。

1. LR1分析表：

**（1）形如**：



1. **工作原理**

·action[Sm,ai ]表示下一个输入符号为ai,栈顶状态为Sm时,分析算法应执行的动作;

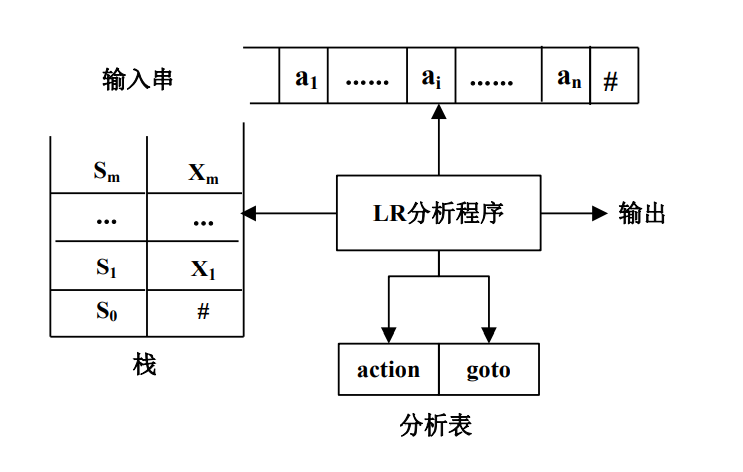
Ⅰ.若action[sm,ai ]=si ,表示移进, 然后栈顶状态为si ;

Ⅱ.若action[sm,ai ]=rj表示使用产生式(j)归约;

Ⅲ.若action[sm,ai ]=acc表示接受输入串;

Ⅳ.若action[sm,ai ]=err表示语法错误. 状态转换表:

·GOTO[Si ,X]表示归约出非终结符号X之后,当前栈顶状态为Si时,分析栈应转换到的下一个状态,即栈顶的新状态.



1. 具体实现
2. 数据结构

Ⅰ. 栈: 分析过程的符号栈，symbal\_s

Ⅱ.“表”: Action、Goto、组合成的分析表Analysis\_Table、分析结果Analysis\_result。

可写一个Table类

参考[(60条消息) C++表格类\_Att1cusss的博客-CSDN博客\_c++ 表格](https://blog.csdn.net/qq_43697906/article/details/115376552)

Ⅲ. 集合：文法集syntax、FIRST集、项目集project、项目集规范族proj\_fam。

1. 函数

Ⅰ. Get\_FIRST(): 根据FIRST集算法

Ⅱ. Closure(): 根据CLOSURE闭包，其中调用Goto（），输入（I1、E），返回I2。

Ⅲ. Get\_Analysis\_Table(): 根据I和GO函数构造分析表

Ⅳ. Analyse(): 根据分析表分析输入串，保存分析过程（每一步由序号、栈符号状态、输入串状态和采取的动作组成。）

Ⅴ. Print\_All(): 输出所有结果。

1. 输出格式说明：
2. First集
3. 项目集规范族
4. 所构建的LR（1）分析表
5. 针对输入串的分析过程
6. 源程序编译步骤：