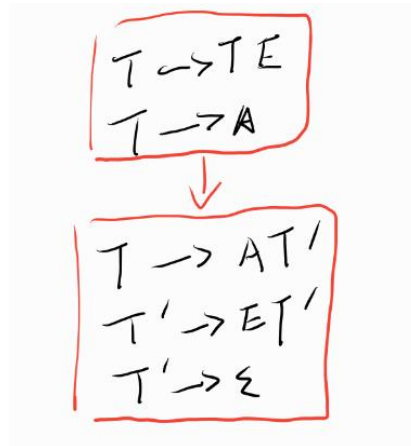


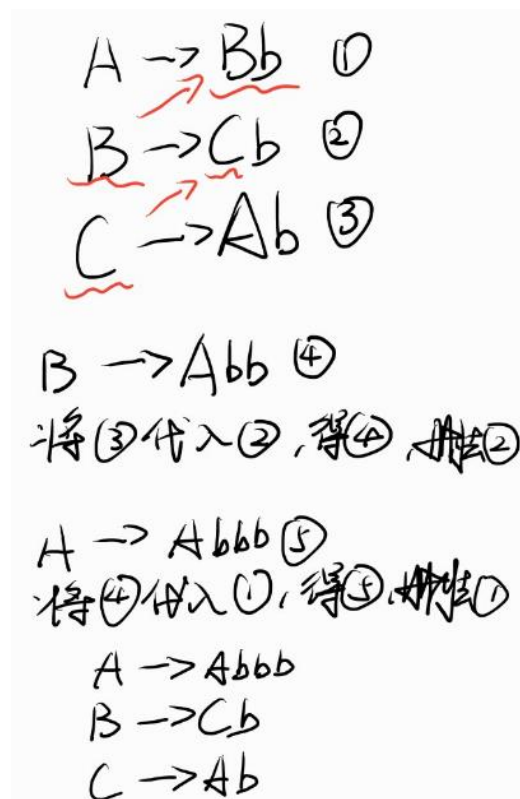
## 语法分析——LL1

为了完成文法分析，我们需要分别进行下面一些步骤：首先要消除左递归，然后分别求解每一个非终结符的 First 集和 Follow 集，最后构建预测表。得到这张表后我们就能够判断字符串是否规定的语法。

首先消除左递归。左递归分为两种，一种是直接左递归，另一种是间接左递归。对于直接左递归我们只需要进行如下的替换就可以了：



而对于间接左递归，我们的思路是先把它转换成直接左递归然后按上面的方法做处理，转换的方法如下：



在实现的时候可以先进进行间接左递归的转换，然后一并消除所有的直接左递归。

转换完成后，我们可以开始求解每一个非终结符的 **First** 集了。首先扫描文法集合，对于每一个文法，只可能出现如下情况：

(1)  $A \rightarrow aD$ ：如果右边第一个字符如  $a$  是终结符，直接将  $a$  加入 **First** ( $A$ )，前提是  $a$  不在 **First** ( $A$ ) 中。

(2)  $A \rightarrow Bd$ ：如果右边第一个字符如  $B$  不是终结符，就将 **First** ( $B$ ) 放到 **First** ( $A$ ) 中，前提是 **First** ( $A$ ) 中元素不能重复。

每扫描完整个文法集合后判断和上次扫描相比，集合是否增大，如果有则再一次进行扫描，直到集合不在扩大为止。

求解完 **First** 集后才能求解 **Follow** 集，因为在求解 **Follow** 集的过程中会使用到 **First** 集。同样，扫描文法集合，只可能出现如下情况：

(1)  $B \rightarrow aA$ ：如果  $A$  右边没有字符，那么将 **Follow** ( $B$ ) 加到 **Follow** ( $A$ ) 中。

(2)  $B \rightarrow aAX$ ：如果  $X$  是终结符，直接将  $X$  放到 **Follow** ( $A$ ) 中。如果  $X$  是非终结符，那么又会有两种情况：

(a) 将除  $\epsilon$  的 **First** ( $X$ ) 放到 **Follow** ( $A$ ) 中。

(b) 如果  $\epsilon$  在 **First** ( $X$ ) 中，那么将 **Follow** ( $X$ ) 加到 **Follow** ( $A$ ) 中。

每扫描完整个文法集合后判断和上次扫描相比，集合是否增大，如果有则再一次进行扫描，直到集合不在扩大为止。

最后我们根据上述集合就可以得到预测表了。同样首先扫描文法集合，对于每一个文法，我们观察它的右部的开头，如果是终结符我们直接把它写在以预测表的行标为文法左部，列标为这个终结符的位置；如果是非终结符，我们将这条文法写在以预测表的行标为文法左部，这个非终结符的 **First** 集的每一个元素为列标的位置；如果是空字符，那么就将这条文法写在以预测表的行标为文法左部，这个左部 **Follow** 集的每一个元素为列标的位置。需要注意的是列标中是不能出现空字符的，所以如果碰到，忽略即可。

有了这张表后我们就能够按照下面的步骤来判断字符串是否符合语法规则：首先给字符串末尾加上“#”。“#”和开始符号  $S$  进栈，字符串下标为 0 的位置开始。遍历栈，会有下面几种情况：

(1) 如果栈顶元素和字符串当前下标所指字符相同则进行弹栈操作，下标前进。

(2) 查表，行标是栈顶元素，列标是字符串下标字符，如果找不到则不匹配，找到则将栈顶弹出，依次压入表项右部元素。

当字符串扫描到#符号时如果栈中只有#符号时则表示匹配，字符串符合语法规则，否则语法不匹配。

程序运行效果如下（运行环境 .NET 4.5 VS2013）:

```
C:\WINDOWS\system32\cmd.exe
S->a
S->(T)
T->T, S
T->S

S:      a:  S->a          (:  S->(T)
T':      ,:  T'->,ST'      ):  T'->ε
T:      a:  T->aT'         (:  T->(T)T'

(a, a):True
(a, a, a):True
(a):True
a:True
请按任意键继续. . .
```