getFirst

1.所有终结符的First集为其本身，对于所有的非终结符，如果还未生成first集则执行第二步

2.对于非终结符α，First（α)=∅，对其所有产生式，如α=x1x2...xn，如果x1~xn是终结符或者已经生成first集，则执行第三步，否则对于其中所有未生成first集的非终结符执行第二步

3.

$$First[\alpha]=\begin{cases}\bigcup_{j=1}^{n}First[x_j] & ,First[x1]First[x2]\dots First[xn]\text{均包含}\varepsilon\\ \bigcup_{j=1}^{i-1}First[x_j]\bigcup First[x_i]-\varepsilon & ,First[x_1]First[x_2]\dots First[x_{i-1}]\text{均包含}\varepsilon\text{且}\varepsilon\notin First[x_i]\end{cases}$$

getFollow

实现方法:

1.初始化，对于开始符S（即Program），Follow[S]={#}，其它非终结符Follow集为空集

2.对于所有的产生式，例如α=x1x2...xn，执行第三步

3.对于x1~xn,依次对其中的非终结符xi执行第四步

4.如果xi等于xn，则

$$Follow[x_i]=Follow[x_i]\bigcup Follow[\alpha]$$

否则，对于xi+1~xn-1，执行第五步

5.

对于xj(i<j<=n)

$$Follow[x_i]=\begin{cases}Follow[x_i]\bigcup x_j & ,x_j\text{是终结符}\\ Follow[x_i]\bigcup First[x_j]-\varepsilon & ,\varepsilon\in First[x_j]\\ Follow[x_i]\bigcup First[x_j] & ,\varepsilon\notin First[x_j]\end{cases}$$

如果xj(i<j<=n)是非终结符，且xj的First集不包含ε，则并直接返回第三步

6.当处理完一遍所有的产生式后，如果有Follow集产生了变化，则重新开始执行第二步。

getPredict

1.对于所有的产生式，如第k个产生式α=x1x2...xn，执行第二步

2.如果x1~xn的First集均包含ε,则

$$Predict[k]=\bigcup_{j=1}^{n}First[j]\bigcup Follow[\alpha]-\varepsilon$$

3.如果x1~xi-1的First集均包含ε，xi的First集不包含ε,则

$$Predict[k]=\bigcup_{j=1}^{i}First[j]-\varepsilon$$

LL（1）

1.将'#'和开始符S(Program)依次入栈，按照Predict集生成LL(1)分析表

$$T\ （A,t)=\begin{cases}A->\alpha,\text{当}t\in Predict[A->\alpha]\\-1,\text{否则}\end{cases}$$

2.如果栈为空，或者输入流处理完毕执行第三步。否则从栈中弹出一个元素，

如果是终结符，判断是否匹配当前输入流字符，不匹配则进行错误处理，匹配则后移输入流，继续执行第二步。

如果是非终极符，判断当前输入流字符是否在该非终极符的LL(1)分析表中，如果在则将对应的产生式如α=x1x2...xn，将

x1x2...xn，倒着依次入栈，如果不在则进行错误处理

3.如果栈非空或者输入流未处理完，语法存在错误，否则语法分析成功。