# **Left Recursion Removal**

**Code:**

import java.util.\*;

class Leftrf

{

public static void main (String args[]) {

char non\_terminal[]=new char[10];

int len=0;

String beta1=new String();

String alpha1=new String();

String production []= new String [20];

int index=3;

Scanner s= new Scanner(System.in);

System.out.println("Enter number of non terminals:");

int n=s.nextInt();

System.out.println("Enter all the non terminals:");

for(int i=0;i<n;i++)

{

System.out.println("Enter a non terminal:");

non\_terminal[i]=s.next().charAt(0);

System.out.println("Enter the grammar for "+non\_terminal[i]+" :");

production[i]=s.next();

}

for(int i=0;i<n;i++)

{

for(int j=0;j<=i;j++)

{

String tp="";

String tp1="";

if(production[i].charAt(index)==non\_terminal[j] && i!=j)

{

int z=index;

int z1=production[i].length()-index-1;

while(production[i].charAt(z)!='|' && z1!=0)

{tp=tp+production[i].charAt(z);

if(production[i].charAt(z)!=non\_terminal[j])

tp1=tp1+production[i].charAt(z);

z++;

}

String temp[]=production[j].split("[>\\|]");

String tt="";

for(int k=1;k<temp.length;k++)

{

tt=tt+temp[k];

tt=tt+"\*\*";

if((k+1)!=temp.length)

tt=tt+"|";

}

//System.out.println("tt "+tt);

production[i]=production[i].replace(tp,tt.replace("\*\*",tp1));

}

}}

for(int i=0;i<n;i++)

{

index=3;

alpha1="";

beta1="";

if(non\_terminal[i]==production[i].charAt(index))

{

System.out.println("Grammar is left recursive.\n");

while(true)

{

len=production[i].length()-index-1;

while(production[i].charAt(index)!=0 && production[i].charAt(index)!='|' && len!=0)

{

if(production[i].charAt(index+1)!='|')

alpha1=alpha1+production[i].charAt(index+1);

index++;

len--;

}

alpha1=alpha1+non\_terminal[i]+"'"+'|';

index++;

if(non\_terminal[i]!=production[i].charAt(index))

break;

}

if(production[i].charAt(index)!=0) {

len=production[i].length()-index-1;

while(len!=0)

{

while(Character.isLetter(production[i].charAt(index))&&len!=0)

{

beta1=beta1+production[i].charAt(index);

index++;

len--;

}

if(production[i].charAt(index)!='|')

beta1=beta1+production[i].charAt(index);

beta1=beta1+non\_terminal[i]+"'";

if(production[i].charAt(index)=='|')

{

beta1=beta1+"|";

len--;

}

index++;

}

System.out.println("Grammar without left recursion:\n");

System.out.println(non\_terminal[i]+"->"+beta1);

System.out.println(non\_terminal[i]+"'"+"->"+alpha1+"eps");

}

else

System.out.println("Grammar can't be reduced\n");

}

else

System.out.println("Grammar is not left recursive.\n");

}

}

}

**Output:**

