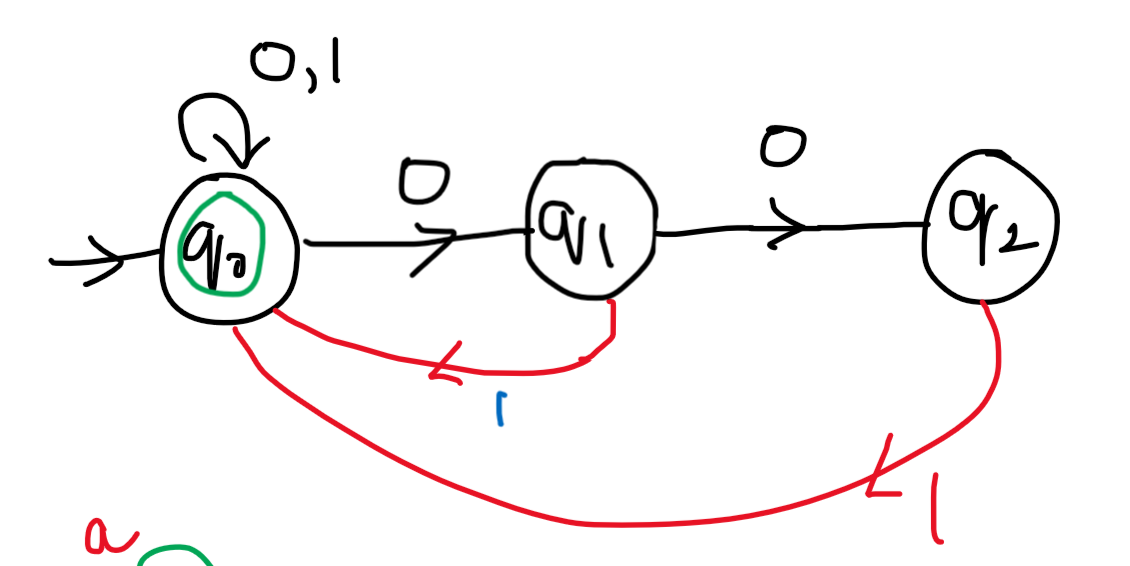
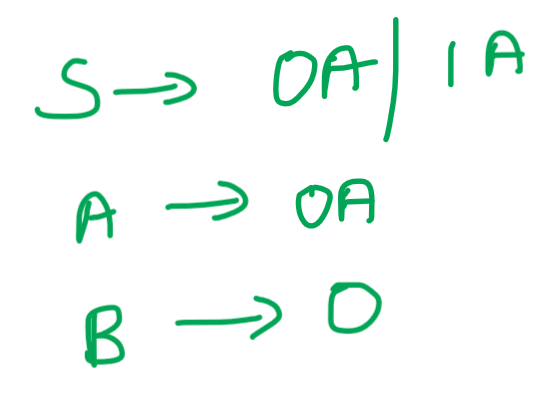
**Regular language to Regular grammar**

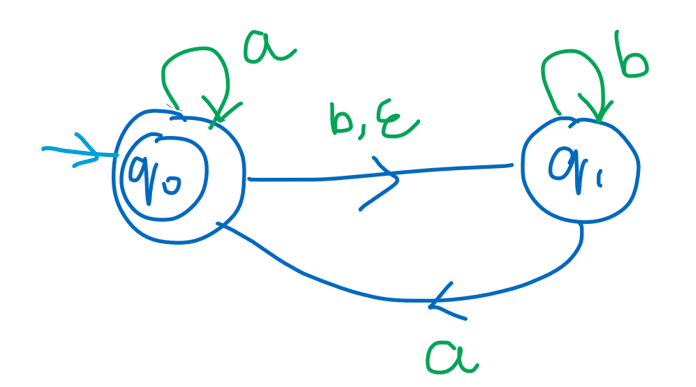
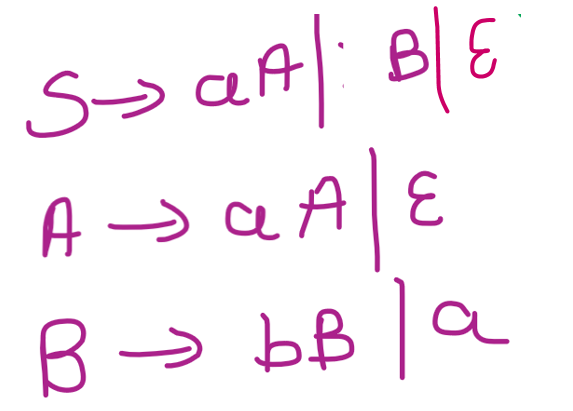
**1)L={w | w ends with 00}**



**2)L={0\*,1\*,0\*} with 3 states**

**3)L={a^n union b^na | n>=0}**



**\* L={ (a^n | n>=1) union (b^ma^k | m,k>=0) }**

**\* L(G) = { ancan / n ≥ 0 }**

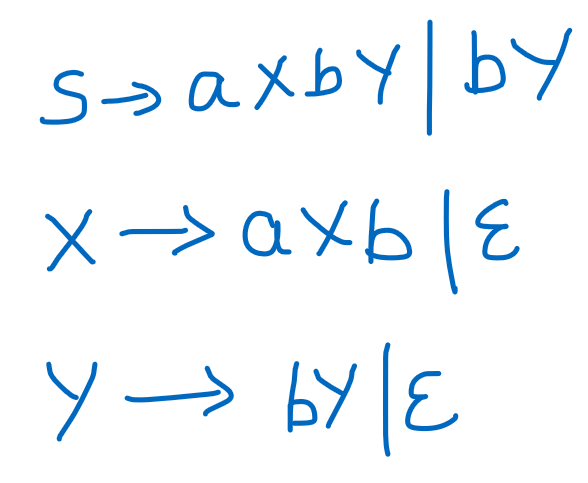
**\* L(G) = { anb2n / n ≥ 0 }**

**\* L(G) = { an+2bn / n ≥ 1 }**

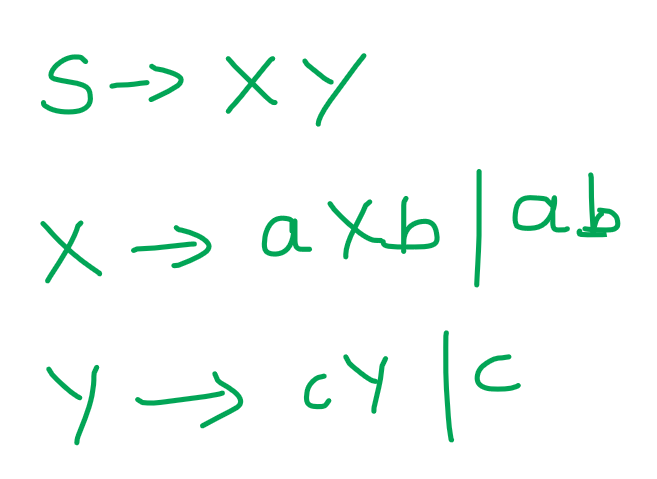
**\* L(G) = { anbn-3 / n ≥ 3 }**

**\* L(G) = { anbm / n ≥ 0, m ˃ n }**

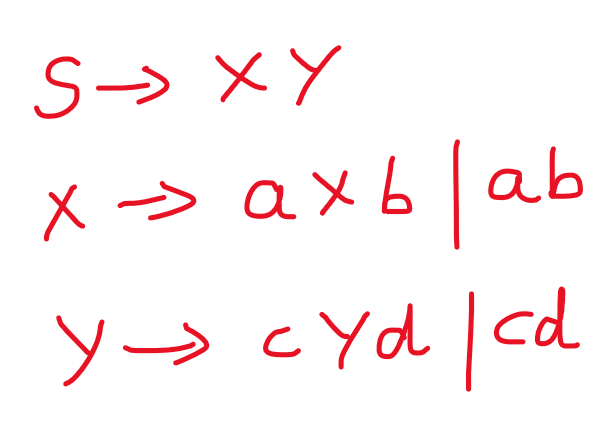
**The grammar can gone any way, but wrong string should not pass through that grammar.**



**\* L(G) = { anbn cm  / n, m ≥ 1 }**



**\* L(G) = { anbn cm dm  / n, m ≥ 1 }**



**\* L(G) = { anbm cm dn  / n, m ≥ 1 }**

**\* L(G) = { anbm / n, m ≥ 1 m ≠ n}**

**\* L(G) = { w {a, b}\* / na(w) = nb(w) + 1 }**

1)L = { a^n b^m | n,m>=1 }

2) **L = { a^n b^n c^m | n,m>=1 }** S 🡪 XY  
 X 🡪 aXb | ab  
 Y 🡪 cY | c

3) **L = { a^n c^m b^n | n,m>=1 }** S 🡪 aXb  
 X 🡪 aXb | CX | c

4) **L = { a^n b^m a^2n | n,m>=0 }** S 🡪 aXaa |epsilon  
 X 🡪 aXaa |bX | epsilon

5) **E = {a,b}**  
**All non empty strings start and ends with the same symbol (or)   
Palindrome**   
S 🡪 aAa | bAb | a |b | epsilon  
A 🡪 aA | bA | epsilon

6) **L = { w (0,1)\* | w^R and |w| is even }**  
S 🡪 epsilon | 1S1 | 0S0

7) **L = { w (0,1)\* | w contains at-least 3 ones }**S 🡪 A1|A1|A1  
A 🡪 epsilon | 0A | 1A



8) **L = { a b c | i,j,k>=0 and i+j = k }**  
S 🡪 aSc | X  
X 🡪 bXc | epsilon



9) **L = { a b c | i,j,k>=0 and i=j }**S 🡪 aXbcC  
X 🡪 aXb | epsilon  
C 🡪 cC | epsilon



10)   
S 🡪 aXc  
X 🡪 bX | aXc | epsilon

11) L = { a b c | i<j }

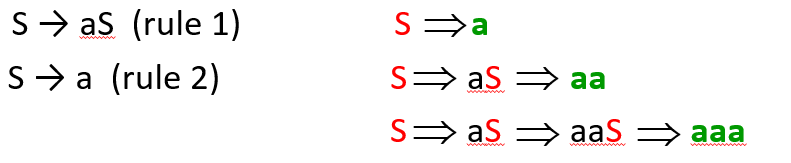


12) L = { a b c | i<k }

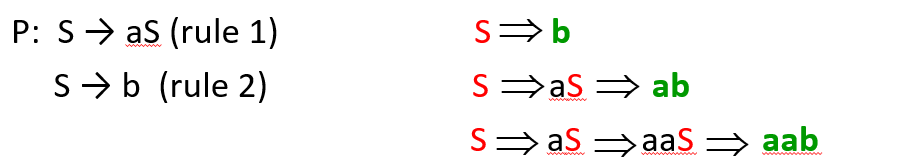


**Regular grammar to Regular language**

**1)**

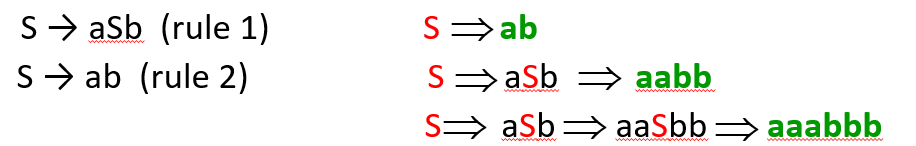


**2)**

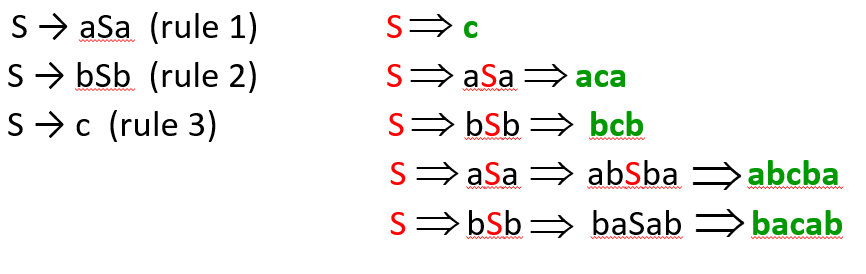




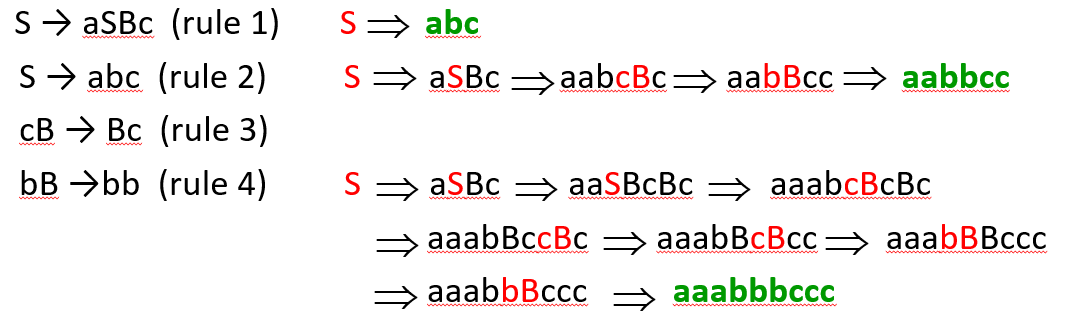
**3)**



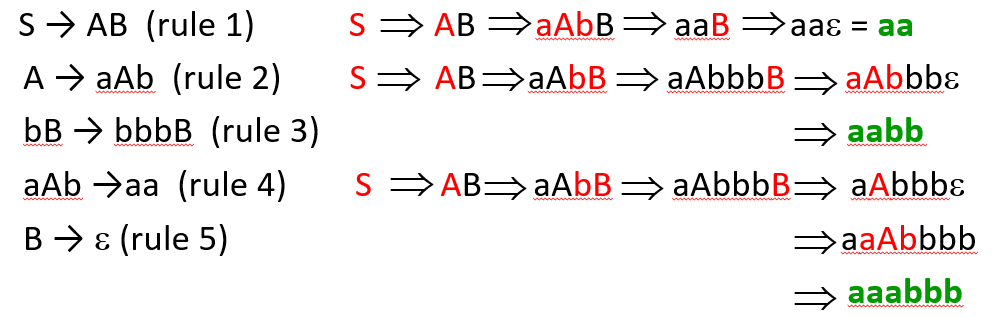
**4)**



**5)**



**6)**



**7)**

