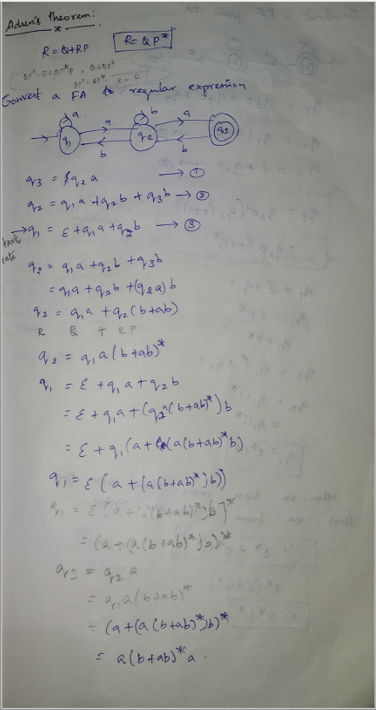
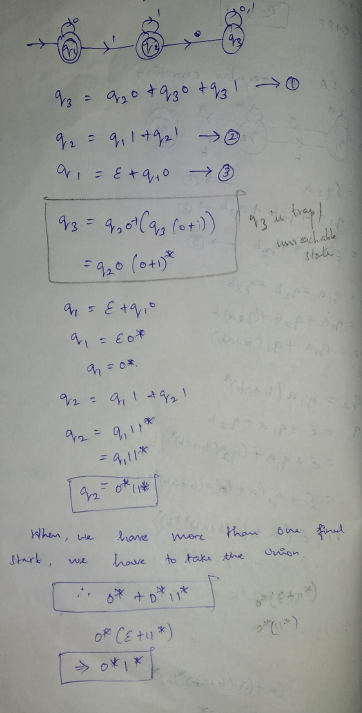
1)

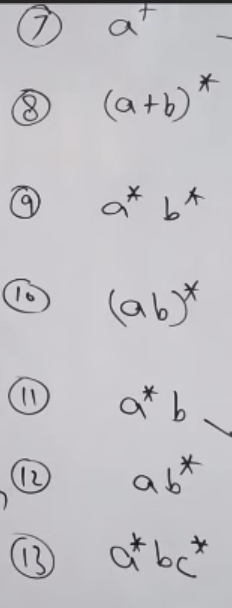
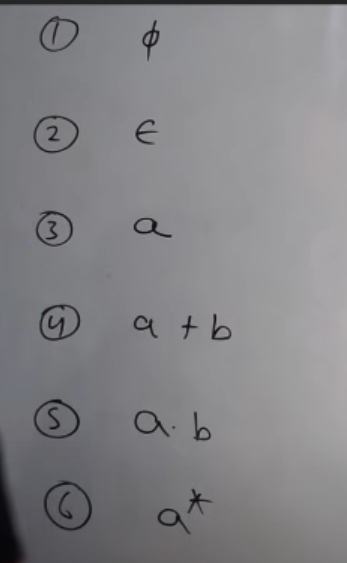


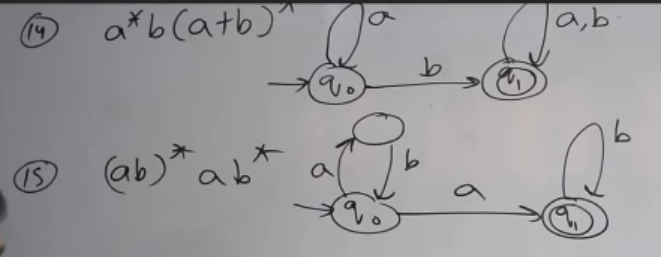
2)



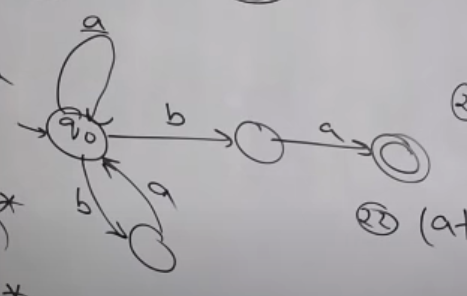
Conversion of Regular expression to finite automata

1) 0\*10\* L={w|w contains a single 1}   
2) (0|1)\* 1 (0|1)\* L = {w|w contains at-least a single 1}  
3) (01) U (10)   
4) (0|1 0|1 0|1)\*  
5) 0(0|1)\*0 + 1(0|1)\*1 + 0 + 1





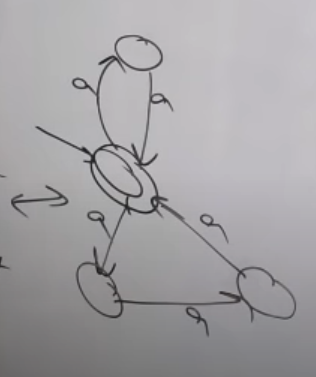
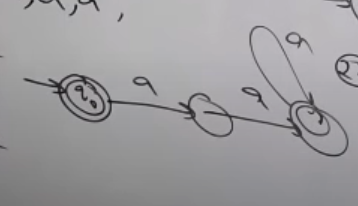
16) (a+ba)\*ba





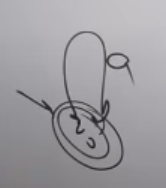
17) (aa+aaa)\*



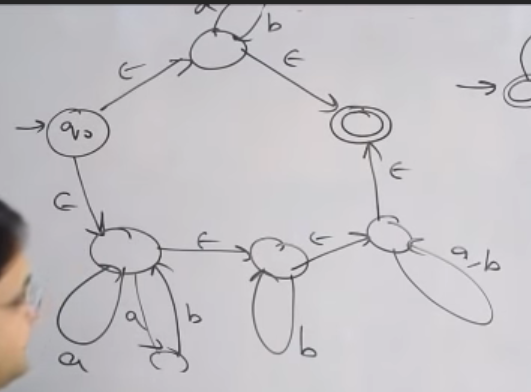
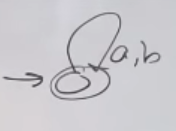
(or



18) (a+aaaa)\*

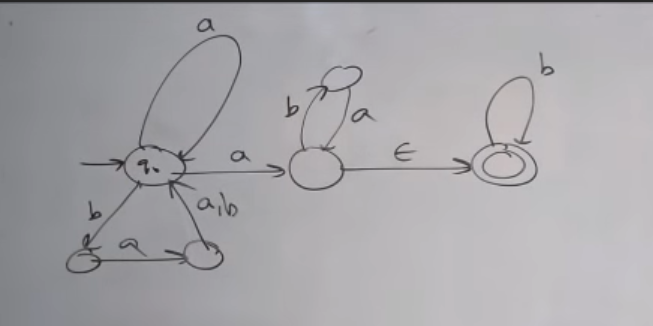


19) (ab)\* + (a+ab)\*b\*(a+b)\*

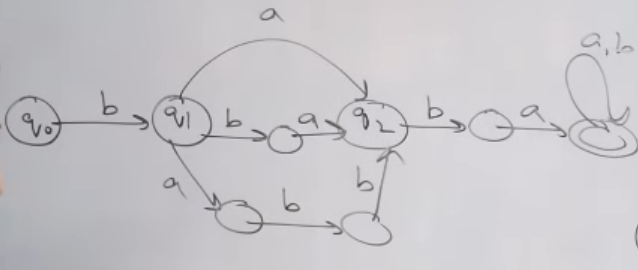
 



20) [a+ba(a+b)]\* a(ba)\* b\*



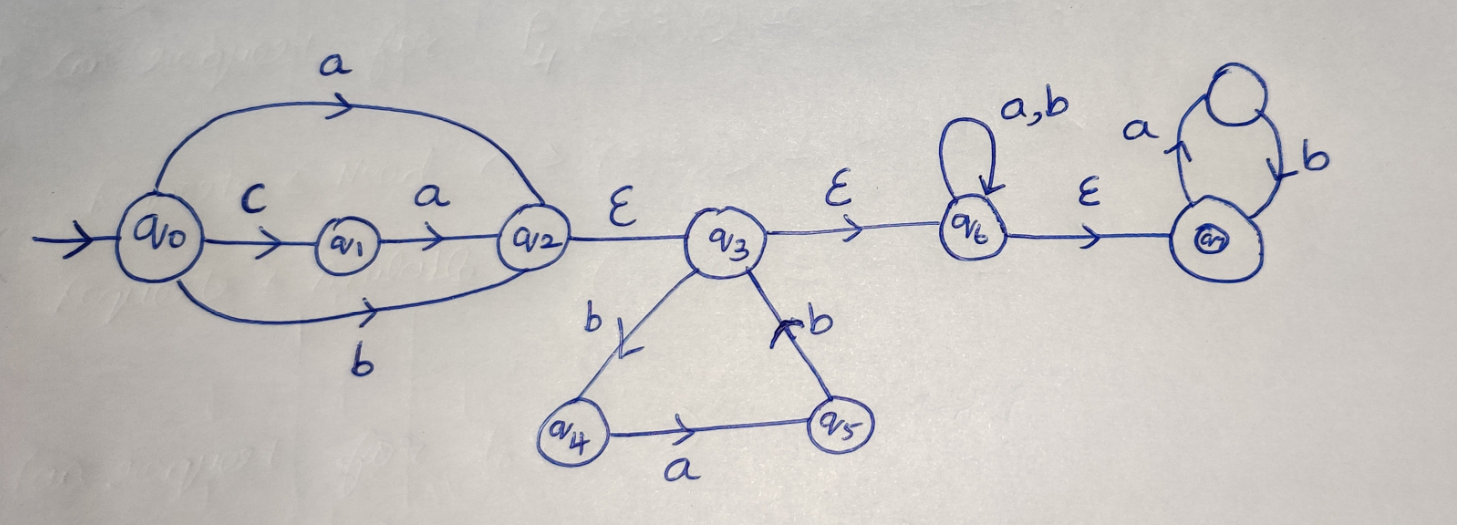
21) b(a+ba+abb) (ba(a+b)\*)



22) (ab U a)\*

23) (a U b)\* aba

24) (a+b+ca)[ (bab)\* + (a+b)\* ]\* (ab)\*



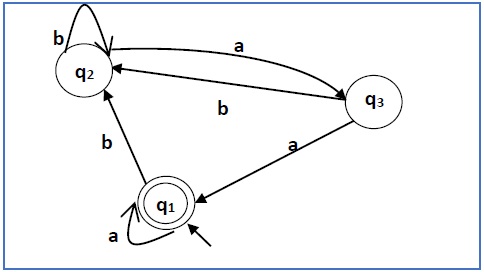
If the input alphabet is then the regular expression   
 🡪 describes the language consisting of all strings of length 1 over this alphabets

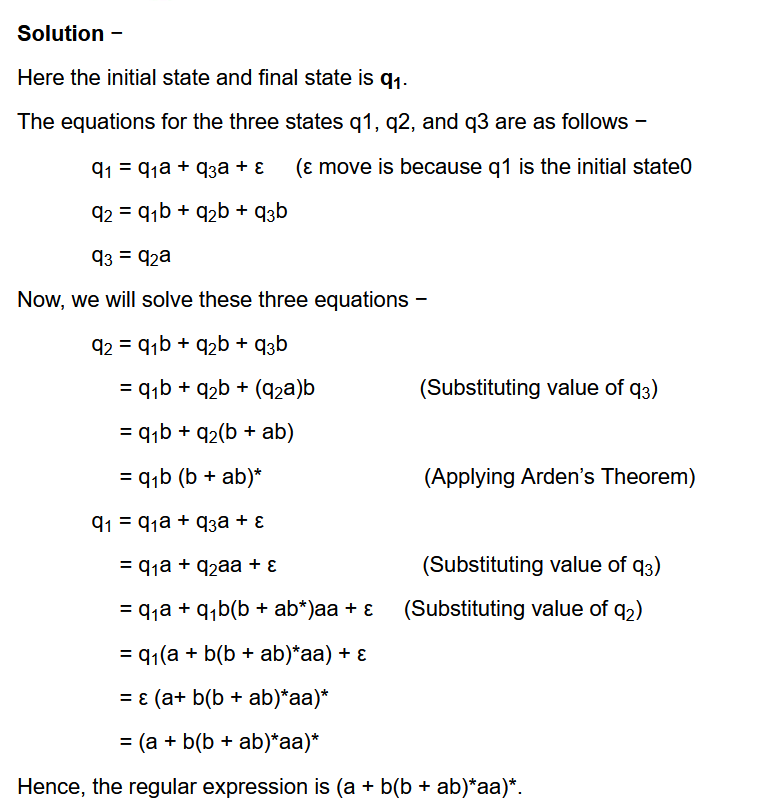
\* 🡪 describes the language that contains all strings

\* 1 🡪 describes the language that contains all the strings ending with 1

**Adren’s theorem**

1)





2)

