```
COMPUTE- PREFIX- FUNCTION (P)
1. m length [P]  //'p' pattern to be matched
2.  [1]  0
3. k  0
4. for q  2 to m
5. do while k > 0 and P [k + 1]  P [q]
6. do k   [k]
7. If P [k + 1] = P [q]
8. then k k + 1
9.  [q]  k
```

String matching with Finite automata

complexity ⇒ O(n)

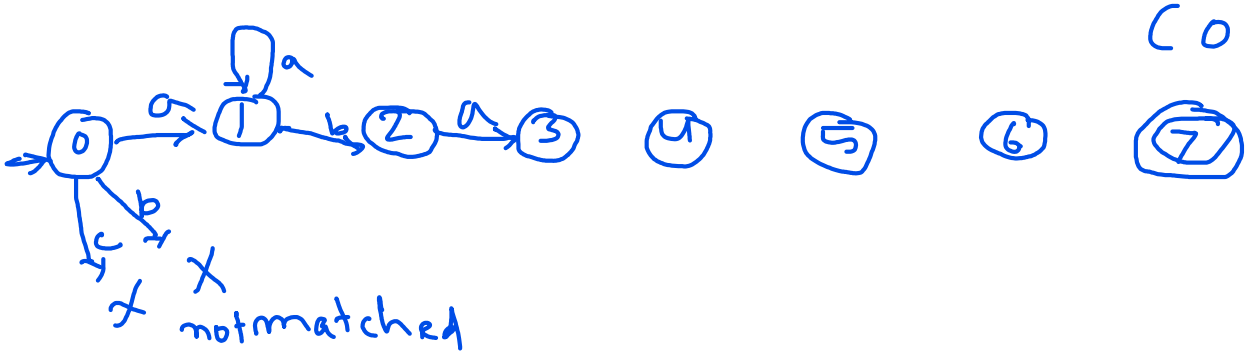
P ⇒ a b a b a c a

Text ⇒ a b ab a b a c a b a

Step 1 – make FA for Pattern string

No. of char ⇒ 7 ⇒ 7 state

(0 to 7

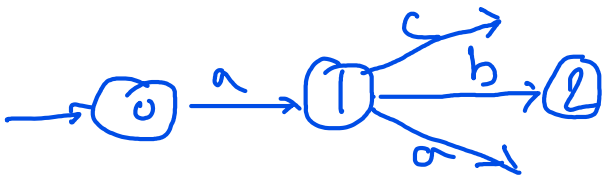


$\Sigma = \{a, b, c\}$

check all these char with Patter char if matched change the state

and also apply prefix & suffix Rule

at state (1) check all char a, b, c



possible string.

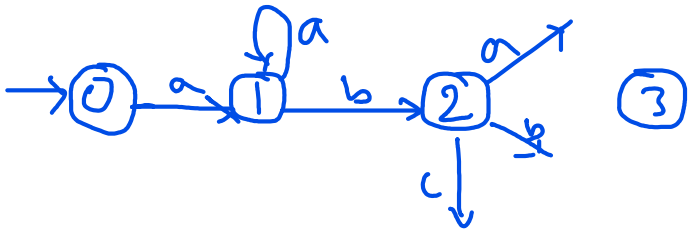
a a , a b , a c

Prefix suffix — matched with 1st pattern char, so make self loop at a

matched with 2nd char so change the state at b

all not matched

at state (2)



a b a , a b b , a b c

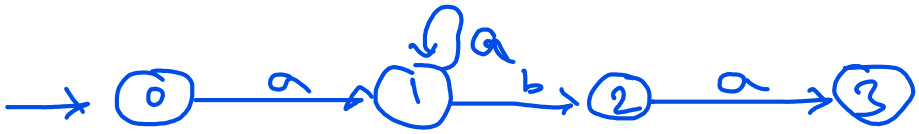
Prefix a, ab, aba
Suffix a, ba, aba

a, ab, abb
b, bb, abb

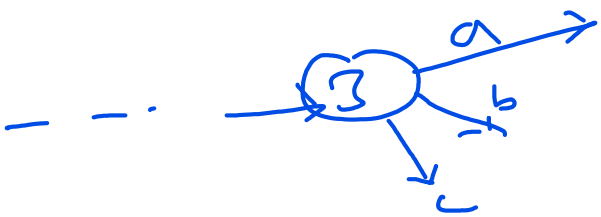
a, ab, abc
c, bc, abc

Longest matched prefix & suffix with pattern string is a b a , this is at char a so change the state at a

So it become



at state 3



strings are

a b a a , a b a b , a b a c ✗ not match

matched pattern string - matched so change state

a b a a

x

x

1 char matched so