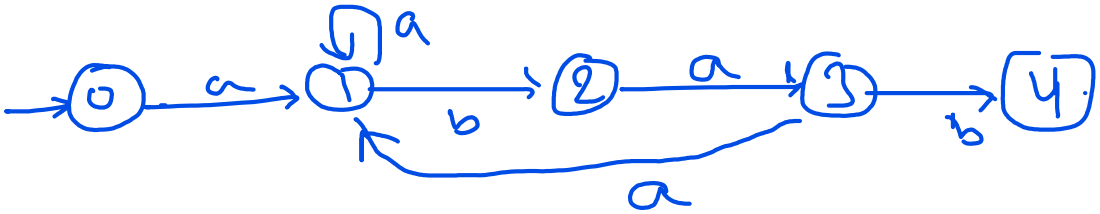


P ⇒ a b a b a c a

Strings: a b a b — matched at b — change state

(1)

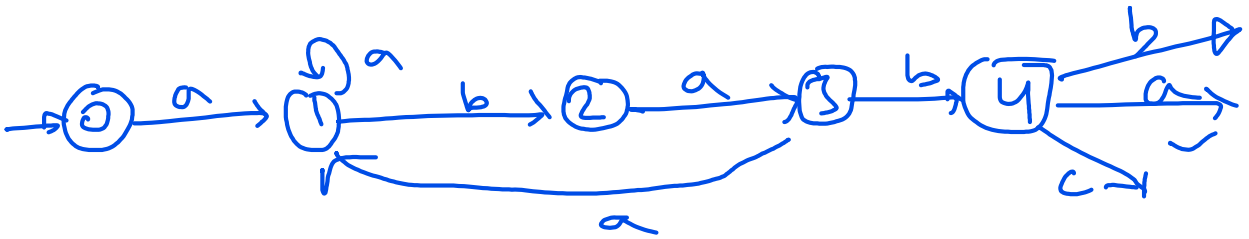
(2) a b a c x — x — not matched at c

X

(3) a b a a — x — matched one length a

X

at state (4)



(1) a b a b a matched at a

so change state to (5)

(2) a b a b b

x

x

x

x

not matched at b

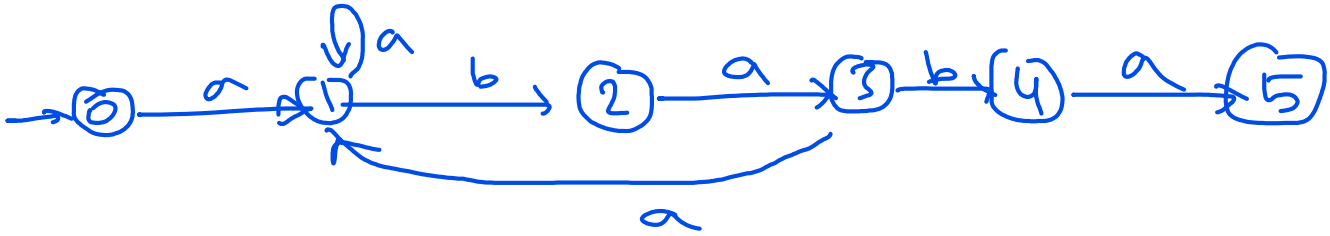
(3) a b a b c

x

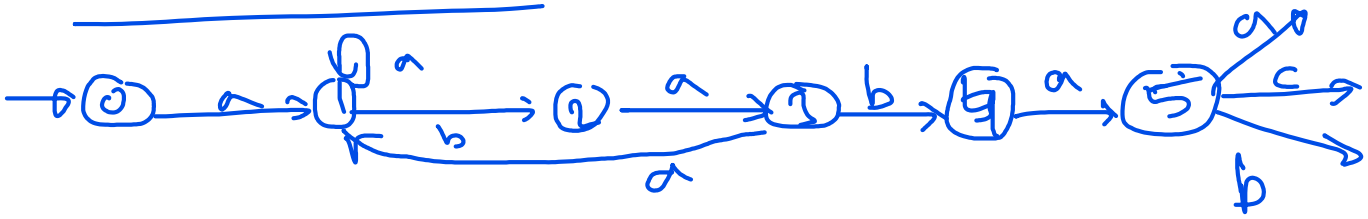
x

x

x



at state (5)



(1)

a b a b a a

x
x
x

1 length char at a is matched so move (5) to (1)

(1)

a b a b a b

x
x
x

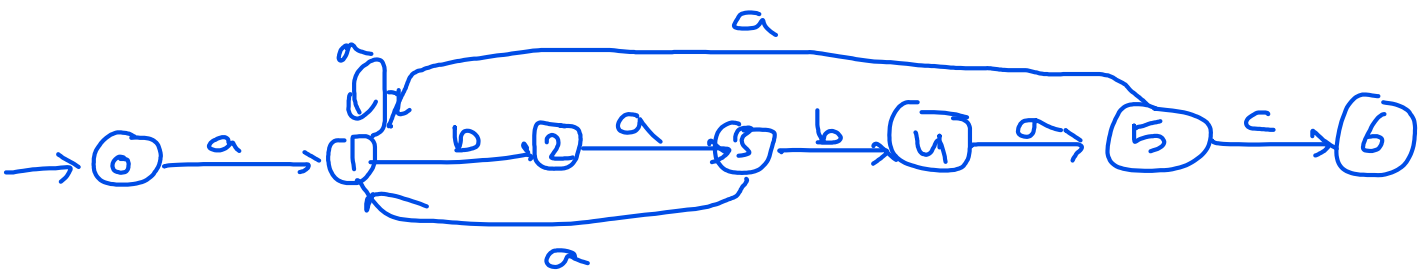
4 length char matched so move (5) to (4) at b

(3)

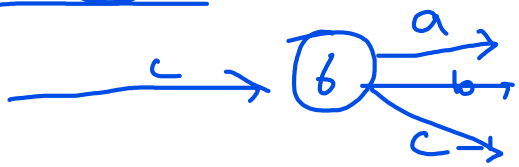
a b a b a c

x
x
x
x
x

matched so change state (5) to (6)



at state (6)

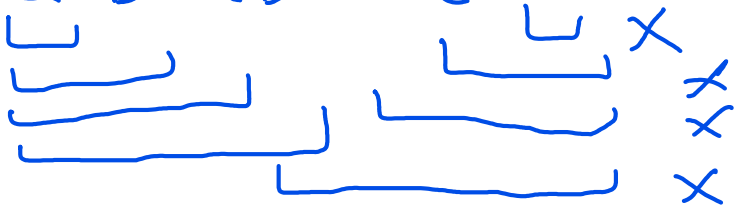


String –

(1) a b a b a c a

matched in Pattern string so state change (6) to (7)

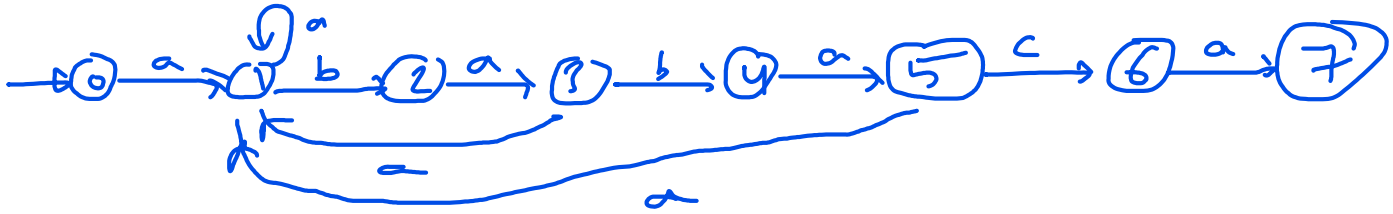
(2) a b a b a c b



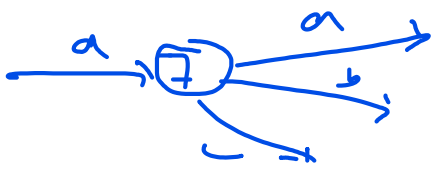
not matched at b

(3) a b a b a c c

not " at c



at state (7)



string - a b a b a c a a

(1)

1 length char matched at a

so state (7) to (1)

(2) a b a b a c a b

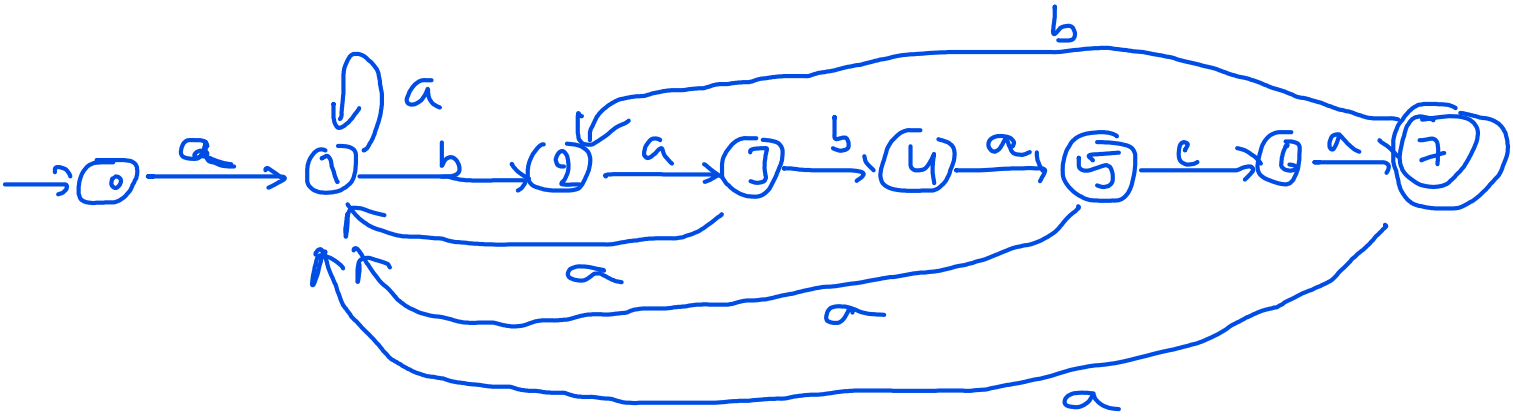
x x

2 length char matched at b

(7) to (2)

(3) a b a b a c a c

not matched



Transition table —

	a	b	c
0	1	0	0
1	1	2	0
2	3	0	0
3	1	4	0
4	5	0	0
5	1	4	6
6	7	0	0
7	1	2	0

FA(T, δ, m)

1. n ← length[T]
2. q ← 0
3. for i ← 1 to n
4. do q ← δ[q, T[i])
5. if q == m
6. Pattern occured with shift i − m

P → a b a b a c a

T → a b a b a b a c a b a

i → 1 2 3 4 5 6 7 8 9 10 11

initial state $q \Rightarrow 0$ ✓

$q \leftarrow \delta(0, T[1])$
$= \delta(0, a)$
$= 1$ ✓

$q \leftarrow \delta(1, T[2])$
$= \delta(1, b)$
$= 2$ ✓

$q \leftarrow \delta(2, T[3])$
$= \delta(2, a)$
$= 3$ ✓

$q \leftarrow \delta(3, T[4])$
$\leftarrow \delta(3, b)$
$= 4$

$q \leftarrow \delta(4, a)$
$= 5$

$q \leftarrow \delta(5, b)$
$= 4$

$q \leftarrow \delta(4, a)$
$= 5$

$q \leftarrow \delta(5, c)$
$= 6$

$q \leftarrow \delta(6, a)$
$= 7$

Now here we got
$q = m$.
$7 = 7$

so return $(i - m)$
$\Rightarrow 9 - 7 = 2$

after 2 shift matched the given pattern from index 3 to continue the procedure for matching remaining element

```
COMPUTE-TRANSITION-FUNCTION (P, )
 1. m  length [P]
 2. for q  0 to m
 3. do for each character a  *
 4. do k  min (m+1, q+2)
 5. repeat kk-1
 6. Until
 7. (q,a)k
```

```
FINITE- AUTOMATON-MATCHER (T,, m),
 1. n  length [T]
 2. q  0
 3. for i  1 to n
 4. do q   (q, T[i])
 5. If q =m
 6. then si-m
 7. print "Pattern occurs with shift s" s
```