

## Unit 2 : Nondeterminism and Kleene's Theorem

### \* Non deterministic finite automata :

NFA stands for Non-deterministic finite automata.

NFA is a state machine consisting of states and transition that can either accept or reject a finite string.

It is easy to construct NFA than DFA for given regular language.

The finite automata are called NFA when there exist many paths for specific input from the current state to the next state.

Every NFA is not DFA, but each NFA can be translated into DFA.

NFA also has five states.

Q : finite set of states

$\Sigma$  : finite set of input symbols

$q_0$  : initial state

F : final state

$\delta$  : Transition function.

$$\delta : Q \times \Sigma \rightarrow 2^Q$$

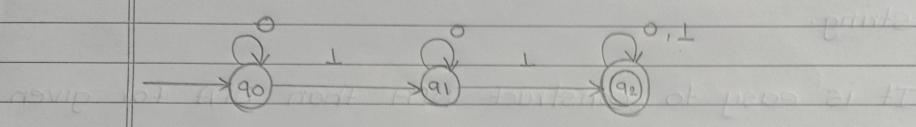
The power to be several states at once.

In deterministic automata only single power or single transition.

String ending with 01:

In deterministic finite automata (DFA):

① Only single transition function will be there

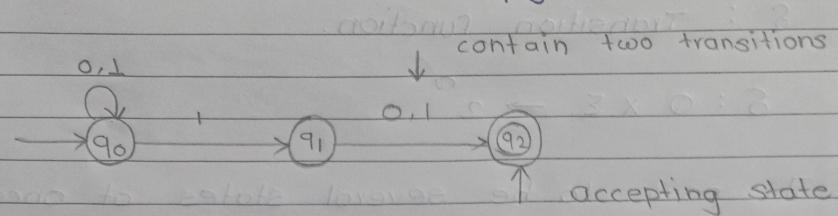


② Instead of accepting state only single transition

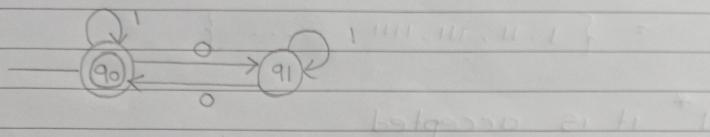
	0	1	0,1	None
0	q0	q0	q1	None
1	q1	q1	q2	None
0,1	q2	q2	q2	None

In non deterministic automata,

It will contain two transition, or two arcs at a single state and accepting state most of the time should be empty.



\* check weather following is NFA/DFA



String acceptance | Reject

- ① 1                      ⑤ 010011
- ② 101                  ⑥ 0\*1\*0, 0, 1 } = \*0
- ③ 1\*                    ⑦ 0\*
- ④ 11001                ⑧ 0001\*1\*0, 00, 0 } =

state diagram  $IP = (Q, \delta)$

$IP = (Q, \delta)$   $\leftarrow$  Q (states)

0    1  $\leftarrow$  alphabets

states	q0	q1	q0	$(Q, \delta)$
	q0	q1	q0	$(Q, \delta)$
	q1	q0	q1	$(Q, \delta)$

$\delta = (Q, \delta)$

$$\textcircled{1} \quad \delta(q_0, 0) = q_1 \quad \textcircled{1} \quad \delta(q_0, 1) = q_0$$

$$\textcircled{2} \quad \delta(q_0, 1) = q_0 \quad \textcircled{2} \quad \delta(q_1, 0) = q_0$$

$$\textcircled{3} \quad \delta(q_1, 0) = q_0 \quad \textcircled{3} \quad \delta(q_1, 1) = q_1$$

$$\textcircled{4} \quad \delta(q_1, 1) = q_1 \quad \textcircled{4} \quad \delta(q_0, 0) = q_0$$

Transitions

①  $1^*$                   A3 belongs to L \*

$L^* = \{ 1, 11, 111, 1111, \dots \}$     101

$$\textcircled{a} \quad 1 \Rightarrow \delta(q_0, 1) = q_0 \quad \textcircled{a} \quad \delta(q_0, 1) = (Q, \delta)$$

$$\textcircled{b} \quad 11 \Rightarrow \delta(q_0, 1) = q_0 \quad \textcircled{b} \quad \delta(q_0, 1) = (Q, \delta)$$

$$\delta(q_0, 1) = q_0 \quad \textcircled{c} \quad \delta(q_0, 1) = (Q, \delta)$$

bela... ton ei 101

$$A^* = \{ \text{I, II, III, IV, V} \}$$

$$= \{ 1, 11, 111, 1111, \dots \}$$

1\* it is accepted

② 0\*

$$0^* = \{ \text{N, O, OO, OOO, OOOO, \dots} \}$$

$$= \{ 0, 00, 000, 0000, \dots \}$$

a) 0  $\Rightarrow \delta(q_0, 0) = q_1$  ~~initial state~~

b) 00  $\Rightarrow \delta(q_0, 0) = q_1$

~~$\delta(q_1, 0) = q_0$~~  ✓

c) 000  $\Rightarrow \delta(q_0, 0) = q_1$  IP ~~initial state~~

$\delta(q_1, 0) = q_0$  oP IP

$\delta(q_0, 0) = q_1$  X

d) 0000  $\Rightarrow \delta(q_0, 0) = q_1$  P = (0, oP) 3 ①

~~$\delta(q_1, 0) = q_0$  = (1, oP) 3 ②~~

~~$\delta(q_0, 0) = q_1$  = (0, IP) 3 ③~~

~~$\delta(q_1, 0) = q_0$  = (1, IP) 3 ④~~

0\* it is accepted by FA \* 1 ①

③ 101 { , II, III, IV, V } = \*

$$\delta(q_0, 1) = q_0 \text{ oP} = (1, oP) 3 \leftarrow 1 \text{ (o}$$

$$\delta(q_0, 0) = q_1 \text{ oP} = (1, oP) 3 \leftarrow 1 \text{ (o}$$

$$\delta(q_1, 1) = q_1 \text{ oP} = (1, oP) 3$$

101 is not accepted.

④  $\{11001, 100, 110, 10, 0, 11, 11, 1\} = \{11000, 11000, 10000, 0000\}$

$$\delta(9_0, 1) = 9_0$$

$$\delta(9_0, 1) = 9_0 \quad \{11, 10, 00, 1, 0\} = 1^* 1^* 0$$

$$\delta(9_0, 0) = 9_1$$

$$\delta(9_1, 0) = 9_0 \quad 1P = (0, 0P) \beta \Leftarrow 0 \quad 0$$

$$\delta(9_0, 1) = 9_0 \quad 0P = (1, 0P) \beta \Leftarrow 1 \quad 0$$

$$1P = (0, 0P) \beta \Leftarrow 00 \quad 0$$

11001 is accepted  $P = (0, 1P) \beta$

$$1P = (0, 0P) \beta \Leftarrow 1000 \quad 0$$

⑤ 010011

$$0P = (0, 1P) \beta$$

$$1P = (0, 0P) \beta$$

$$\delta(9_0, 0) = 9_1 \quad 1P = (1, 0P) \beta$$

$$\delta(9_1, 1) = 9_1 \quad 1P = (0, 1P) \beta$$

$$\delta(9_1, 0) = 9_0$$

$$\delta(9_0, 0) = 9_1$$

$$\delta(9_1, 1) = 9_1$$

$$\delta(9_1, 1) = 9_1$$

010011 it is not accepted

⑥  $0^* 1^*$

$$0^* = \{ \text{1}, 0, 00, 000, \dots \}$$

$$1^* = \{ \text{1}, 1, 11, 111, \dots \}$$

$$0^* 1^* = \{ \text{1}, 11, 111, 1111, 01, 011, 0111, 001, 0011, 00111, 000111, 0001111, \dots \}$$

$$= \{ 1, 11, 111, 0, 01, 011, 0111, 00, 001, 0011, 00111, \\ 000, 0001, 00011, 000111, \dots \}$$

$$0^* 1^* = \{ 0, 1, 00, 01, 11, \dots \} \quad \text{IP} = (1 - \text{op})^2$$

$$a) 0 \Rightarrow \delta(q_0, 0) = q_1 \quad \text{op} = X(0, \text{IP})^2$$

$$b) 1 \Rightarrow \delta(q_0, 1) = q_0 \quad \text{op} = \checkmark(1, \text{op})^2$$

$$c) 00 \Rightarrow \delta(q_0, 0) = q_1 \quad \delta(q_1, 0) = q_0 \quad \text{IP} = (0, \text{op})^2$$

$$d) 0001 \Rightarrow \delta(q_0, 0) = q_1 \quad \delta(q_1, 0) = q_0 \quad \text{IP} = (0, \text{op})^2$$

$$\delta(q_0, 1) = q_1 \quad \text{IP} = (1, \text{op})^2$$

$$\delta(q_1, 1) = q_1 \quad \text{IP} = X(1, \text{op})^2$$

$$\delta(q_1, 0) = q_0 \quad \text{IP} = \checkmark(1, \text{op})^2$$

$$\delta(q_0, 0) = q_1 \quad \text{IP} = (0, \text{op})^2$$

$$\delta(q_1, 0) = q_1 \quad \text{IP} = (1, \text{op})^2$$

$$\delta(q_1, 1) = q_0 \quad \text{IP} = X(1, \text{op})^2$$

$$\delta(q_0, 1) = q_0 \quad \text{IP} = \checkmark(0, \text{op})^2$$

$$\delta(q_0, 0) = q_0 \quad \text{IP} = (0, \text{op})^2$$

$$\delta(q_1, 0) = q_1 \quad \text{IP} = (1, \text{op})^2$$

$$\delta(q_1, 1) = q_1 \quad \text{IP} = X(1, \text{op})^2$$

$$\delta(q_0, 1) = q_0 \quad \text{IP} = \checkmark(0, \text{op})^2$$

$$\delta(q_0, 0) = q_1 \quad \text{IP} = (0, \text{op})^2$$

$$\delta(q_1, 0) = q_1 \quad \text{IP} = (1, \text{op})^2$$

$$\delta(q_1, 1) = q_0 \quad \text{IP} = X(1, \text{op})^2$$

$$\delta(q_0, 1) = q_0 \quad \text{IP} = \checkmark(0, \text{op})^2$$

$$\delta(q_0, 0) = q_1 \quad \text{IP} = (0, \text{op})^2$$

$$\delta(q_1, 0) = q_1 \quad \text{IP} = (1, \text{op})^2$$

$$\delta(q_1, 1) = q_0 \quad \text{IP} = X(1, \text{op})^2$$

Draw the state diagram using following table & string accepted by FA 110001 and 101000

state \ IP	0	1
$\rightarrow q_0$	$q_2$	$q_1$
$q_1$	$q_3$	$q_0$
$q_2$	$q_0$	$q_3$
$q_3$	$q_1$	$q_2$

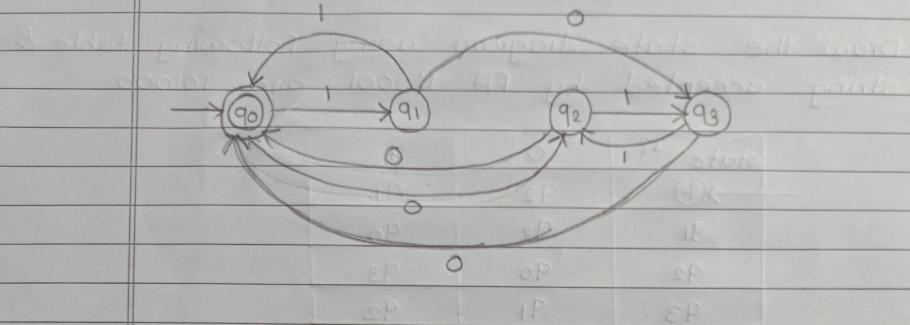
$$\begin{aligned}
 \delta(q_0, 110001) &= \delta(q_1, 10001) \\
 &= \delta(q_0, 0001) \\
 &= \delta(q_2, 001) \\
 &= \delta(q_0, 01) \\
 &= \delta(q_2, 1) \\
 &= q_3
 \end{aligned}$$

It is not accepted by FA.

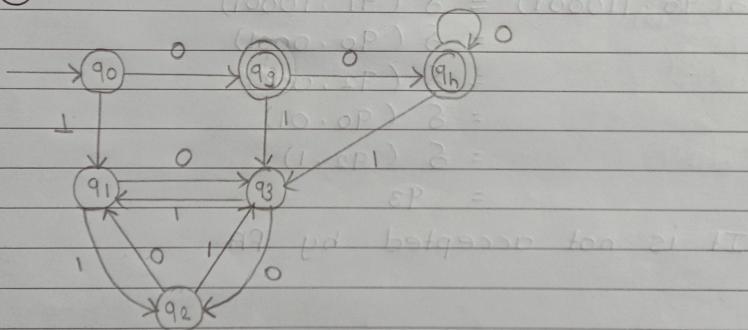
② 101000

$$\begin{aligned}
 \delta(q_0, 101000) &= \delta(q_1, 01000) \\
 &= \delta(q_3, 1000) \\
 &= \delta(q_2, 000) \\
 &= \delta(q_0, 00) \\
 &= \delta(q_2, 0) \\
 &= q_0
 \end{aligned}$$

It is accepted by FA



$$② (10001, 1P) \beta = (100011, 0P) \beta$$



$q_0 \Rightarrow \text{Initial state } \beta = (000101, 0P) \beta$

$$Q = \{q_0, q_1, q_2, q_3, q_g, q_h\}$$

$$\Sigma = \{0, 1\} \cup \{0P\} \beta =$$

$$F = \{q_g, q_h\} \cup \{0P\} \beta =$$

$$(0, 0P) \beta =$$

$$0P =$$

A7 पर लगाना है IT

Transition table

		0	1	0000, 00, 0 } = *0
→	q0	q3	q1	
	q1	q3	q2	
	q2	q1	q3	
	q3	q2	q1	
	(q0)	qh	q3	
	(qh)	qh	q3	

Ex: ① 10101

$$|S_1|=5$$

$$\delta(q_0, 10101)$$

$$\delta(q_1, 0101)$$

$$\delta(q_3, 101)$$

$$\delta(q_1, 01)$$

$$\delta(q_3, 1)$$

= q<sub>1</sub> Non final state

$$\textcircled{2} \quad S_2 = 0^* 11$$

$$0^* 11 = \{ \lambda 11, 011, 0011, 00011, \dots \}$$

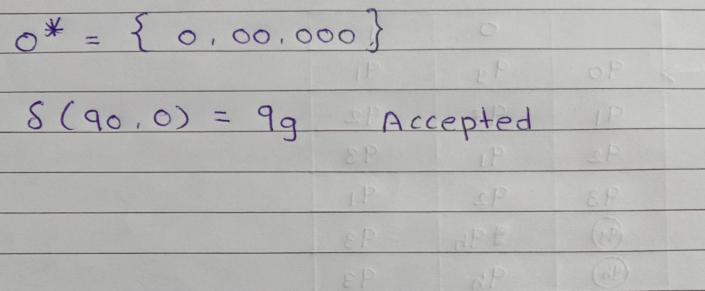
$$S_{12} = 11$$

$$\delta(q_0, 11)$$

$$\delta(q_1, 1)$$

= q<sub>2</sub> Not accepting

③  $s_3 = o^*$  state accepting



$\delta(q_0, 0) = q_1$  Accepted

10101 ① → x

2 → 1101

(10101, 0P) 2

(1010, 1P) 2

(101, 2P) 2

(10, 3P) 2

(1, 4P) 2

state final 0011 → 1P =

$$u^* o = o \circ \textcircled{1}$$

$$\{ \dots, 11000, 1100, 110, 11 \} = u^* o$$

11 = 010

(11, 0P) 2

(1, 1P) 2

partgama total 0P =

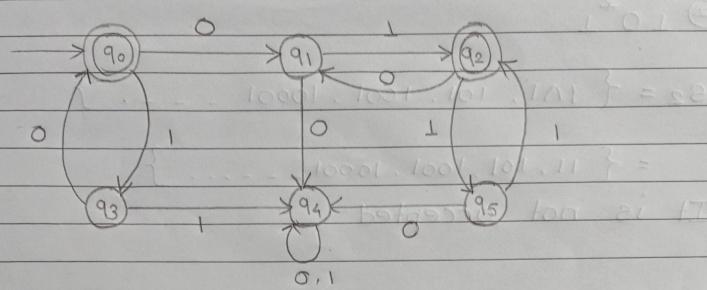


fig : state diagram

	0	1	
q0	q1	q3 = (1010, 0P) 3	
q1	q4	q2 = (1010, 0P) 3	
q2	q1	q5 = (101, 1P) 3	
q3	q0	q4 = (10, 1P) 3	
q4	q4	q4 = (1, 1P) 3	
q5	q4/2	q2 = (101, 0P) 3	

fig : transition table

① 1111

1S1 = 5

$$\begin{aligned}
 S(q_0, 1111) &= S(q_3, 1111) \\
 &= S(q_4, 111) \\
 &= S(q_4, 11) \\
 &= S(q_4, 1) \\
 &= q_4
 \end{aligned}$$

Not accepted.

②  $10^* 1$

$$S_2 = \{ 1\wedge 1, 101, 1001, 10001, \dots \}$$

$$= \{ 11, 101, 1001, 10001, \dots \}$$

It is not accepted

③ 0101

minipal state : p17

$$|S_3| = 4$$

$$\delta(q_0, 0101) = q_2$$

$$\delta(q_0, 0101) = P$$

$$\therefore \delta(q_1, 101) = P$$

$$\therefore \delta(q_2, 01) = P$$

$$\therefore \delta(q_1, 1) = P$$

$$= q_2 \text{ Accepting state}$$

start notleamt : p17

111 ①

$$z = 121$$

$$(111, \epsilon P) b = (111, \epsilon P) z$$

$$(111, \epsilon P) z =$$

$$(111, \epsilon P) =$$

$$(1, \epsilon P) =$$

$$P =$$

but q333, 101

## \* NFA with Null Transitions

A nondeterministic finite automata with  $\lambda$  (null) transitions [abbreviated NFA -  $\lambda$ ] is 5 tuple  $(Q, \Sigma, q_0, A, \delta)$  where  $Q$  and  $\Sigma$  are Finite sets  $q_0 \in Q$ ,  $A \subseteq Q$

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$$

① It contain empty inputs and they called as null transitions.

② If the finite automata  $M = \{Q, \Sigma, q_0, A, \delta\}$  is automata with  $n$  null transitions then automata without  $n$  null transitions will be  $M' = \{Q, \Sigma, q_0, A', \delta'\}$

NFA with null transitions is allowed to make transition not only on input form the alphabet but also with null input. i.e., without any symbol. This transition without input is called null transitions.

Null transition makes the designing of NFA simpler as it reduces the complexity of the design and in some cases reduces the number of states too.

Ex:

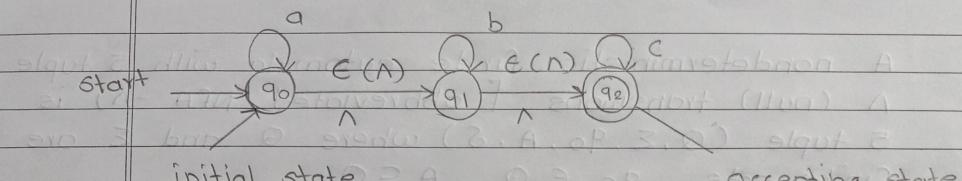


fig: automata with  $n$  (null) transitions

$$S \leftarrow (S_0 S_1 S_2) \times Q : S$$

In above fig. it contain 4 inputs a,b,c  
with  $n$  b so we construct transition table

transition table

status	a	b	c	$\lambda$
$\rightarrow q_0$	$\{q_0\}$	$\emptyset$	$\emptyset$	$\{q_1\}$
$q_1$	$\emptyset$	$\{q_1\}$	$\emptyset$	$\{q_2\}$
$\textcircled{q}_2$	$\emptyset$	$\emptyset$	$\{q_2\}$	$\emptyset$

Q1. In the above automaton, if the initial state is  $q_1$ , then the final state is  $q_2$ .  
Q2. If the initial state is  $q_2$ , then the final state is  $q_1$ .  
Q3. If the initial state is  $q_0$ , then the final state is  $q_2$ .

Q4. If the initial state is  $q_1$ , then the final state is  $q_0$ .  
Q5. If the initial state is  $q_0$ , then the final state is  $q_1$ .  
Q6. If the initial state is  $q_2$ , then the final state is  $q_0$ .

### NFA with $\epsilon$ / In Epsilon Transitions

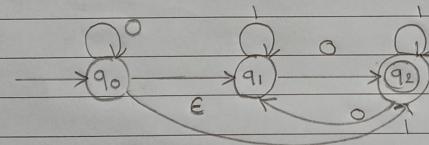
It allows FA to jump from one state to another state without consuming any symbol input

Add one column for  $\epsilon$  moves

- Easier For NFA transitions.

Epsilon Transition are those can be reached from state  $q$  by repeatedly making  $\epsilon$  transition including itself.

Ex : Transaction Diagram



Transaction Table

State	Alphabets		
	0	1	$\epsilon$
$q_0$	$q_0$	$q_2$	$\{q_0, q_1\}$
$q_1$	$q_2$	$q_1$	$\emptyset$
$q_2$	$q_1$	$q_2$	$\emptyset$

### \* Computational Tree for NFA

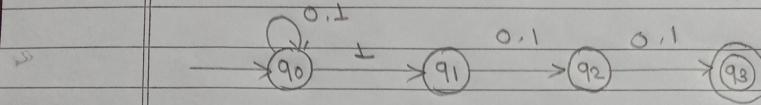
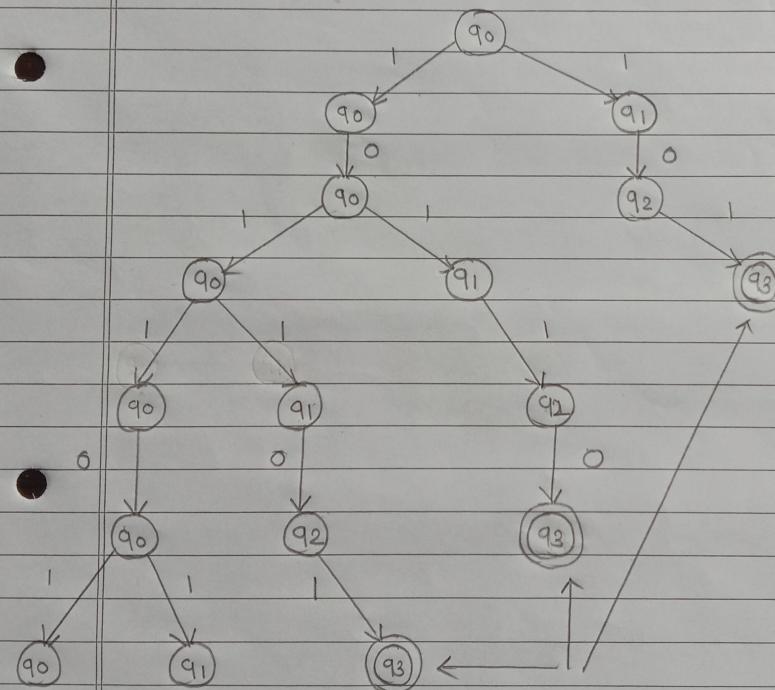


fig : NFA



Accepting states

### \* Difference between DFA and NFA

- |   |  |
|---|--|
| DFA can have more than one transition from a state.                               | NFA can have many transitions from a state.  |
| DFA stands for Deterministic finite automata.                                     | NFA stands for non-deterministic finite automata.  |
| Empty string transition, DFA can not use.   | NFA can use empty string transition.   |
| DFA can be understood as one machine.   | NFA can be understood as multiple machines.  |
| DFA is more difficult to construct.   | NFA is easier to construct.  |
| Time needed for executing input string is less.                                   | Time needed for executing input string is more.  |
| All DFA are NFA.  | Not all NFA are DFA.   |
| DFA requires more space.  | NFA require less space.  |
| Dead state may be required.   | Dead state is not required.  |
| $S : Q \times \Sigma \rightarrow Q$<br>i.e., next possible state belongs to $Q$ . | $S : Q \times \Sigma \rightarrow 2^Q$<br>i.e., next possible state belongs to power set of $Q$ . |

- |  |   |
|--|---|
| ⑥ In DFA, the next possible state is distinctly set.   | In NFA, each pair of set and input symbol can have many possible next states.     |
| ⑦ DFA rejects the string in case it terminates in a state that is different from the accepting states. | NFA rejects the string in the event of all branches dying or refusing the string. |

banteban ad am A71 banteban ad am A70 Ⓛ  
sudam alitum ad am am ad am

Invadas at reis ad ei A74 thui7ib siom ei A79 Ⓛ  
tautono at

caituras red labaan smit tuars 107 labaan smit Ⓛ  
smi ei caitura lugui ecu ei punto lugui par-

A70 am A71 110 101 A71 am A70 111 Ⓛ

stop easi sriper A72 singe siom sriper A70 Ⓛ

tan ei state bao ad pom state bao Ⓛ  
beruper

$0^m \leftarrow 3 \times 0 : 2$  Ⓛ  $0 \leftarrow 3 \times 0 : 2$  Ⓛ  
state address from 1st state address from 1st  
0 to 1st row of registed Ⓛ 0 of registed

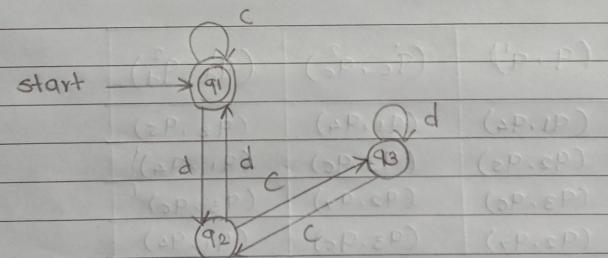
### \* Equivalence of FA's

Two FA's over  $\Sigma$  are equivalent, if they accept the same set of strings over  $\Sigma$ .

They are called to be non equal if they contain one transaction at  $\epsilon$  transaction reaches at final position and other does not.

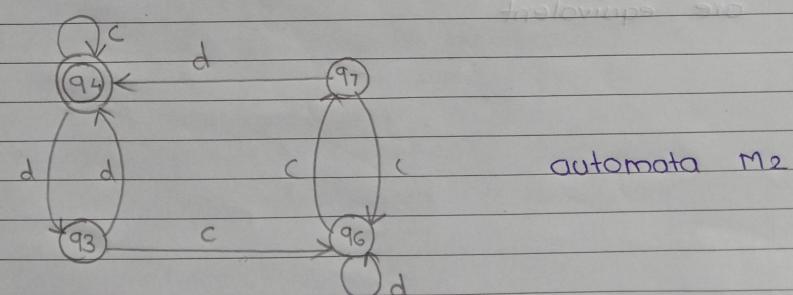
The method called as 'comparision' method.

Ex: Equivalent finite automata



Equivalent automata M1 using set diff

(part 4) as more automata ai simple to compare  
transitions



In two automata  $M_1$  and  $M_2$ , we have  
 (91) and (94) as accepting and initial state.

They contain transactions  $t_c$  and  $t_d$  at each level

Now in  $M_1$  and  $M_2$  we know that  $\Sigma = \{c, d\}$

$M_1$  having start state 91 and

$M_2$  having start state 94

comparision table:

$(q, q')$	$(q'_c, q''_c)$	$(q'_d, q''_d)$
$(q_1, q_4)$	$(q_1, q_4)$	$(q_2, q_5)$
$(q_2, q_5)$	$(q_3, q_6)$	$(q_1, q_4)$
$(q_3, q_6)$	$(q_2, q_7)$	$(q_3, q_6)$
$(q_2, q_7)$	$(q_3, q_6)$	$(q_1, q_4)$

Here the pairs in 2nd and 3rd row are identical or appear in comparision row so they are equivalent

\* Kleene's theorem, (Part 1) p. moore  
parallel to random notes

### Part 1. Moore (Machines)

Developed by Moore in 1956 by scientist  
Moore

Moore machine is a tuple  $(P, (Q, \Sigma, q_0, \delta, \lambda))$

Where  $(Q, \Sigma, q_0, \delta)$  symbols from finite automata

$\lambda$  output symbol / Tape symbol

$\lambda$  mapping from  $Q$  to  $\Delta$ .

i.e., contains all outputs

If moore machine contains the symbols from

$q_1, q_2, \dots, q_n$  &  $n \geq 0$  then

$\lambda(q_1), \lambda(q_2), \dots, \lambda(q_n)$  will contain all the strings.

Ex: consider moore machine  $M = (Q, \Sigma, \Delta, \delta, q_0, \lambda)$   
Where,

$$Q = \{q_0, q_1, q_2, q_3\}$$

$$\Sigma = \{0, 1\}$$

$$\Delta = \{a, b\}$$

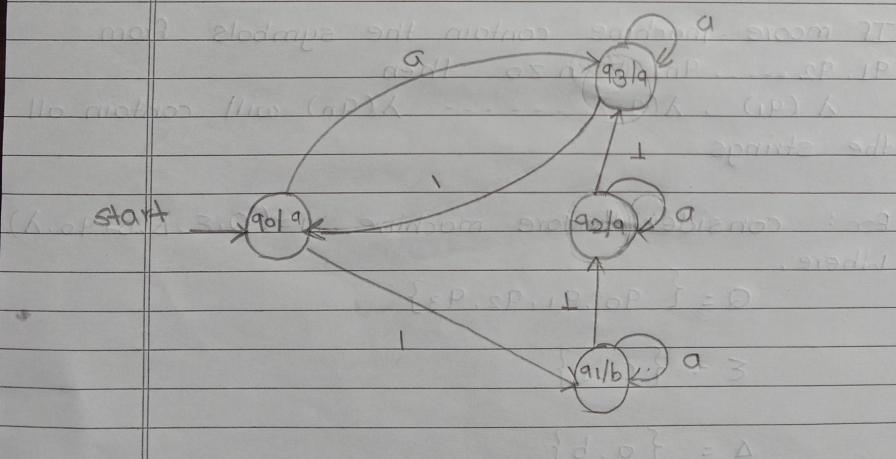
Now from given data we can construct the Moore machine as follows.

(q<sub>0</sub>, q<sub>1</sub>, q<sub>2</sub>, q<sub>3</sub>) S<sub>00011</sub> / λ → q<sub>0</sub>

trans state	0	1	output λ	key/over
q <sub>0</sub>	q <sub>3</sub>	q <sub>1</sub>	a	S <sub>00011</sub>
(q <sub>1</sub> , q <sub>2</sub> , q <sub>3</sub> )	q <sub>1</sub>	q <sub>2</sub>	b	S <sub>00011</sub>
q <sub>2</sub>	q <sub>2</sub>	q <sub>3</sub>	a	S <sub>00011</sub>
q <sub>3</sub>	q <sub>3</sub>	q <sub>0</sub>	(a, b)	S <sub>00011</sub>

Now if we consider input string 0111 then transition will be

q<sub>0</sub> → q<sub>3</sub> → q<sub>0</sub> → q<sub>1</sub> → q<sub>2</sub> → q<sub>3</sub>



From the transition diagram the output string will be aaaba

### \* kleenes theorem

Any regular language can be accepted by finite automata.

Proof :

If a language accepted by finite automata then it will accepted by also nondeterministic finite automata with null transition.

It will contain the operations like union, kleene and concatenation.

If  $L_1$  and  $L_2$  are two regular languages then by induction hypothesis, we get,

$L_1 \cup L_2$ ,  $L_1 L_2$  and  $L_1^*$

Now consider if  $L_1$  and  $L_2$  are recognized NFA-n with  $M_1$  and  $M_2$  then

$$M_i = (Q_i, \Sigma, q_i, A_i, \delta_i)$$

Union :

$$M_u = (Q_u, \Sigma, q_u, A_u, \delta_u)$$

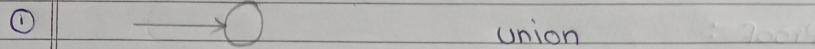
$$\therefore Q_u = Q_1 \cup Q_2 \cup \{q_u\}$$

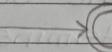
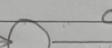
$$A_u = A_1 \cup A_2$$

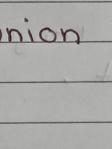
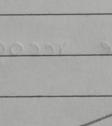
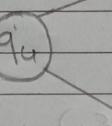
Here  $Q_u$  is accepted by either  $Q_1$  or  $Q_2$

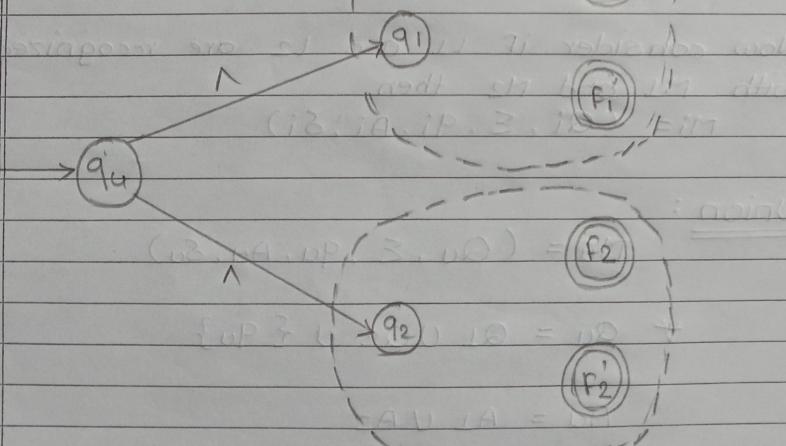
$$A_u = A_1 \cup A_2$$

shai? NFA - A for three basic regular language



at ②  concatenation  $\rightarrow \Sigma^*$   
at ③  Kleene star  $\rightarrow \Sigma^*$

① Union :  union  $\rightarrow \Sigma^*$   
at ②  concatenation  $\rightarrow \Sigma^*$   
at ③  Kleene star  $\rightarrow \Sigma^*$



shai? NFA - A for three basic regular language

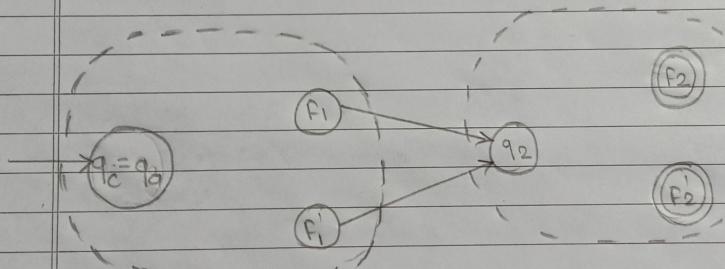
② Concatenation :

$$M_c = (Q_c, \Sigma, q_c, A_c, \delta_c)$$

Here no need of new state in it

$$q_c = q_1 \quad \text{initial state}$$

$$A_2 = A_c \quad \text{Accepting state}$$



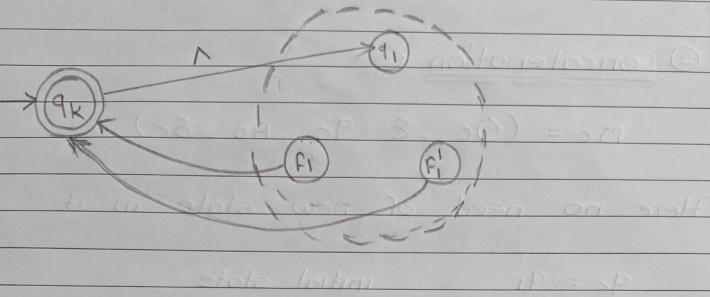
③ kleene :

$$M_k = (Q_k, \Sigma, q_k, A_k, \delta_k)$$

$q_k$  new state not in  $Q_L$

$$Q_k = Q_L \cup \{q_k\}$$

$$A_k = \{q_k\}$$



④ Reed :

$$(x_3, x_6, x_7, z, y) = \text{all}$$

10 at top state with >P

$$\{x_7\} \cup 10 = \text{all}$$

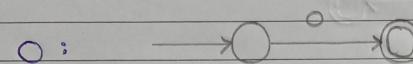
$$AR = \{x_7\}$$

Ex : construct the NPA - A for  $(00+1)^*$  (10)\* (b)

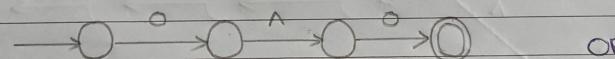
$\oplus (00+1)^* (10)^*$

$(00+1)^*$

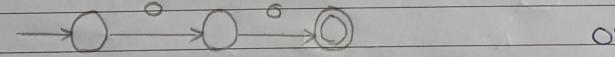
a)  $00 \Rightarrow$



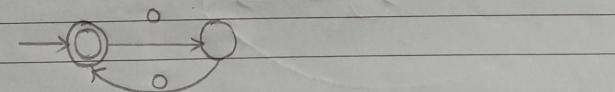
NPA - A for  $00$ ,



OR



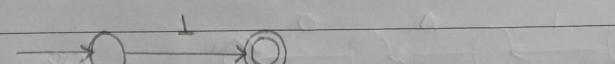
OR



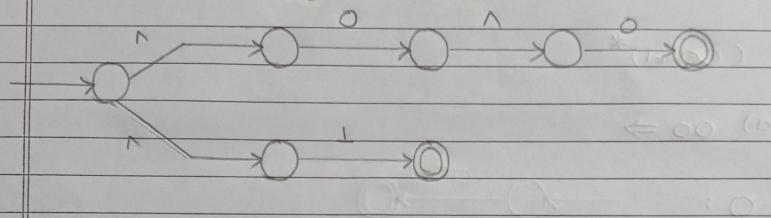
b)  $00$  : non-deterministic pd (1+00)



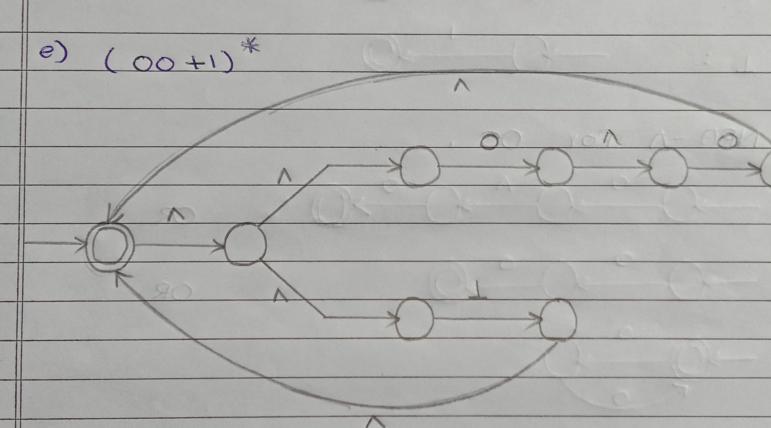
c)  $1$  :



d)  $(00+1)^*$  (0-111 011 1011) by concatenation



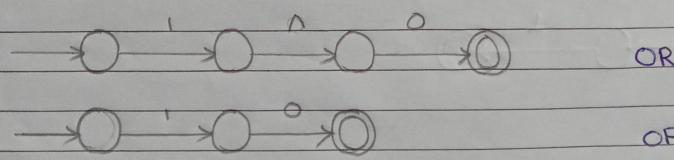
e)  $(00+1)^*$

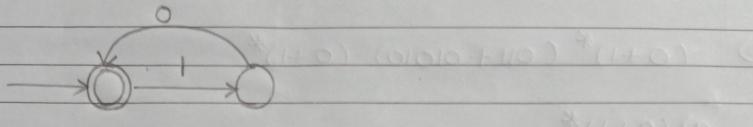


$(00+1)$  by concatenation :  $00 \cup 1$

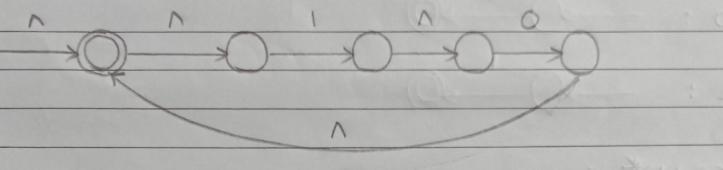
f)  $(10)^*$

g)  $10$





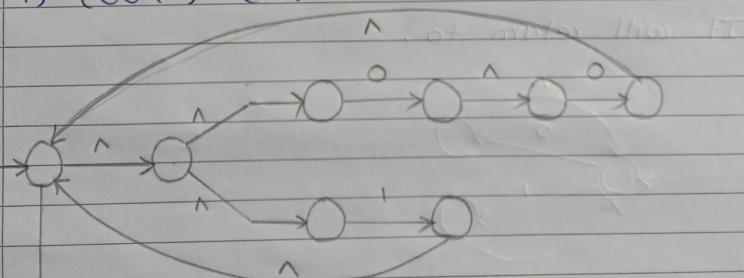
h)  $(10)^*$



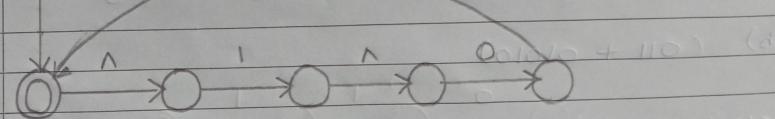
Now as we having 2 NFA - A we have to concatenate each other.

$$\underline{(00+1)^* \cdot (10)^*} \neq (10)^* \cdot (00+1)^*$$

i)  $(00+1)^*, (10)^*$

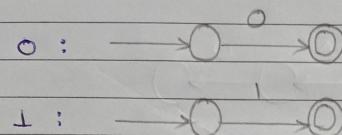


Ans bar to not a all bumpy all  
mengko hulung langsung mala mala mala

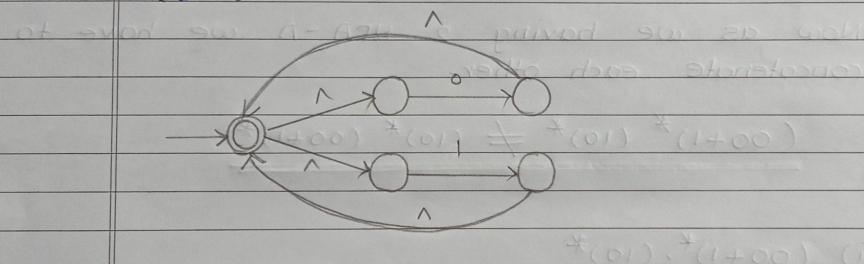


②  $(0+1)^* (011 + 01010) (0+1)^*$

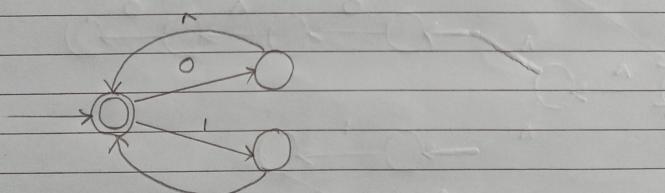
a)  $(0+1)^*$



$(0+1)^*$  :

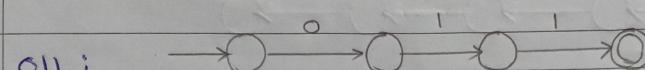


It will return to,

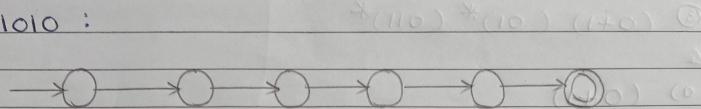


The required NFA - N for 1st and 3rd sub  
Regular Expression from original Regular Expression

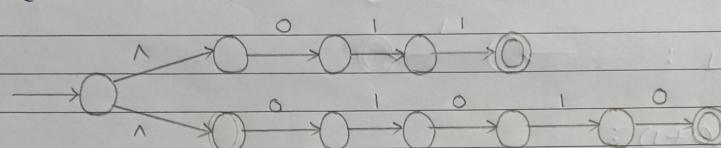
b)  $(011 + 01010)$



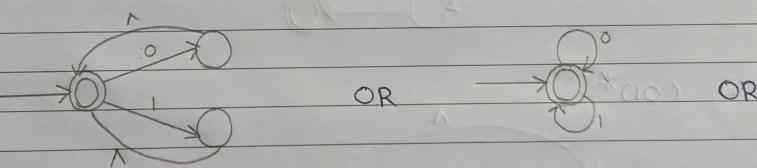
01010 :



$(011 + 01010)$  :

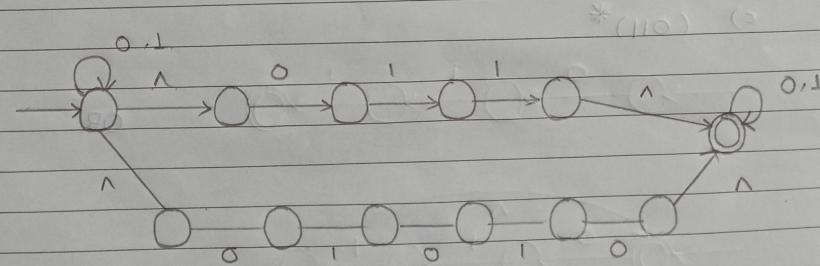


final result,



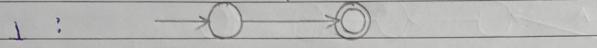
0,1

$(0+1)^*$

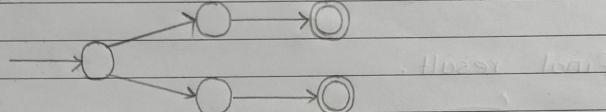


③  $(0+1)(01)^*(011)^*$

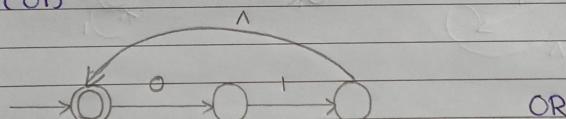
a)  $(0+1)$



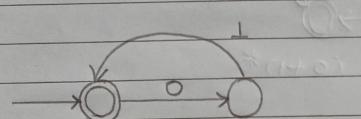
$(0+1)$ :



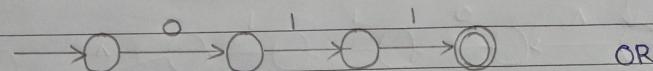
b)  $(01)^*$



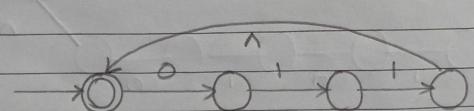
OR



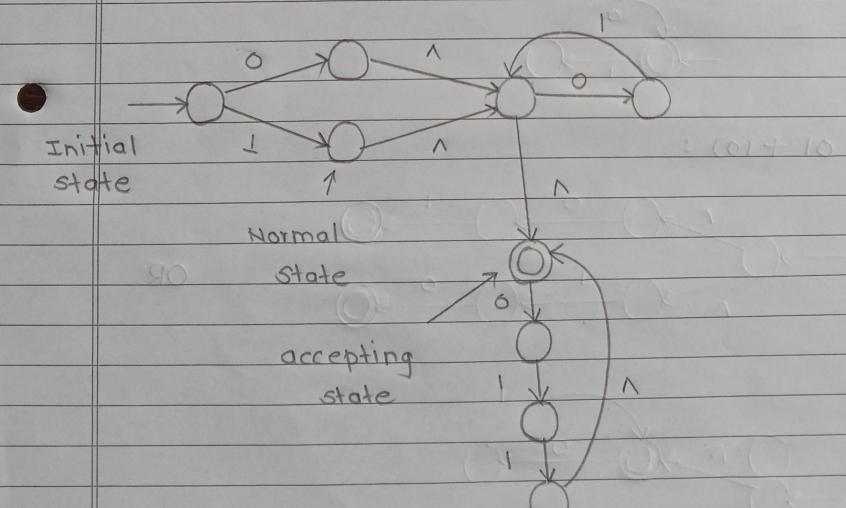
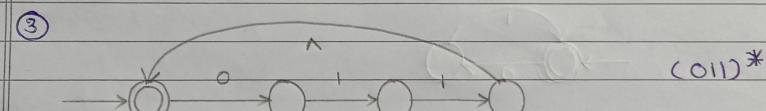
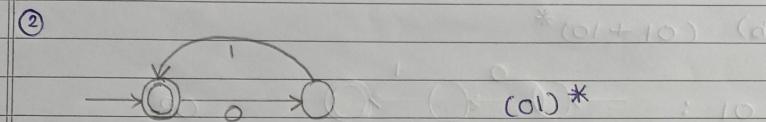
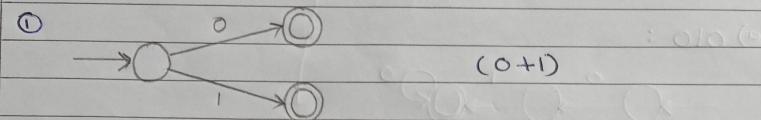
c)  $(011)^*$



OR

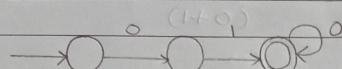


final result :

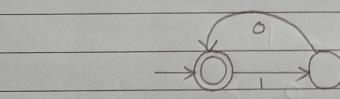
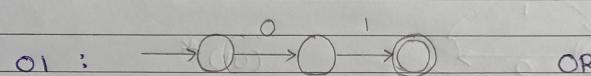


Q)  $010^* + 0(01+10)^* 11$  : Minimise

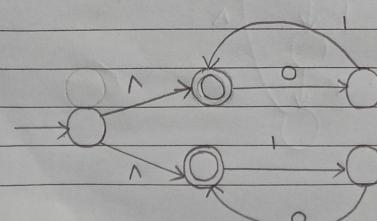
a)  $010$ :



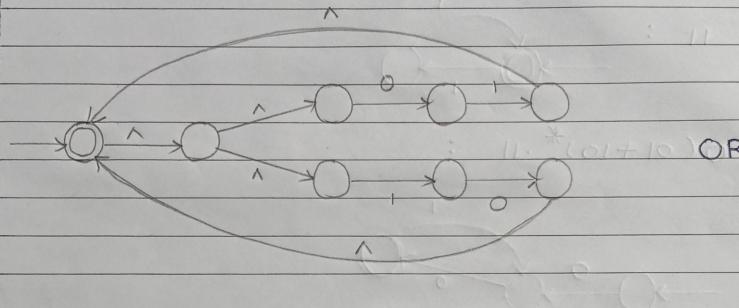
b)  $(01+10)^*$



$(01+10)$ :

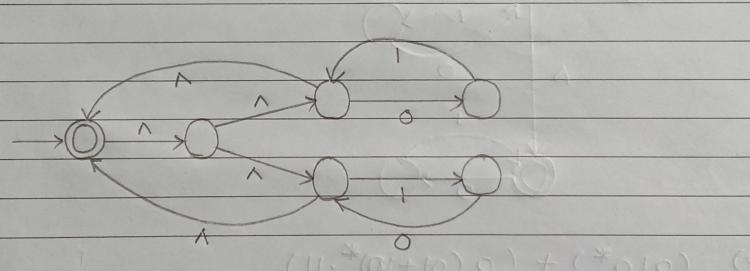


$(01+10)^*$  :



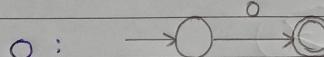
$11^*(01+10)0$  (b)

OR

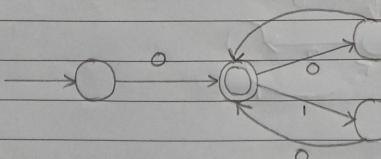


$(11^*(01+10)0) + (*010)$  (c)

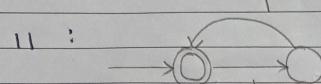
c)  $0(01+10)^*$



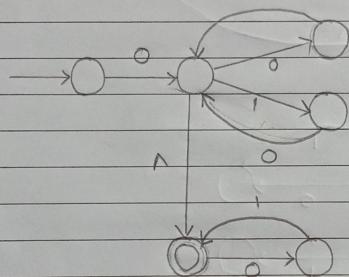
90



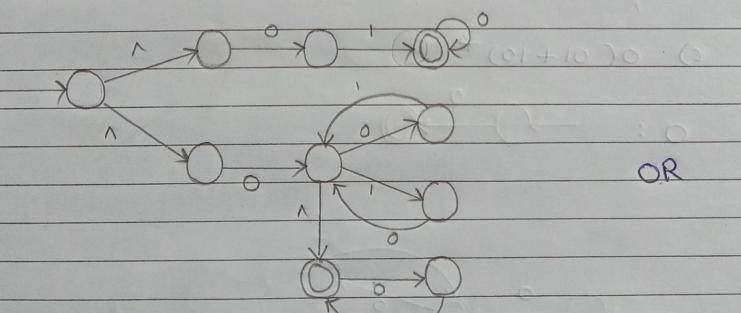
d)  $0(01+10)^* \cdot 11$  :  $* (01+10)$



$0(01+10)^* \cdot 11$  :



e)  $(010^*) + (0(01+10)^* \cdot 11)$



OR