THEORY OF COMPUTATION

Introduction to the TOC

The purpose of TOC is to develop formal mathematical models of computation that reflect real-world computers.

TOC's areas:

(i) Complexity theory - classify prob according to their degree of difficulty

(ii) Computation theory - solvable / unsolvable

(iii) Automata theory - work with different types of computation model.

Example: Finite Automata - Text processing, compiler

context-free grammar - Define programming lang, AI

Turing machine - Abstract model of real computer

UNIT I : FINITE AUTOMATA

BASIC MATHEMATICAL NOTATION AND TECHNIQUES:

ALPHABET

An alphabet is a finite non-empty set of symbols and Σ is used to represent alphabet.

eg: $\Sigma = \{a, b\}$ $\{0, 1, 2, \ldots\}$

STRING:

A string is a finite sequence of symbols chosen from some alphabet

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

String operations

(i) Empty string :

string with zero occurance of symbols and represented by ϵ (epsilon)

(ii) Length of a string

No. of position for symbols in string (or) no. of symbols.

$|1011| = |1011|$
4

(iii) Power of the Alphabet

No. of occurance of particular alphabet

$\Sigma^0 = \epsilon$
$\Sigma^1 = \{0,1\}$
$\Sigma^2 = \{00, 01, 10, 11\}$

(iv) Concatenation of string

$x = ab$, $y = cd$ then $xy = abcd$

(v) Reverse (w^R)

$w = ab$, $w^R = ba$

(vi) Substring

string $w = abcd$

substring $= bc$

LANGUAGE:

Set of all strings which are chosen from some Σ^* where Σ is an alphabet.

Kleene closure $\Sigma^* \rightarrow \Sigma^+ \cup \{\epsilon\} \geq 0$

Positive closure $\Sigma^+ \rightarrow \Sigma^1 \cup \Sigma^2 \cup \ldots \geq 1$

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

WORKING WITH SET

a b.

ab bo $\in$

a

b

ab

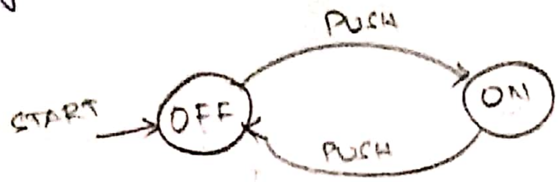
0, 1

$w \in \{a, b\}^*$

* Set element | Member
* subset
* Proper subset [Subset should not be equal to set]
* Multiset [Multiple occurance of elements)
* Infinite set
* Integer $\mathbb{Z}$
* Empty set ϕ
* Singleton set
* Unordered pair
* Union, intersection, complement
* Sequence $(0,1,2) \neq (1,2,0)$
  - Finite (TUPLE)
  - Infinite
* Power set
* cartesian product (or) cross product

FINITE STATE M/C OR STATE M/C [FSM]

It is a mathematical model of computation used to design both computer programs and sequential circuits.



FINITE STATE AUTOMATA [FSA]

It is used to recognize patterns within inputs taken from some character sets

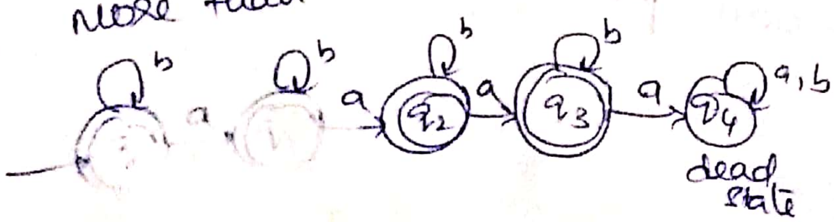
DETERMINISTIC FINITE AUTOMATA (DFA)

Finite automaton can be represented by a 5- tuple $(Q, \Sigma, \delta, q_0, F)$

* Q - finite nonempty set of states
* Σ - Finite nonempty set of inputs - i/p alphabet
* δ - Transition function which maps $Q \times \Sigma \rightarrow Q$
* $q_0 \in Q$ is initial state
* $F \subseteq Q$ set of final states

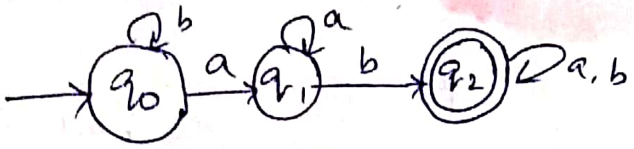
EXAMPLES OF DETERMINISTIC FINITE AUTOMATA

✓ Not more than 3 a's $\Sigma = \{a, b\}$

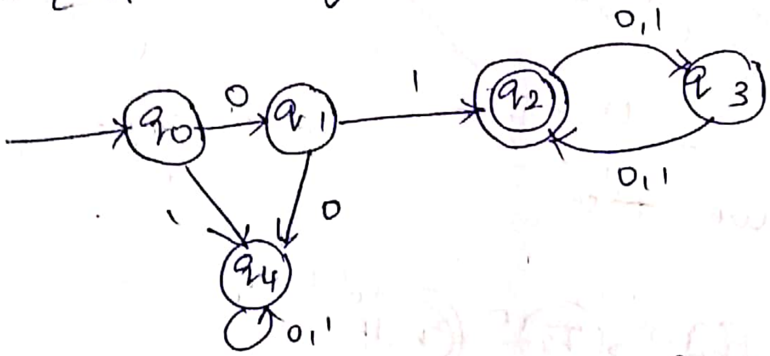


Scanned with CamScanner

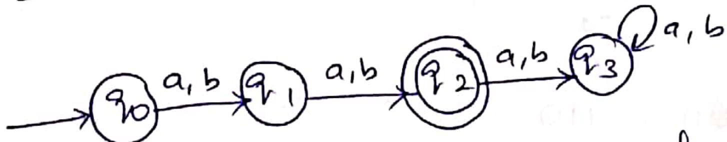
2. $L = \{x \in \{a,b\}^* \mid x$ contains substring $ab\}$



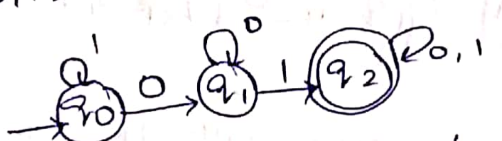
3. $L = \{w \mid w$ is of even length & begins with $01\}$



4. $\Sigma = \{a,b\}$ string length is 2

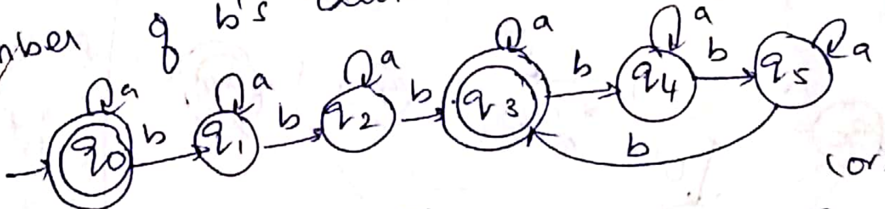


5. $\Sigma = \{0,1\}$ sequence 01 somewhere in the string



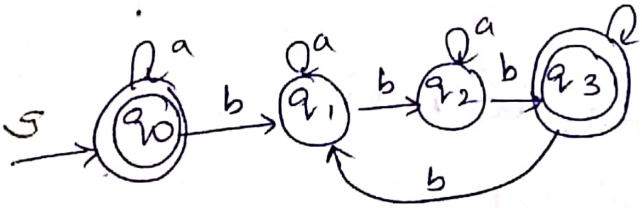
6. Number of b's divisible by 3

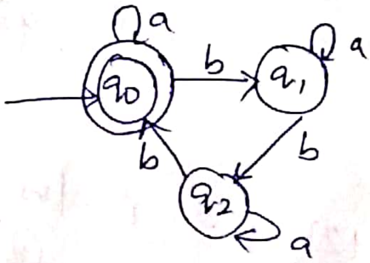
copied



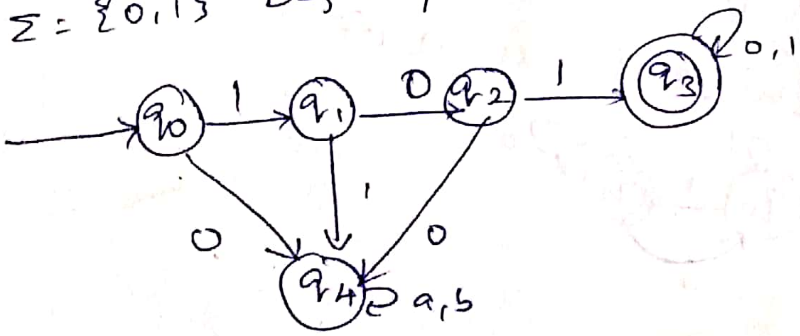
(or)

(or)

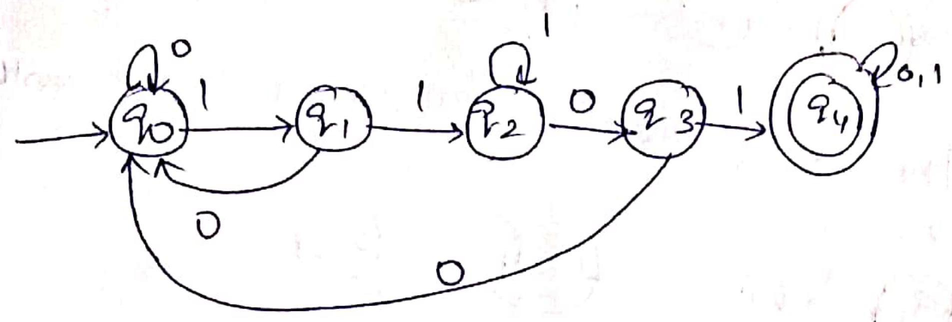




7. $\Sigma = \{0,1\}$ beginning with 101



8) 1101 substring



i/p 1101

1) 0 0 1 1 1 0 1 1

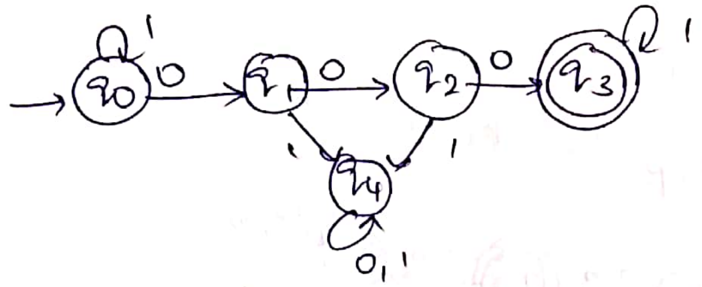
2) 0 0 1 1 0 1

3) 1 1 1 0 1 0 1

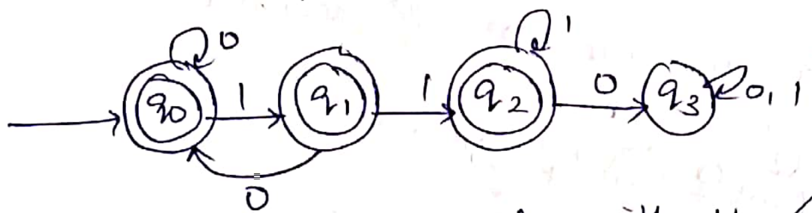
14)

15)

9) 3 consecutive 0's



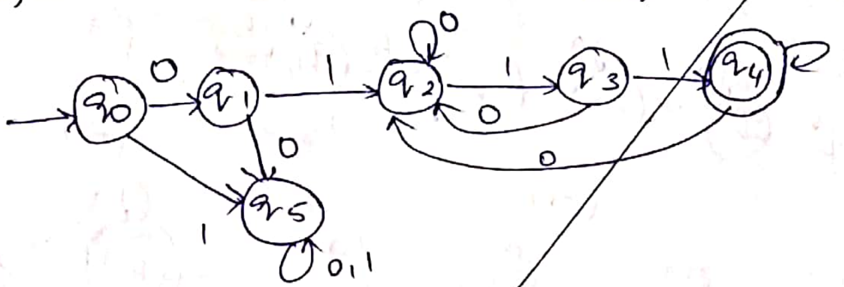
10) Not containing 110



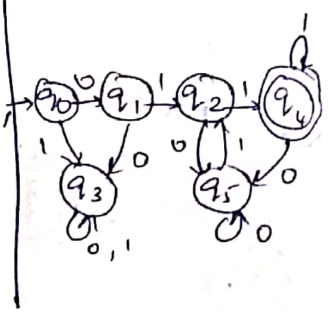
16)

11) Begins with 01 & ends with 11

Length : 4



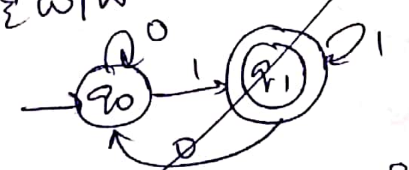
Length : 3



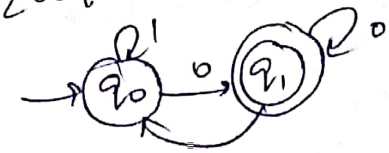
17)

18)

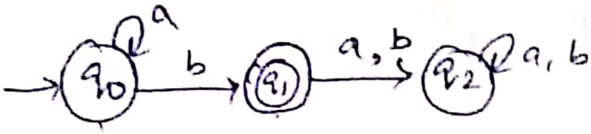
12) L = {w | w ends in a 1}



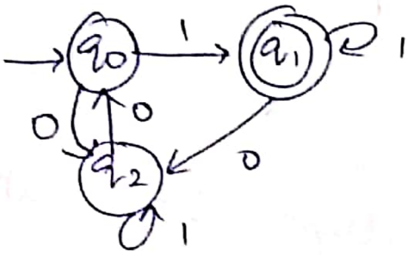
13) L = {w | w ends in a 0}



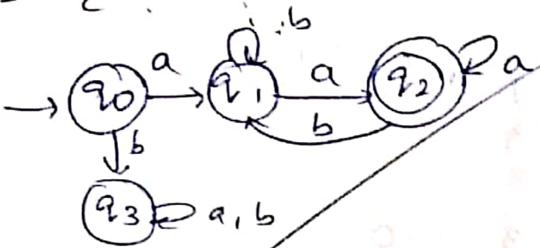
14) $L = \{a^n b : n \geq 0\}$



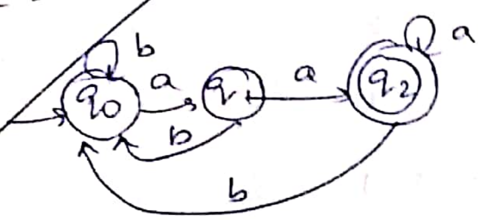
15) $L = \{w | w$ contains atleast one 1 and an even number of 0's followed by the last 1$\}$



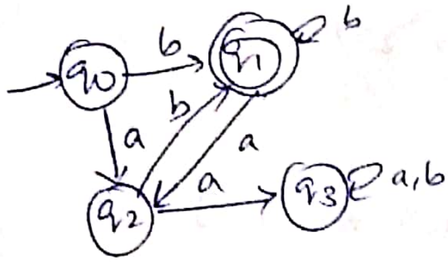
16) $L = \{awa : w \in \{a,b\}^*\}$



17) $L = \{w / w \in \{a,b\}^*$ ends with aa$\}$

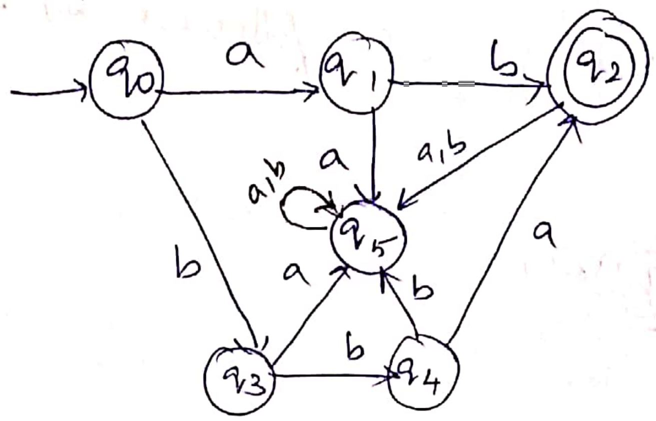


18) Ends with b and doesn't contain aa



Scanned with CamScanner

19) string contains either ab (or) bba



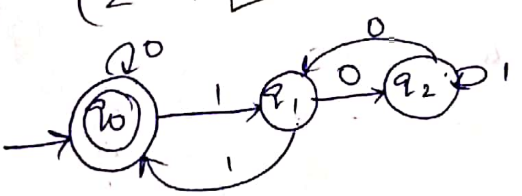
20) $\Sigma = \{0,1\}$ set of all strings when interpreted as binary integer is multiple of 3

(b * M + w) mod n

Base — b; mod value — M; i/p — w; number — n

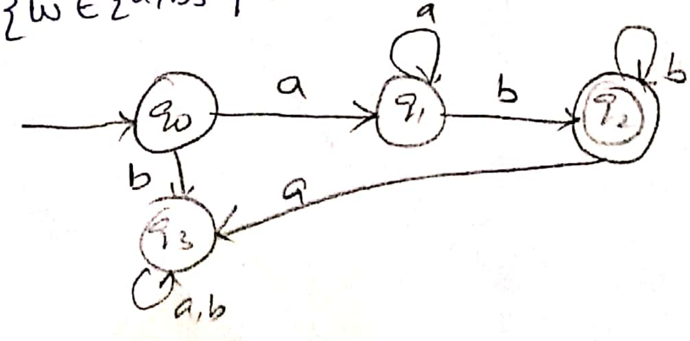
start state, i/p, end state

(2 * 0 + 0) mod 3 = 0.
(2 x 0 + 1) mod 3 = 1.
(2 x 1 + 0) mod 3 = 2
(2 x 1 + 1) mod 3 = 0
(2 x 2 + 0) mod 3 = 1
(2 x 2 + 1) mod 3 = 2



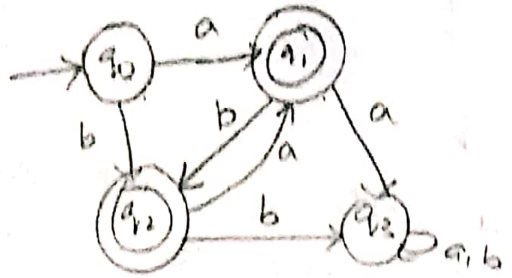
	0	1
→ q_0	q_0	
q_1		
* q_2		

21) $L = \{w \in \{a,b\}^* / w = a^n b^m, n, m > 0\}$

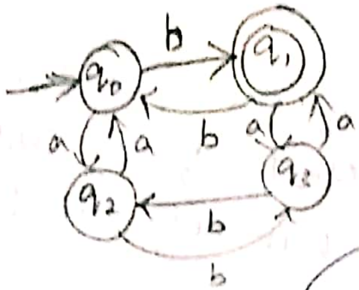


Scanned with CamScanner

22) $L = \{w \mid w \in \{a,b\}^*$, w has neither aa nor bb as substring$\}$



23) $L = \{w \mid w \in \{a,b\}^*$, odd number of b's & even no. of a's$\}$



LANGUAGE

Let $M = (Q, \Sigma, \delta, q, F)$ be a finite automaton. The language $L(M)$ accepted by M is defined to be the set of all strings that are accepted by M.

$L(M) = \{w \mid w$ is a string over Σ and M accepts $w\}$

LANGUAGE IS REGULAR

A language L is called regular if and only if there exists some deterministic finite accepter M such that $L = L(M)$

NOTATIONS:

1. Language
2. Transition diagram
3. Transition table

FUNCTION OF DFA (ETF)

BASIS:

$$\hat{\delta}(q, \epsilon) = q$$

If we are in a state q and read no input then we are still in the same state q.

INDUCTION:

w is the string of the form $w = xa$.

where a is the last symbol of w and x is a string consisting of all but not the last symbol

eg: $w = 1101$; $x = 110$, $a = 1$

$$\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), a)$$

To compute $\hat{\delta}(q, w)$, first compute $(\hat{\delta}(q, x))$. suppose $\hat{\delta}(q, x) = P$, then $\hat{\delta}(q, w) = \delta(P, a)$

EXAMPLE: $L = \{w / w \text{ ends in a } 1\}$

δ	0	1
$\rightarrow q_0$	q_0	q_1
$* q_1$	q_0	q_1

$w = 001$

$\hat{\delta}(q_0, \epsilon) = q_0$

$\hat{\delta}(q_0, 0) = \delta(\hat{\delta}(q_0, \epsilon), 0) = \delta(q_0, 0) = q_0$

$\hat{\delta}(q_0, 00) = \delta(\hat{\delta}(q_0, 0), 0) = \delta(q_0, 0) = q_0$

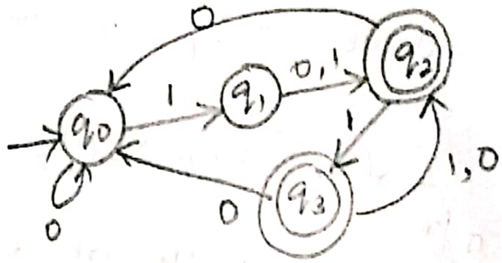
$\hat{\delta}(q_0, 001) = \delta(\hat{\delta}(q_0, 00), 1) = \delta(q_0, 1) = q_1$

NON - DETERMINISTIC FINITE AUTOMATA (NFA)

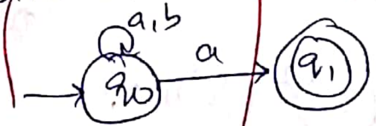
→ models of search - and - backtrack algorithm & simplicity
5 tuple Notation $(Q, \Sigma, \delta, q_0, F)$

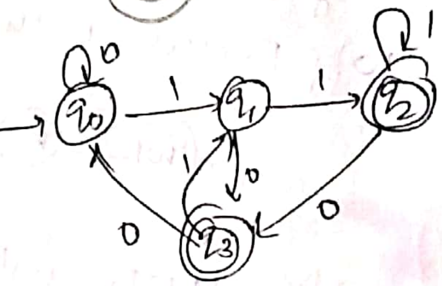
$$\delta = \Sigma \times Q \rightarrow 2^Q$$

1. second last symbol is 1 , $\Sigma = \{0,1\}$



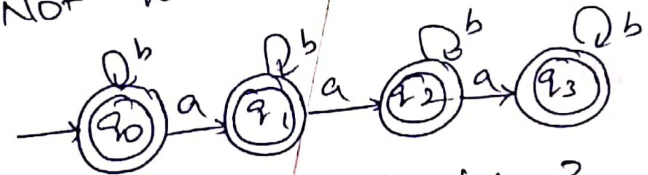
2. Ends with a



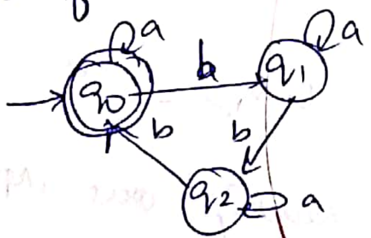


3. $(a|b)^*ab$

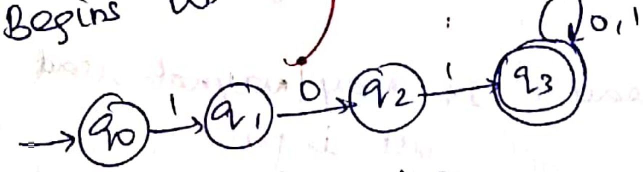
4. Not more than 3 a's



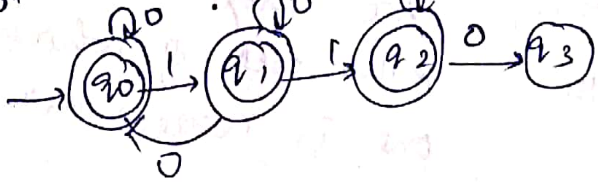
5. No. of b's divisible by 3



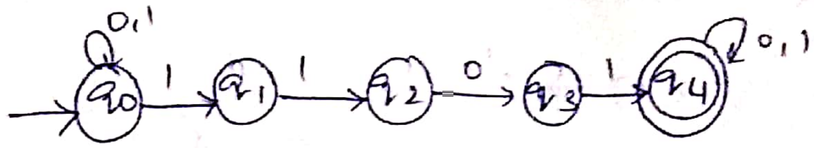
6. Begins with 101



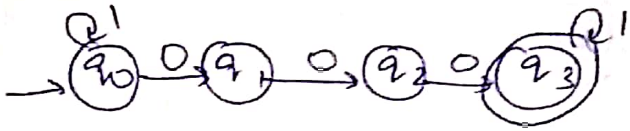
7. Not containing 110



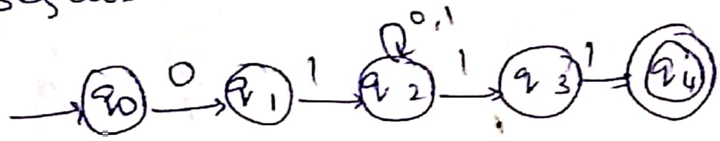
8) substring 1101



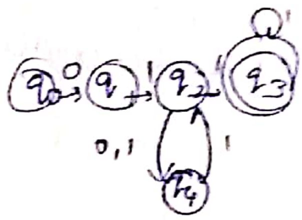
9) ~~Sttb~~ Exactly 3 consecutive 0's



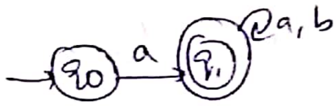
10) Begins with 01 ends with 11



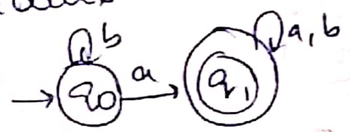
(or)



11) Starts with a



12) Contains a



	DFA	NFA
1.	$\delta : Q \times \Sigma \rightarrow Q$	$\delta : Q \times \Sigma \rightarrow 2^Q$
2.	Reading one input, moves to only one state	On reading one input, it can go to zero or more states
3.	All states has to read all i/ps	It may/maynot read all inputs
4.	For an accepted string, there is only one path	Acc string has many paths out of which only one is correct path.

EXTENDED TRANSITION FUNCTION FOR NFA (ETF)

BASIS:

$$\hat{\delta}(q, \epsilon) = \{q\}$$

Being in a state Reading no input, the system remains in the same state

INDUCTION:

Suppose $w = xa$, where a is the final input and x is rest of input.

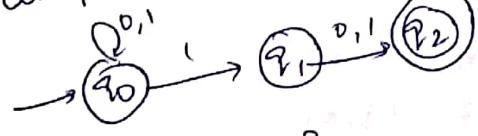
$$\hat{\delta}(q, x) = \{p_1, p_2, \dots p_k\}$$

Let $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \dots r_n\}$

Then $\hat{\delta}(q, w) = \{r_1, r_2, \dots r_m\}$

$\hat{\delta}(q, w)$ is computed by first computing $\hat{\delta}(q, x)$ and by then following any transition from any of those states ~~tats~~ that is labelled a.

	0	1
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	ϕ	ϕ



$\hat{\delta}(q_0, \epsilon) = \{q_0\}$

$\hat{\delta}(q_0, 0) = \delta(\hat{\delta}(q_0, \epsilon), 0) = \delta(\{q_0\}, 0) = \{q_0\}$

$\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1) = \delta(\{q_0\}, 1) = \{q_0, q_1\}$

$\delta(q_0, 010) = \delta(\hat{\delta}(q_0, 01), 0) = \delta(\{q_0, q_1\}, 0) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$

$\hat{\delta}(q_0, 010) =$

LANGUAGE

$$\boxed{L(A) = \{ w \mid \hat{\delta}(q_0, w) \cap F \neq \phi \}}$$

...CE OF NFA & DFA

METHODS OF CONVERSION

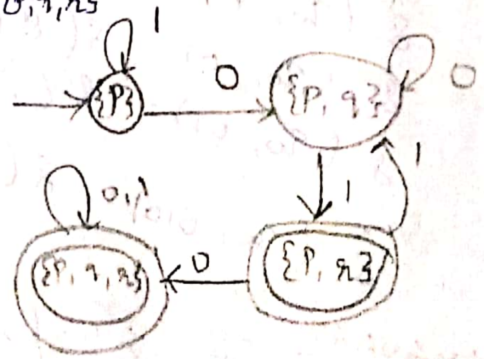
* SUBSET construction
* LAZY construction

(1)

δ	0	1
$\rightarrow p$	{p, q}	{p}
q	ϕ	{r}
* r	{p, r}	{q}

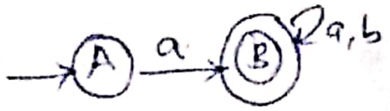
LAZY CONSTRUCTION

$\rightarrow \delta$	0	1
p	{p, q}	{p}
q	ϕ	{r}
* r	{p, r}	{q}
{p, q}	{p, q}	{p, r}
* {p, r}	{p, q, r}	{p, q}
* {q, r}	{p, r}	{q, r}
* {p, q, r}	{p, q, r}	{p, q, r}



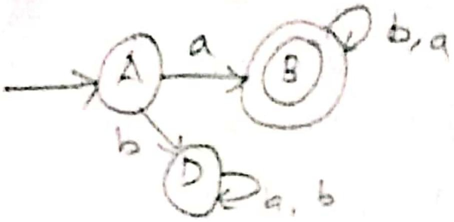
SUBSET CONSTRUCTION

δ	0	1
$\rightarrow$ {p}	{p, q}	{p}
{p, q}	{p, q}	{p, r}
* {p, r}	{p, q, r}	{p, q}
* {p, q, r}	{p, q, r}	{p, q, r}

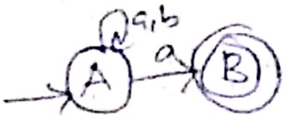
2. 

	a	b
{A}	{B}	$\emptyset$
{B}	{B}	B

	a	b
→ A	B	D
* B	B	B
D	D	D

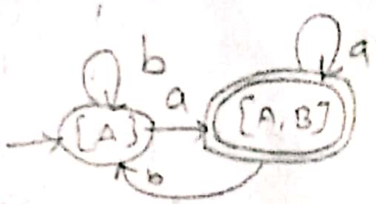


3. L = { ends with an a } $\Sigma = \{a, b\}$

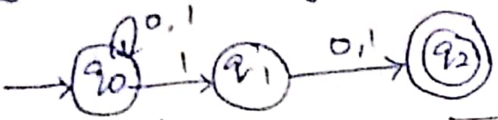


	a	b
→ {A}	{A, B}	{A}
* {B}	ϕ	ϕ

	a	b
→ [A]	[A, B]	[A]
* [A, B]	[A, B]	[A]

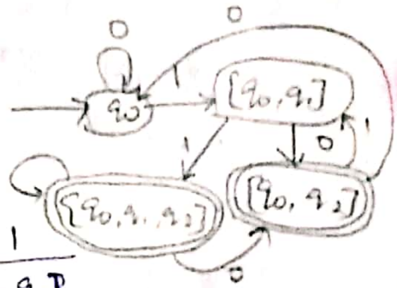


4. Second last symbol is 1



	0	1
→ q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
* q_2	ϕ	ϕ

	0	1
→ $[q_0]$	$[q_0]$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_2]$	$[q_0]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_2]$	$[q_0, q_1, q_2]$

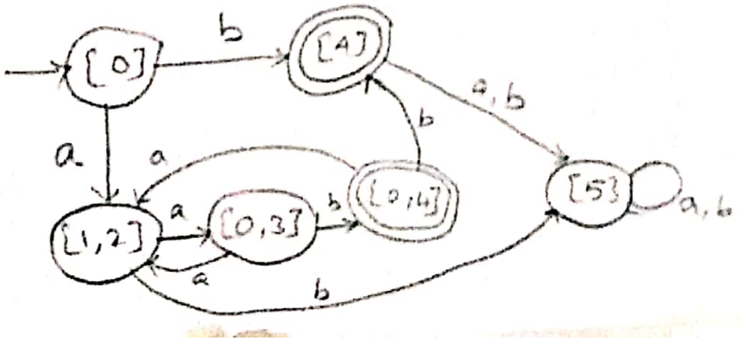


5.

δ	a	b
→ 0	{1, 2}	{4}
1	{0}	ϕ
2	{3}	ϕ
3	ϕ	{0}
* 4	ϕ	ϕ

δ	a	b
→ [0]	[1, 2]	[4]
* [4]	[5] D	[5] D
[1, 2]	[0, 3]	~~[5]~~ D
[0, 3]	[1, 2]	[0, 4]
* [0, 4]	[1, 2]	[4]
[5]	[5]	[5]

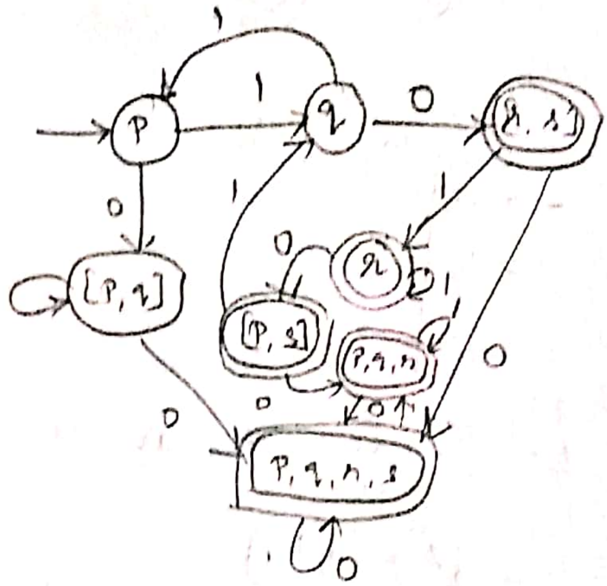
[D]



Scanned with CamScanner

6.

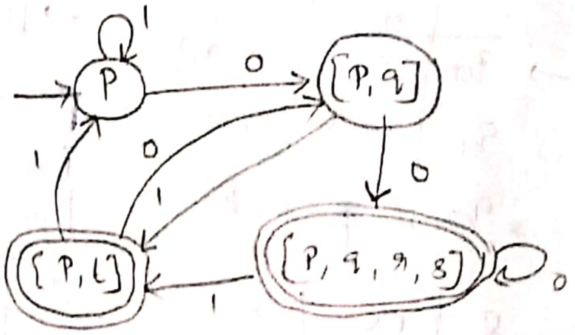
δ	0	1
→ p	{p, q}	{q}
q	{r, s}	{p}
* r	{p, s}	{r}
* s	{q, r}	φ



δ	0	1
→ p	[p, q]	[q]
q	[r, s]	[p]
* r	[p, s]	[r]
[p, q]	[p, q, r, s]	[p, q]
* [r, s]	[p, q, r, s]	[r]
* [p, s]	[p, q, r]	[q]
* [p, q, r]	[p, q, r, s]	[p, q, r]
* [p, q, r, s]	[p, q, r, s]	[p, q, r]

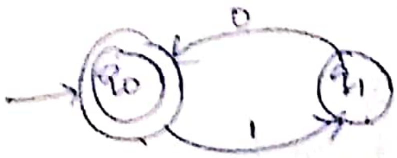
7.

δ	0	1
→ p	{p, q}	{p}
q	{r, s}	{t}
r	{p, r}	{t}
* s	φ	φ
* t	φ	φ



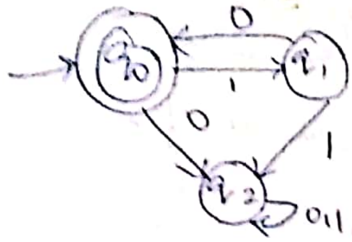
δ	0	1
→ p	[p, q]	[p]
[p, q]	[p, q, r, s]	[p, t]
* [p, t]	[p, q]	[p]
* [p, q, r, s]	[p, q, r, s]	[p, t]

8. $L = \{(10)^n : n \geq 0\}$



	0	1
$* \to q_0$	ϕ	$[q_1]$
q_1	$[q_0]$	ϕ

	0	1
$* \to q_0$	q_2	q_1
q_1	q_0	q_2
q_2	q_2	q_2



CONVERTING NFA to DFA - Theorem [JUNE 2016]

If $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ is the DFA constructed from NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ by subset construction, then $L(D) = L(N)$

BASIS

$|w| = 0 \quad w = \epsilon$

$\hat{\delta}_D(\{q_0\}, \epsilon) = \hat{\delta}_N(q_0, \epsilon) = q_0$

INDUCTION

Let $w = xa$ where a is the final symbol of w by inductive hypothesis.

$\hat{\delta}_D(\{q_0\}, x) = \hat{\delta}_N(q_0, x)$

Let both these sets of N states reach $\{p_1, p_2, \ldots p_k\}$

By inductive part of definition, NFA tells us

$\hat{\delta}_N(q_0, w) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$

The subset construction on the other hand tells us that

$\delta_D(\{p_1, p_2, \ldots p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$

we know, $\hat{\delta}_D(\{q_0\}, x) = \{p_1, p_2, \ldots p_k\}$

$\hat{\delta}_D(\{q_0\}, w) = \delta_D(\hat{\delta}_D(\{q_0\}, x), a)$

$= \delta_D(\{p_1, p_2, \ldots p_k\}, a)$

$= \bigcup_{i=1}^{k} \delta_N(p_i, a) = \hat{\delta}_N(q_0, w)$

Hence $L(D) = L(N)$

FINITE AUTOMATA WITH EPSILON TRANSITION

DFA $\quad \delta \Rightarrow Q \times \Sigma \rightarrow Q$

NFA $\quad \delta \Rightarrow Q \times \Sigma \rightarrow 2^Q$

NFA-ϵ $\quad \delta \Rightarrow Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$

1. Any number of zeros followed by any number of one's followed by any number of 2's

[NOV 2014]
[DEC 2015]
[JUNE 2016]

012
01
1
2
0
12
21-X
210-X
0

ϵ-closure $(q_0) = \{q_0, q_1, q_2\}$
ϵ-closure $(q_1) = \{q_1, q_2\}$
ϵ-closure $(q_2) = \{q_2\}$

ϵ-closure $(\delta(q_0, 0)) = \epsilon$-closure $(\delta(\{q_0, q_1, q_2\}, 0))$
$= \epsilon$-closure $(q_0) = \{q_0, q_1, q_2\}$

$q_0 \cup \phi \cup \phi$
q_0

ϵ-closure $(\delta(q_0, 1)) = \epsilon$-closure $(\delta(\{q_0, q_1, q_2\}, 1))$
$= \epsilon$-closure $(q_1) = \{q_1, q_2\}$

ϵ-closure $(\delta(q_0, 2)) = \epsilon$-closure $(\delta(\{q_0, q_1, q_2\}, 2))$
$= \epsilon$-closure $(q_2) = \{q_2\}$

ϵ-closure $(\delta(q_1, 0)) = \sim(\delta(\{q_1, q_2\}, 0)) = \sim(\phi) = \phi$

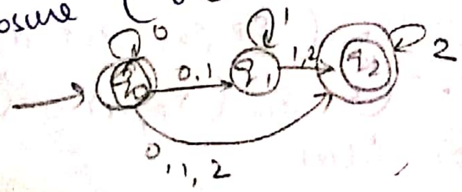
ϵ-closure $(\delta(q_1, 1)) = \sim(\delta(\{q_1, q_2\}, 1)) = \sim(q_1) = \{q_1, q_2\}$

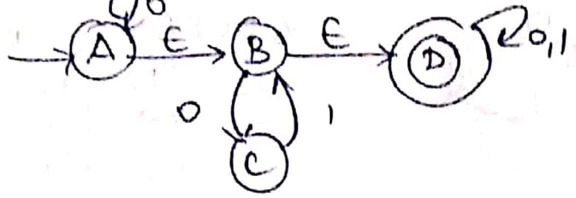
ϵ-closure $(\delta(q_1, 2)) = \sim(\delta(\{q_1, q_2\}, 2)) = \sim(q_2) = \{q_2\}$

ϵ-closure $(\delta(q_2, 0)) = \sim(\delta(\{q_2\}, 0)) = \phi$

ϵ-closure $(\delta(q_2, 1)) = \sim(\delta(\{q_2\}, 1)) = \phi$

ϵ-closure $(\delta(q_2, 2)) = \sim(\delta(\{q_2\}, 2)) = \sim(q_2) = \{q_2\}$



2. 

ϵ-closure $(A) = \{A, B, D\}$
ϵ-closure$(B) = \{B, D\}$
ϵ-closure$(C) = \{C\}$
ϵ-closure$(D) = \{D\}$

ϵ-closure $(\delta(A,0)) = \sim(\delta(\{A,B,D\},0)) = \sim(A,C,D) = \{A,B,C,D\}$

ϵ-closure $(\delta(A,1)) = \sim(\delta(\{A,B,D\},1)) = \sim(D) = \{D\}$

ϵ-closure $(\delta(B,0)) = \sim(\delta(\{A,B,D\},0)) = \sim(C,D) = \{C,D\}$

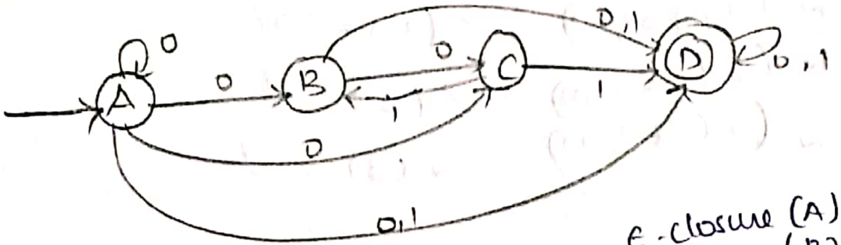
ϵ-closure $(\delta(B,1)) = \sim(\delta(\{B,D\},1)) = \sim(D) = \{D\}$

ϵ-closure $(\delta(C,0)) = \sim(\delta(\{C\},0)) = \sim(\phi) = \phi$

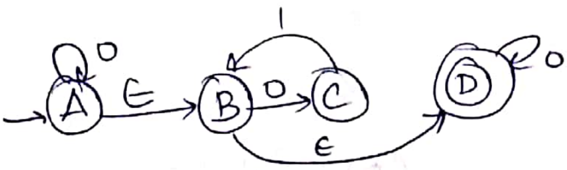
ϵ-closure $(\delta(C,1)) = \sim(\delta(\{C\},1)) = \sim(B) = \{B,D\}$

ϵ-closure $(\delta(D,0)) = \sim(\delta(\{D\},0)) = \sim(D) = \{D\}$

ϵ-closure $(\delta(D,1)) = \sim(\delta(\{D\},1)) = \sim(D) = \{D\}$



ϵ-closure $(A) = \{A, B, D\}$
$(B) = \{B, D\}$
$(C) = \{C\}$
$(D) = \{D\}$

3. 

$\sim(\delta(A,0)) = \sim(\delta(\{A,B,D\},0)) = \sim(A,C,D) = \{A,B,C,D\}$

$\sim(\delta(A,1)) = \sim(\delta(\{A,B,D\},1)) = \sim(\phi) = \phi$

$\sim(\delta(B,0)) = \sim(\delta(\{B,D\},0)) = \sim(C,D) = \cancel{\{D\}}\ \{C,D\}$

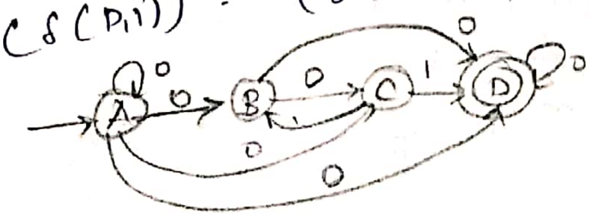
$\sim(\delta(B,1)) = \sim(\delta(\{B,D\},1)) = \sim(\phi) = \phi$

$\sim(\delta(C,0)) = \sim(\delta(\{C\},0)) = \sim(\phi) = \phi$

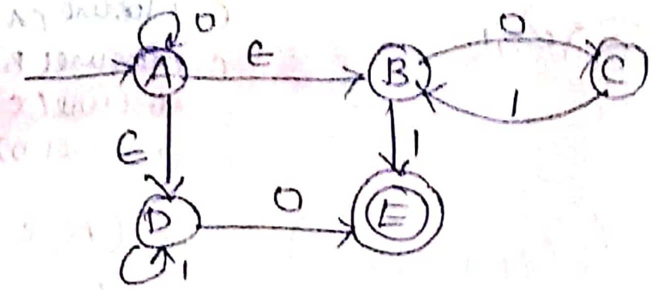
$\sim(\delta(C,1)) = \sim(\delta(\{C\},1)) = \sim(B) = \{B,D\}$

$\sim(\delta(D,0)) = \sim(\delta(\{D\},0)) = \sim(D) = \{D\}$

$\sim(\delta(D,1)) = \sim(\delta(\{D\},1)) = \sim(\phi) = \phi$



4)



ϵ-closure $(A) = \{A, B, D\}$
$\sim (B) = \{B\}$
$\sim (C) = \{C\}$
$\sim (D) = \{D\}$
$\sim (E) = \{E\}$

$\sim \delta(A,0) = \sim(\delta(\{A,B,D\},0)) = \sim(A,C,E) = \{A,B,C,D,E\}$

$\sim \delta(A,1) = \sim(\delta(\{A,B,D\},1)) = \sim(B,\cancel{E}) = \{D,E\}$

$\sim \delta(B,0) = \sim(\delta(\{B\},0)) = \sim(C) = \{C\}$

$\sim \delta(B,1) = \sim(\delta(\{B\},1)) = \sim(E) = \{E\}$

$\sim \delta(C,0) = \sim(\delta(\{C\},0)) = \sim(\phi) = \phi$

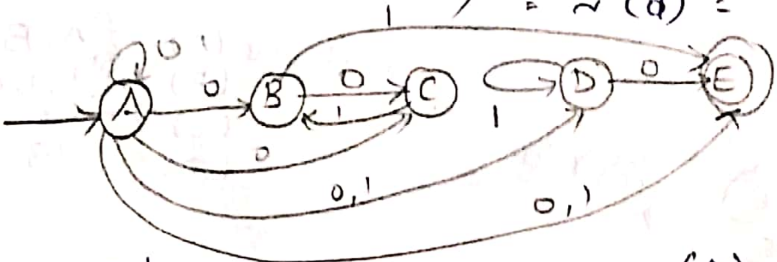
$\sim \delta(C,1) = \sim(\delta(\{C\},1)) = \sim(B) = \{B\}$

$\sim \delta(D,0) = \sim(\delta(\{D\},0)) = \sim(E) = \{E\}$

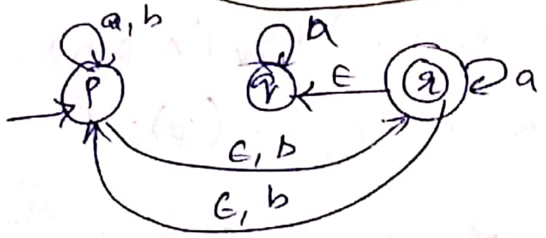
$\sim \delta(D,1) = \sim(\delta(\{D\},1)) = \sim(D) = \{D\}$

$\sim \delta(E,0) = \sim(\delta(\{E\},0)) = \sim(\phi) = \phi$

$\sim \delta(E,1) = \sim(\delta(\{E\},1)) = \sim(\phi) = \phi$



5)



$\sim (P) = \{P, q, r\}$
$\sim (q) = \{q\}$
$\sim (r) = \{P, q, r\}$

$\sim \delta(P,a) = \sim(\delta(\{P,q,r\},a)) = \sim(P,q,r) = \{P,q,r\}$

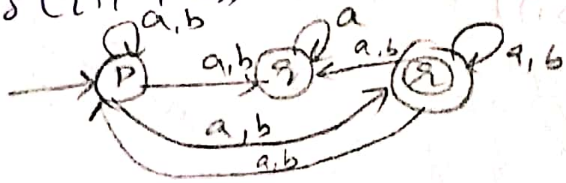
$\sim \delta(P,b) = \sim(\delta(\{P,q,r\},b)) = \sim(P,r) = \{P,q,r\}$

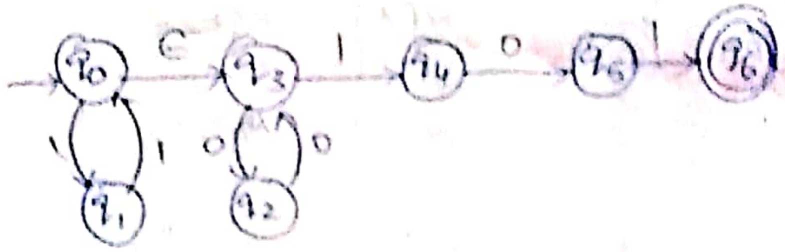
$\sim \delta(q,a) = \sim(\delta(\{q\},a)) = \sim(q) = \{q\}$

$\sim \delta(q,b) = \sim(\delta(\{q\},b)) = \sim(\phi) = \phi$

$\sim \delta(r,a) = \sim(\delta(\{P,q,r\})) = \sim(r) = \{P,q,r\}$

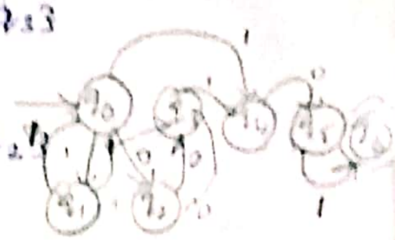
$\sim \delta(r,b) = \sim(\delta(\{P,q,r\})) = \sim(p) = \{P,q,r\}$

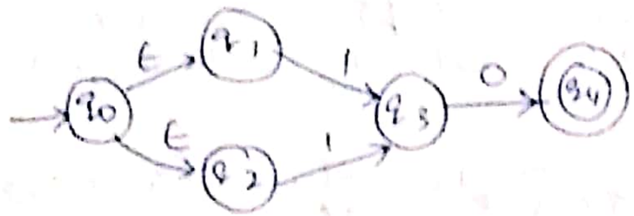


6. 

$\sim(q_0) = \{q_0, q_3\}$
$\sim(q_1) = \{q_1\}$
$\sim(q_2) = \{q_2\}$
$\sim(q_3) = \{q_3\}$
$\sim(q_4) = \{q_4\}$
$\sim(q_5) = \{q_5\}$
$\sim(q_6) = \{q_6\}$

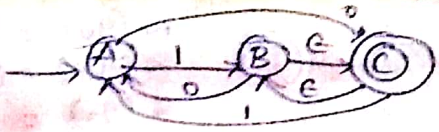
$\sim(\delta(q_0,0)) = \sim(\delta(\{q_0,q_3\},0)) = \sim(q_2) = \{q_2\}$
$\sim(\delta(q_0,1)) = \sim(\delta(\{q_0,q_3\},1)) = \sim(q_1) = \{q_1, q_4\}$
$\sim(\delta(q_1,0)) = \sim(\delta(q_1,0)) = \sim(\phi)$
$\sim(\delta(q_1,1)) = \sim(\delta(q_1,1)) = \sim(q_0) = \{q_0,q_3\}$
$\sim(\delta(q_2,0)) = \sim(\delta(q_2,0)) = \sim(q_3) = \{q_3\}$
$\sim(\delta(q_2,1)) = \sim(\delta(q_2,1)) = \sim(\phi) = \phi$
$\sim(\delta(q_3,0)) = \sim(\delta(q_3,0)) = \sim(q_2) = \phi\{q_2\}$
$\sim(\delta(q_3,1)) = \sim(\delta(q_3,1)) = \sim(q_4) = \{q_4\}$
$\sim(\delta(q_4,0)) = \sim(\delta(q_4,0)) = \sim(q_5) = \{q_5\}$
$\sim(\delta(q_4,1)) = \sim(\delta(q_4,1)) = \sim(\phi) = \phi$
$\sim(\delta(q_5,0)) = \sim(\delta(q_5,0)) = \sim(\phi) = \phi$
$\sim(\delta(q_5,1)) = \sim(\delta(q_5,1)) = \sim(q_6) = \{q_6\}$
$\sim(\delta(q_6,0)) = \sim(\delta(q_6,0)) = \sim(\phi) = \phi$
$\sim(\delta(q_6,1)) = \sim(\delta(q_6,1)) = \sim(\phi) = \phi$



7. 

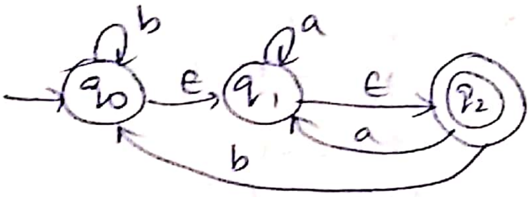
$\sim(q_0) = \{q_0, q_1, q_2\}$
$\sim(q_1) = \{q_1\}$
$\sim(q_2) = \{q_2\}$
$\sim(q_3) = \{q_3\}$
$\sim(q_4) = \{q_4\}$

$\sim(\delta(q_0,0)) = \sim(\delta(\{q_0,q_1,q_2\},0)) = \sim(\phi) = \phi$
$\sim(\delta(q_0,1)) = \sim(\delta(\{q_0,q_1,q_2\},1)) = \sim(q_3) = \{q_3\}$
$\sim(\delta(q_1,0)) = \sim(\delta(q_1,0)) = \sim(\phi) = \phi$
$\sim(\delta(q_1,1)) = \sim(\delta(q_1,1)) = \sim(q_3)\ q_3$
$\sim(\delta(q_2,0)) = \sim(\delta(q_2,0)) = \sim(\phi) = \phi$
$\sim(\delta(q_2,1)) = \sim(\delta(q_2,1)) = \sim(q_3) = q_3$
$\sim(\delta(q_3,0)) = \sim(\delta(q_3,0)) = \sim(q_4)$
$\sim(\delta(q_3,1)) = \sim(\delta(q_3,1)) = \sim(\phi) = \phi$
$\sim(\delta(q_4,0)) = \sim(\delta(q_4,0)) = \sim(\phi) = \phi$
$\sim(\delta(q_4,1)) = \sim(\delta(q_4,1)) = \sim(\phi) = \phi$

8) 

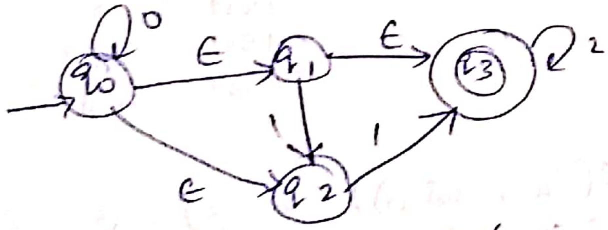
$\sim(A) = \{A\}$
$\sim(B) = \{B, C\}$
$\sim(C) = \{B, C\}$

$\sim(\delta(A,0)) = \sim\delta(\{A\},0) = \sim(C) = \{C, B\}$
$\sim(\delta(A,1)) = \sim\delta(\{A\},1) = \sim(B) = \{B, C\}$
$\sim(\delta(B,0)) = \sim\delta(\{B,C\},0) = \sim(A) = \{A\}$
$\sim(\delta(B,1)) = \sim\delta(\{B,C\},1) = \sim(A) = \{A\}$
$\sim(\delta(C,0)) = \sim\delta(\{B,C\},0) = \sim(A) = \{A\}$
$\sim(\delta(C,1)) = \sim\delta(\{B,C\},1) = \sim(A) = \{A\}$

9) 

$\sim(q_0) = \{q_0, q_1, q_2\}$
$\sim(q_1) = \{q_1, q_2\}$
$\sim(q_2) = \{q_2\}$

$\sim(\delta(q_0,a)) = \sim(\delta(\{q_0,q_1,q_2\},a)) = \sim(q_1) = \{q_1, q_2\}$
$\sim(\delta(q_0,b)) = \sim(\delta(\{q_0,q_1,q_2\},b)) = \sim(q_0) = \{q_0, q_1, q_2\}$
$\sim(\delta(q_1,a)) = \sim(\delta(\{q_1,q_2\},a)) = \sim(q_1) = \{q_1, q_2\}$
$\sim(\delta(q_1,b)) = \sim(\delta(\{q_1,q_2\},a)) = \sim(q_0) = \{q_0, q_1, q_2\}$
$\sim(\delta(q_2,a)) = \sim(\delta(\{q_2\},a)) = \sim(q_1) = \{q_1, q_2\}$
$\sim(\delta(q_2,b)) = \sim(\delta(\{q_2\},b)) = \sim(q_0) = \{q_0, q_1, q_2\}$

10) 

$\sim(q_0) = \{q_0, q_1, q_2, q_3\}$
$\sim(q_1) = \{q_1, q_3\}$
$\sim(q_2) = \{q_2\}$
$\sim(q_3) = \{q_3\}$

$q_0, 0 = \{q_0, q_1, q_2, q_3\}$
$q_0, 1 = \{q_2, q_3\}$
$q_0, 2 = \phi\ \{q_3\}$

$q_1, 0 = \phi$
$q_1, 1 = \{q_2\}$
$q_1, 2 = \phi\ \{q_3\}$

$q_2, 0 = \phi$
$q_2, 1 = \{q_3\}$
$q_2, 2 = \phi$

$q_3, 0 = \phi$
$q_3, 1 = \phi$
$q_3, 2 = \{q_3\}$

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
$*q_3$	q_3	q_0
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	q_6
q_7	q_6	q_3

$q_0 = q_4$
$q_1 = q_5$
$q_2 = q_4$

A language L is accepted by some E-NFA iff L is accepted by DFA [DEC [JUNE

BASIS:

If $|w| = 0$ then $w = \epsilon$

Let $D = \{Q_D, \Sigma, \delta_D, q_D, F_D)$

$E = \{Q_E, \Sigma, \delta_E, q_0, F_E)$

We know that $\hat{\delta}_E(q_0, \epsilon) = \epsilon\text{-closure}(q_0)$

we also know that $q_D = \epsilon\text{-closure}(q_0)$

for a DFA, $\hat{\delta}_D(p, \epsilon) = p$ for any state p

$\hat{\delta}_D(q_D, \epsilon) = \epsilon\text{-closure}(q_0)$

Thus, $\hat{\delta}_E(q_0, \epsilon) = \hat{\delta}_D(q_D, \epsilon)$

INDUCTION:

Suppose $w = xa$ where a is the final symbol of w and assume that statement holds for x that is

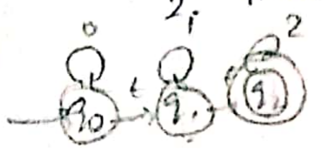
$\hat{\delta}_E(q_0, x) = \hat{\delta}_D(q_0, x)$

Let both these states be $\{p_1, p_2, p_3, \ldots p_k\}$

By the definition of $\hat{\delta}$ for NFA-ϵ, we compute $\hat{\delta}_E(q_0, w)$

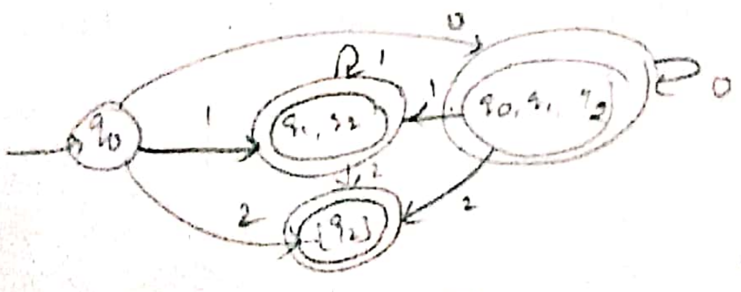
1. Let $\{r_1, r_2, \ldots r_m\}$ be $\bigcup_{i=1}^{k} \delta_E(p_i, a)$

2. Then $\hat{\delta}_E(q_0, w) = \bigcup_{j=1}^{m} \epsilon\text{-Closure}(r_j)$



$\hat{\delta}_D(q_D, w) = \delta_D(\{p_1, p_2, \ldots p_k\}, a) = \hat{\delta}_E(q_0, w)$

Thus $\hat{\delta}_D(q_D, w) = \hat{\delta}_E(q_0, w)$



NFA-ϵ :

$\epsilon\text{-closure}(\hat{\delta}(q_0, 0)) = \{q_0, q_1, q_2\}$

DFA

$(q_0, 0) = [q_0, q_1, q_2]$

also of $\underline{01}$ $\begin{cases} \text{NFA-}\epsilon - \{q_1, q_2\} \\ \text{DFA } [q_1, q_2] \end{cases}$

Scanned with CamScanner

DFA

CASE 1:

$q_0 \to 0 \to q_2$, $q_0 \to 1 \to q_3$

$q_1 \to 0 \to q_2$, $q_1 \to 1 \to q_3$

	0	1
q_0	q_2	q_3
q_1	q_2	q_3

CASE 2:

$q_0 \to 0, 1 \to q_2$

$q_1 \to 0, 1 \to q_3$

	0	1
q_0	q_2	q_2
q_1	q_3	q_3

CASE 3:

$q_0 \to 0 \to q_3$

$q_1 \to 0 \to q_4$

	0	1
q_0	q_3	q_3
q_1	q_4	q_4

CASE 4:

$q_0 \to 0 \to q_2$

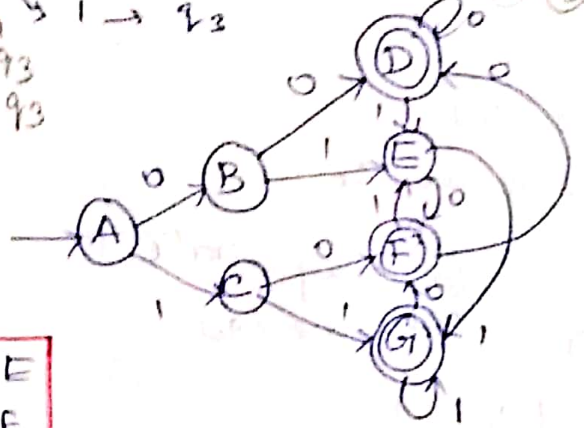
$q_1 \to 0 \to q_4$

$q_2 \equiv q_4$

then $q_0 \equiv q_1$

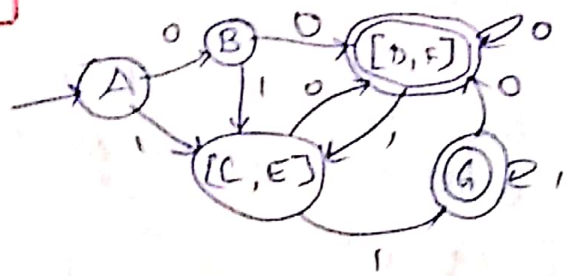
	0
q_0	q_2
q_1	q_4

1.



	A	B	C	D	E	F
B	x					
C	x	x				
D	x	x	x			
E	x	x	✓	x		
F	x	x	x	✓	x	
G	x	x	x	x	x	x

$\boxed{C \equiv E \quad D \equiv F}$



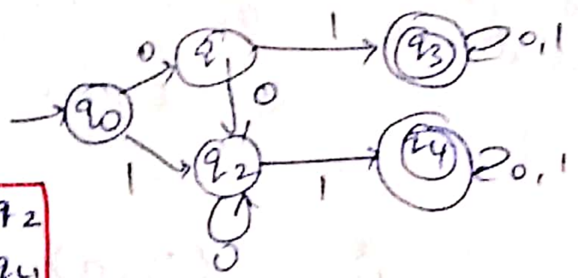
$q_0 \to 0 \to q_1$

$q_2 \to 0 \to q_1$

$q_0 \to 0 \to q_3$

$q_2 \to 0 \to q_3$

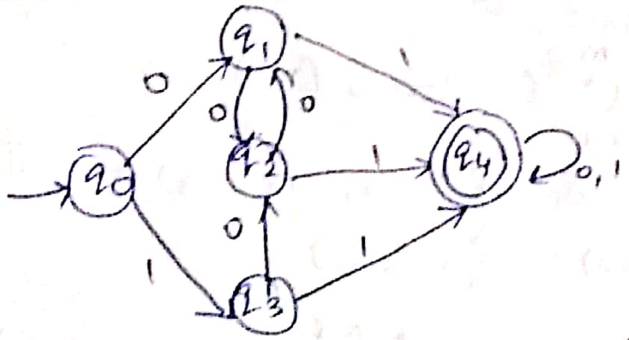
2.



	q_0	q_1	q_2	q_3
q_1	x			
q_2	x	✓		
q_3	x	x	x	
q_4	x	x	x	✓

$\boxed{q_1 \equiv q_2 \quad q_3 \equiv q_4}$

3.

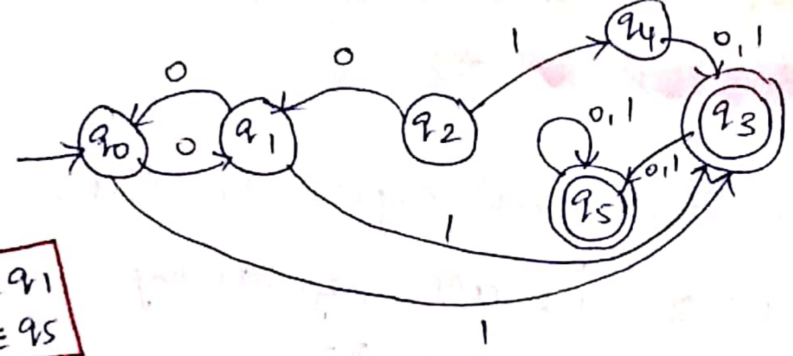


	q_0	q_1	q_2	q_3
q_1	x			
q_2	x	✓		
q_3	x	✓	✓	
q_4	x	x	x	x

$\boxed{q_1 \equiv q_2 \equiv q_3}$

1

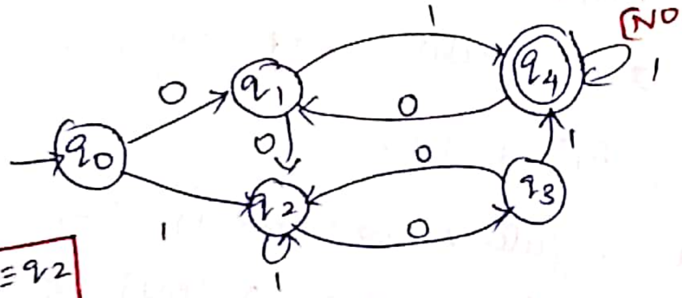
4.



q_1	✓				
q_2	X	X			
q_3	X	X	X		
q_4	X	X	X	X	
q_5	X	X	X	✓	X
	q_0	q_1	q_2	q_3	q_4

$\boxed{\begin{array}{l} q_0 \equiv q_1 \\ q_3 \equiv q_5 \end{array}}$

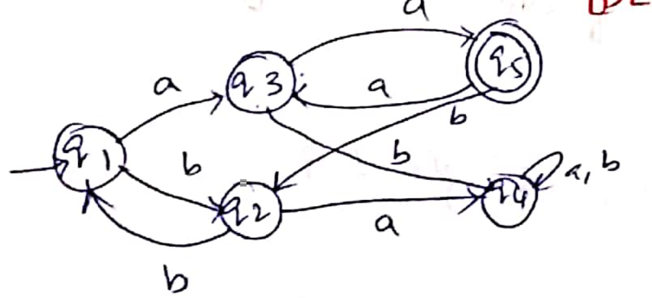
5. [NOV 2014]



q_1	X			
q_2	✓	X		
q_3	X	✓	X	
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

$\boxed{\begin{array}{l} q_0 \equiv q_2 \\ q_1 \equiv q_3 \end{array}}$

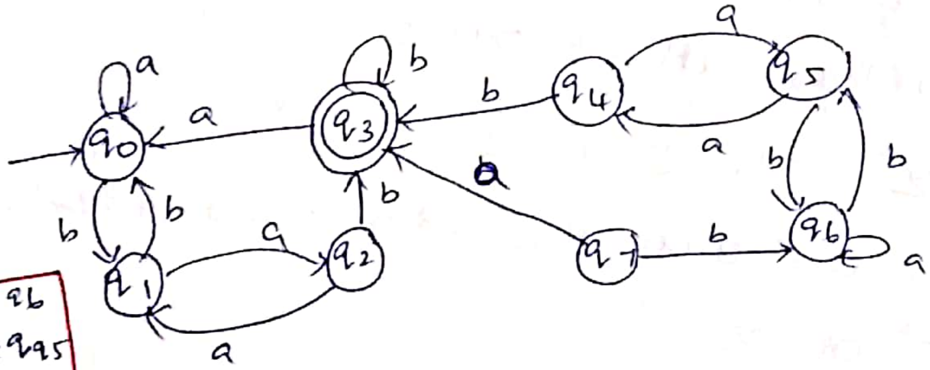
6. [DEC 2015]



cannot be minimized

2

7. [JUNE 2016]



q_1	X						
q_2	X	X					
q_3	X	X	X				
q_4	X	X	✓	X			
q_5	X	✓	X	X	X		
q_6	✓	X	X	X	X	X	
q_7	X	X	X	X	X	X	X
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

$\boxed{\begin{array}{l} q_0 \equiv q_6 \\ q_1 \equiv q_5 \\ q_2 \equiv q_4 \end{array}}$

REGULAR EXPRESSIONS & LANGUAGES

Equivalent to NFA-ϵ. Algebric description

3 main expressions: Union, concatenation, closure (kleene)

Informal: Languages that can be accepted by finite automata are the same as regular languages which can be represented by formulas of RE.

FORMAL:

Let Σ be the given alphabet, then

(i) ϕ, ϵ and $a \in \Sigma$ are all regular expressions, these are called primitive RE

(ii) If r_1 and r_2 are regular expression, so are $r_1 + r_2$, $r_1 . r_2$, r_1^* and (r_1)

(iii) A string is a regular expression iff it can be derived from the primitive RE by the finite no. of application of the rules in (ii)

1) L = {w/w contains a single 1}

$0^* 1 0^*$

2) L = {w/w has atleast one 1}

$\Sigma^* 1 \Sigma^*$ (or) $(0+1)^* 1 (0+1)^*$ $\qquad 0^* 1^+ 0^*$

0101 - string

3) L = {w/w contains the string 001 as a substring}

$\Sigma^* 001 \Sigma^*$

4) L = {w/ every 0 in w is followed by atleast one 1}

10 01 101

101

10111111

111 1011

1011 011 0111

$\cancel{(01)^* (1)^*}$ (or) $\cancel{1^*(01)^*}$ or $\boxed{1^* (01^+)^*}$

5) L = {w/w is a string of even lengths}

$(\Sigma\Sigma)^*$

6) L = {w/ the length of w is a multiple of 3}

$(\Sigma\Sigma\Sigma)^*$

7) $L = \{\epsilon, 0, 1, 01\}$

$(\epsilon+0)(\epsilon+1)$ or $\epsilon + 0 + 1 + 01$

8) $L = \{01, 10\}$

$(01+10)$

11) Atleast 2 consecutive 0's

$(0+1)^* 00 (0+1)^*$

12) All strings of 0's & 1's

$(0+1)^*$

13) Any no. of 0's followed by 1's & 2's $0^* 1^* 2^*$

14) min one occurance

$0^+ 1^+ 2^+$ (or) $00^* 11^* 22^*$

9) $L = \{w \mid w$ starts and ends with the same symbol$\}$

$0\Sigma^* 0 + 1\Sigma^* 1 + 0 + 1$

10) $L = \{x \in \{0,1\}^* \mid x$ ends with 1 and doesn't contain the substring $00\}$

$(1+01)^+$

THOMSON'S CONSTRUCTION RE to NFA-ϵ

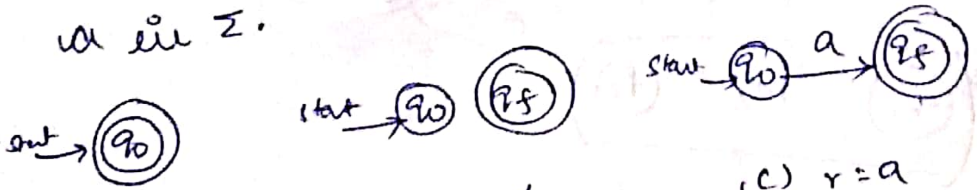
Let R be a regular expression then, there exists an NFA-ϵ transition that accept $L(r)$

PROOF:

We show by induction on the no. of operators in the regular expression r then there is a NFA with ϵ transition having one final state and ~~[illegible]~~ no transitions out of this final state such that $L(M) = L(r)$

BASIS: (zero operator)

The expression r must be ϵ, ϕ or a for some a in Σ.



(a) $r = \epsilon$ (b) $r = \phi$ (c) $r = a$

INDUCTION (≥ 1 operation)

CASE 1: $r = r_1 + r_2$

$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, \{f_1\})$

$M_2 = (Q_2, \Sigma_2, \delta_2, q_2, \{f_2\})$

$L(M_1) = L(r_1)$ and $L(M_2) = L(r_2)$

We may assume Q_1 and Q_2 are disjoint.

Let q_0 be a new initial state and f_0 a new final state

construct

$M = (Q_1 \cup Q_2 \cup \{q_0, f_0\}, \Sigma_1 \cup \Sigma_2, \delta, q_0, \{f_0\})$

we define δ by

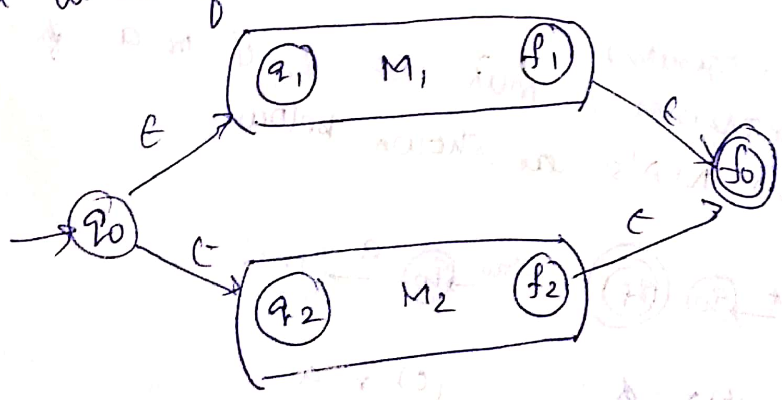
(i) $\delta(q_0, \epsilon) = \{q_1, q_2\}$

(ii) $\delta(q, a) = \delta_1(q, a)$ for q in $Q_1 - \{f_1\}$ and a in $\Sigma_1 \cup \{\epsilon\}$

(iii) $\delta(q, a) = \delta_2(q, a)$ for q in $Q_2 - \{f_2\}$ and a in $\Sigma_2 \cup \{\epsilon\}$

(iv) $\delta(f_1, \epsilon) = \delta_1(f_2, \epsilon) = \{f_0\}$

There is a path labelled x in M from q_0 to f_0 iff there is a path labelled x in M_1 from q_1 to f_1 (or) path in M_2 from q_2 to f_2 hence $L(M) = L(M_1) \cup L(M_2)$



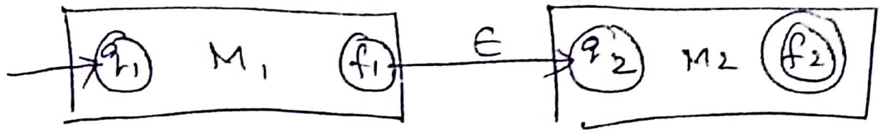
CASE : 2 $r = r_1 r_2$. Let M_1 and M_2 be as in case 1 and construct $M = (Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \delta, \{q_1\}, \{f_2\})$

where δ is given by

(i) $\delta(q, a) = \delta_1(q, a)$ for q in $Q_1 - \{f_1\}$ and a in $\Sigma_1 \cup \{\epsilon\}$

(ii) $\delta(f_1, \epsilon) = \{q_2\}$

(iii) $\delta(q, a) = \delta_2(q, a)$ for q in Q_2 and a in $\Sigma_2 \cup \{\epsilon\}$



Every path in M from q_1 to f_2 is a path labeled by some string x from q_1 to f_1 followed by the edge from f_1 to q_2 labeled ϵ, followed by a path labeled by some string y from q_2 to f_2. Thus $L(M) = \{xy \mid x$ is in $L(M_1)$ and y is in $L(M_2)\}$ and $L(M) = L(M_1)L(M_2)$ as desired.

CASE : 3 $r = r_1^*$

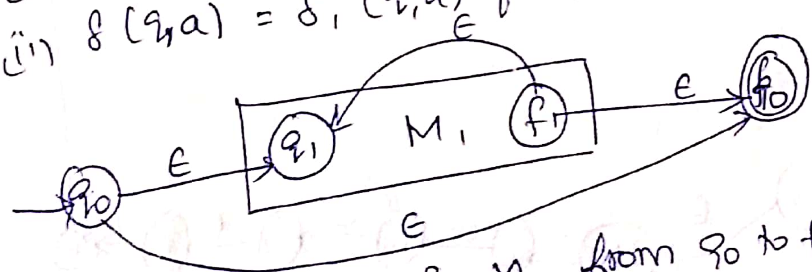
Let $M_1 = (Q_1, \Sigma_1, \delta_1, q_1, \{f_1\})$ and $L(M_1) = r_1$

construct $M = (Q_1 \cup \{q_0, f_0\}, \Sigma_1, \delta, q_0, \{f_0\})$

where δ is defined by

(i) $\delta(q_0, \epsilon) = \delta(f_1, \epsilon) = \{q_1, f_0\}$

(ii) $\delta(q, a) = \delta_1(q, a)$ for q in $Q_1 - \{f_1\}$ and a in $\Sigma_1 \cup \{\epsilon\}$

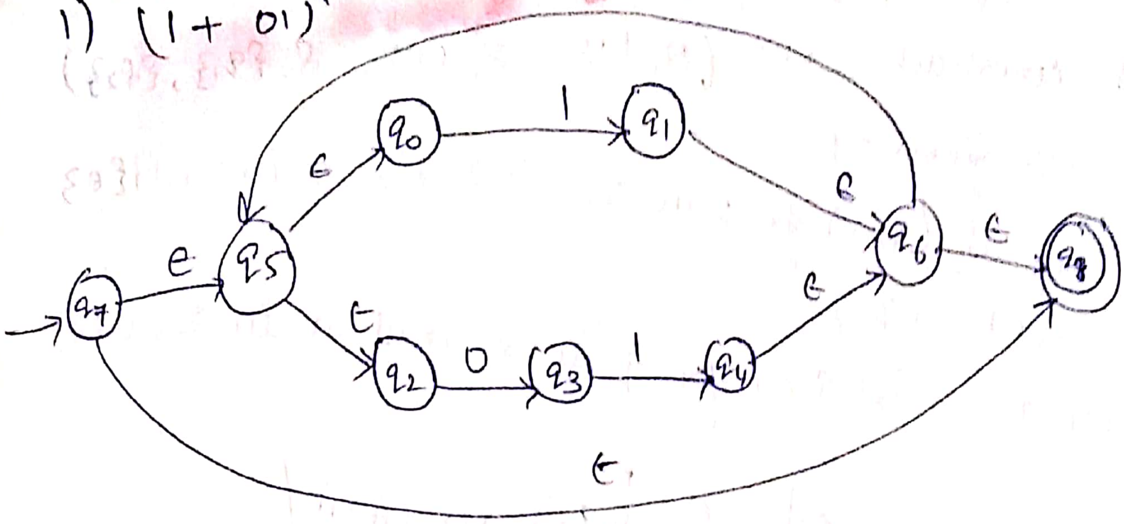


There is a path in M from q_0 to f_0 labeled x iff $x = x_1 x_2 \ldots x_j$ for some $j \geq 0$ such that x_j is in $L(M_1)$

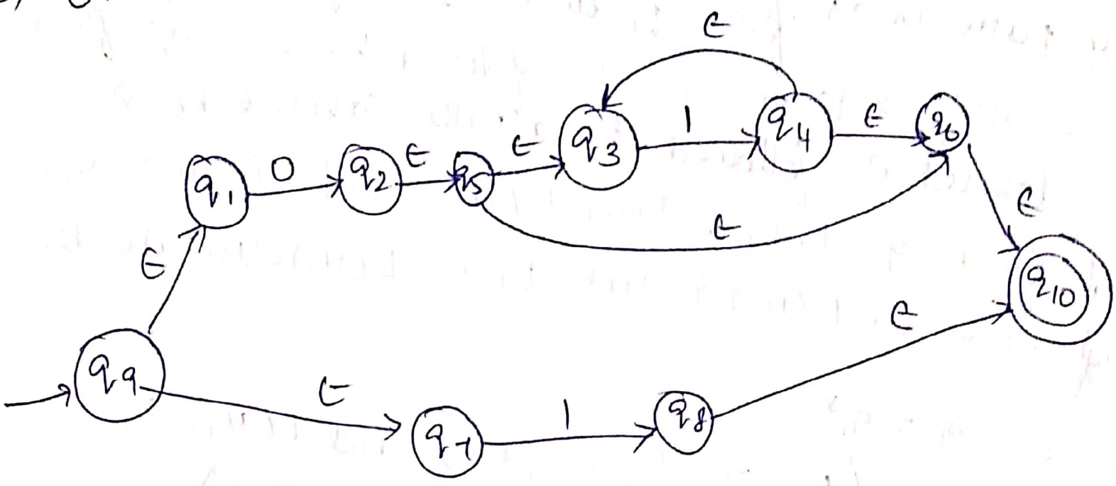
Hence $L(M) = L(M_1)^*$

Scanned with CamScanner

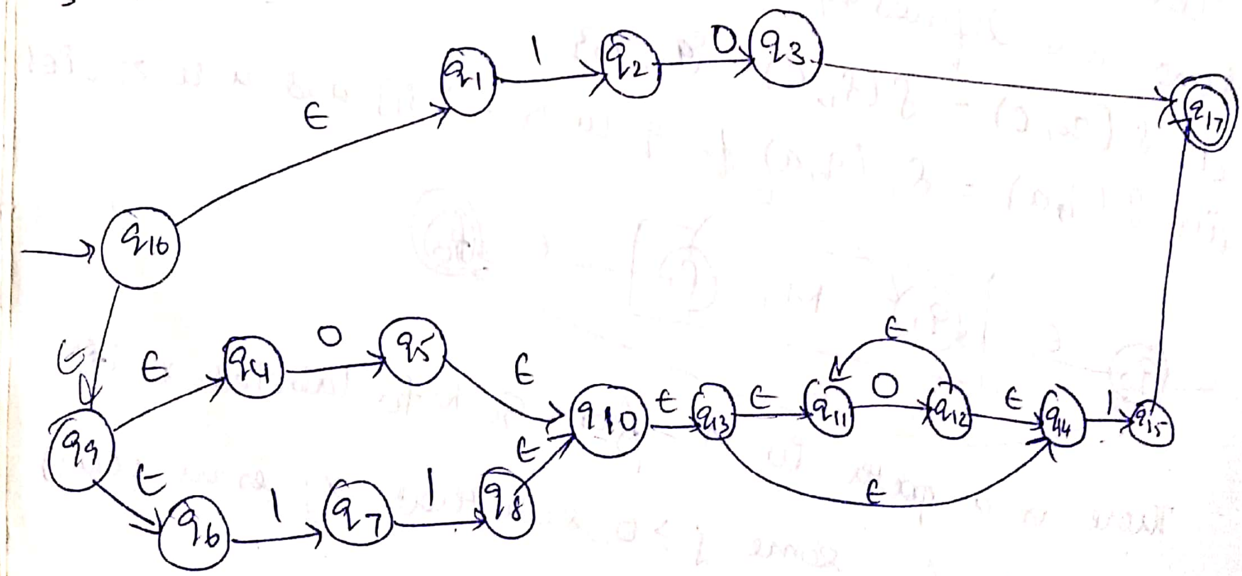
1) $(1+01)^*$



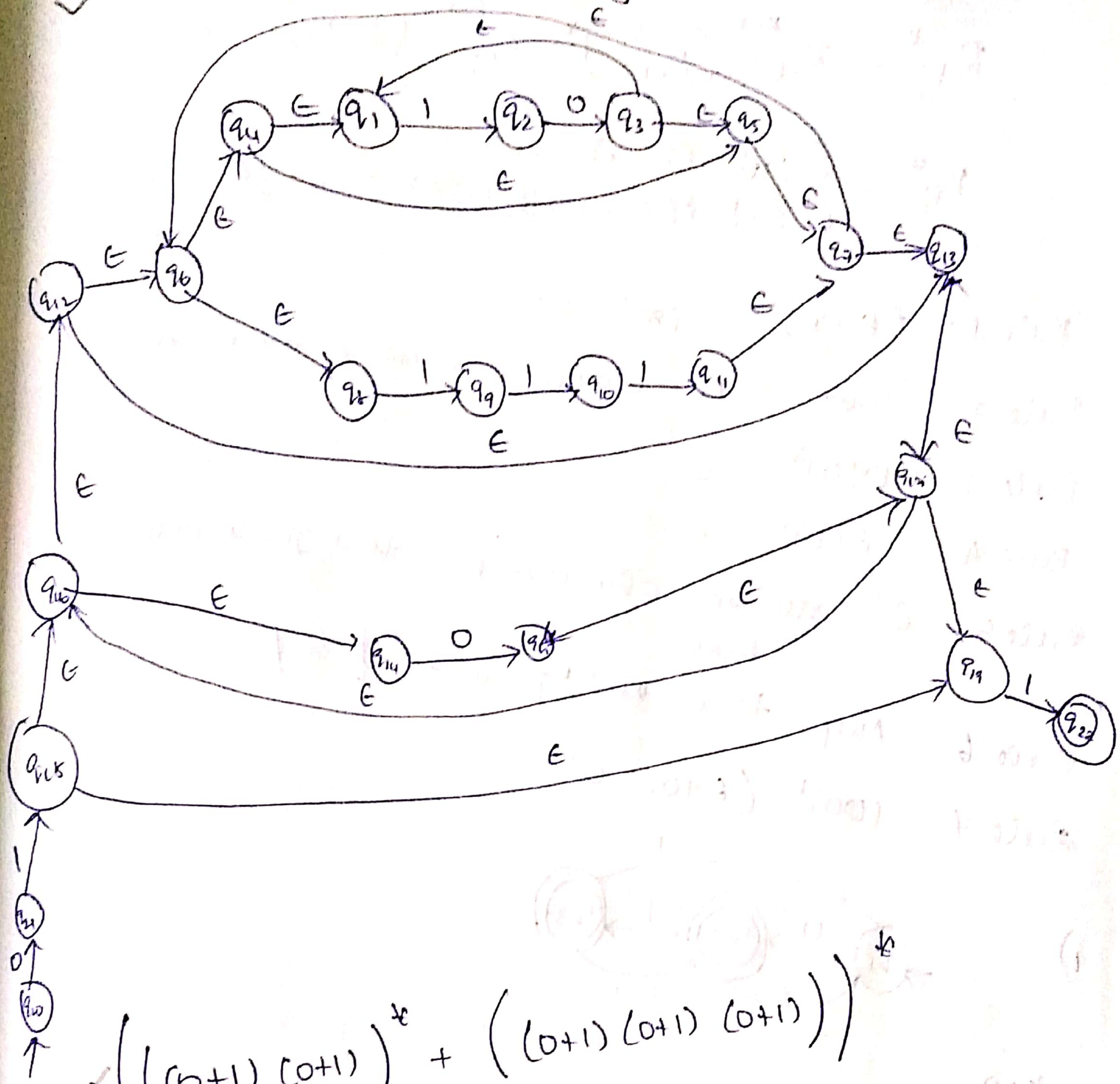
2) $01^* + 1$



3. $10 + (0+11)\, 0^*\, 1$



4. $01\left[\left((10)^* + 111 \right)^* + 0 \right]^* 1$



5) $\left(\left((0+1)(0+1) \right)^* + \left((0+1)(0+1)(0+1) \right) \right)^*$

6) $(11+0)^* (00+1)^*$

REGULAR EXPRESSION

$$R_{ij}^{k} = R_{ik}^{k-1} (R_{kk}^{k-1})^{*} R_{kj}^{k-1} \cup R_{ij}^{k-1}$$

$$R_{ij}^{0} = \begin{cases} \{a \mid \delta(q_i, a) = q_j\} & \text{if } i \neq j \\ \{a \mid \delta(q_i, a) = q_j \cup \{\epsilon\}\} & \text{if } i = j \end{cases}$$

Rule 1: $(\epsilon + r)^{*} = (r)^{*} = r^{*}$

Rule 2: $00+0 = 00$, $0w+0 = 0w$ iff r_{ij} are equal

Rule 3: $0(0)^{*} = 0^{*}$

Rule 4: $\epsilon(\epsilon)^{*} = \epsilon^{*}$

Rule 5: ϵ^{*} can be eliminated when given with a term.

Rule 6: Any term multiplied by ϕ is ϕ

Rule 7: $(00)^{*}(\epsilon+0) = 0^{*}$.

1) 

$k=0$

$r_{11}^{0} = \epsilon$ $r_{21}^{0} = 0$ $r_{31}^{0} = \phi$

$r_{12}^{0} = 0$ $r_{22}^{0} = \epsilon$ $r_{32}^{0} = 0+1$

$r_{13}^{0} = 1$ $r_{23}^{0} = 1$ $r_{33}^{0} = \epsilon$

$k=1$

$$r_{11}^{1} = r_{11}^{0} (r_{11}^{0})^{*} r_{11}^{0} + \cancel{R_{11}^{0}}$$

$$= \epsilon(\epsilon)^{*}\epsilon \quad \text{(Rule 4)} \quad = \epsilon(\epsilon)^{*}$$

$$= \epsilon^{*}$$

$r_{12}^1 = r_{11}^0 (r_{11}^0)^* r_{12}^0 + \cancel{r_{12}^0}$

$= \epsilon (\epsilon)^* 0$

$= \epsilon^* 0$

$= 0$

$r_{13}^1 = r_{11}^0 (r_{11}^0)^* r_{13}^0 + \cancel{r_{13}^0}$

$= \epsilon (\epsilon)^* 1$

$= 1$

$r_{21}^1 = r_{21}^0 (r_{11}^0)^* r_{11}^0 + \cancel{r_{21}^0} = 0(\epsilon)^* \epsilon$

$= 0(\epsilon)^* \epsilon$

$= 0$

$r_{22}^1 = r_{21}^0 (r_{11}^0)^* r_{12}^0 + r_{22}^0$

$= 0(\epsilon)^* 0 + \epsilon$

$= 00 + \epsilon$

$= 0(00+\epsilon)^* 1(0+\epsilon) + 1$

$: 0(00)^*(0+\epsilon)1$

$: 00^* 1 = 0^* 1$

$r_{23}^1 = r_{21}^0 (r_{11}^0)^* r_{13}^0 + r_{23}^0 = 0(\epsilon)^* 1 + 1 = 01 + 1$

$r_{31}^1 = r_{31}^0 (r_{11}^0)^* r_{11}^0 + \cancel{r_{31}^0} = \phi (\epsilon)^* \epsilon = \phi$

$r_{32}^1 = r_{31}^0 (r_{11}^0)^* r_{12}^0 + r_{32}^0 = \phi(\epsilon)^* 0 + (0+1) = 0+1$

$r_{33}^1 = r_{31}^0 (r_{11}^0)^* r_{13}^0 + r_{33}^0 = \phi(\epsilon)^* 1 + \epsilon = \epsilon$

$\underline{k=2}$

$r_{11}^2 = r_{12}^1 (r_{22}^1)^* r_{21}^1 + \cancel{r_{12}^1} = 0(00+\epsilon)^* 0 = (00)^*$

$r_{12}^2 = r_{12}^1 (r_{22}^1)^* r_{22}^1 + \cancel{r_{12}^1} = 0(00+\epsilon)^* (00+\epsilon) = 0(00)^*$

$r_{13}^2 = r_{12}^1 (r_{22}^1)^* r_{23}^1 + r_{13}^1 = 0(00+\epsilon)^*(01+1) + 1 = 0(00+\epsilon)^* 1(0+\epsilon)$ $= 0^* 1$

$r_{21}^2 = r_{22}^1 (r_{22}^1)^* r_{21}^1 + \cancel{r_{21}^1} = (00+\epsilon)(00+\epsilon)^* 0 = (00)^* 0$

$r_{22}^2 = r_{22}^1 (r_{22}^1)^* r_{22}^1 + \cancel{r_{22}^1} = (00+\epsilon)(00+\epsilon)^* = (00)^*$

$r_{23}^2 = r_{22}^1 (r_{22}^1)^* r_{23}^1 + \cancel{r_{23}^1} = (00+\epsilon)(00+\epsilon)^*(01+1) = 0^* 1$

$r_{31}^2 = r_{32}^1 (r_{22}^1)^* r_{21}^1 + r_{31}^1 = (0+1)(00+\epsilon)^* 0 + \phi = (0+1)(00)^* 0$

$r_{32}^2 = r_{32}^1 (r_{22}^1)^* r_{22}^1 + \cancel{r_{32}^1} = (0+1)(00+\epsilon)^*(0+1) = (0+1)(00)^*$

$r_{33}^2 = r_{32}^1 (r_{22}^1)^* r_{23}^1 + r_{33}^1 = (0+1)(00+\epsilon)^*(01+1) + \epsilon = (0+1)0^* 1$

$r_{12}^3 + r_{13}^3$

$r_{12}^3 = r_{13}^2 (r_{33}^2)^* r_{32}^2 + r_{12}^2$

$= 0^*1(\epsilon + (0+1)0^*1)^*(0+1)(00)^* + 0(00)^*$

$= 0^*1((0+1)0^*1)^*(0+1)(00)^* + 0(00)^*$

$r_{13}^3 = r_{13}^2 (r_{33}^2)^* r_{33}^2 + \cancel{r_{13}^2}$

$= 0^*1(\epsilon+(0+1)0^*1)^*(\epsilon+(0+1)0^*1)$

$= 0^*1((0+1)0^*1)^*$

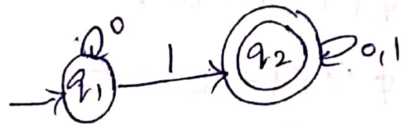
Hence

$r_{12}^3 + r_{13}^3 = 0^*1((0+1)0^*1)^*(0+1)(00)^* + 0(00)^* +$

$0^*1((0+1)0^*1))^*$

$= 0^*1((0+1)0^*1)^*(0+1)(00)^* + 0(00)^*$

2.



$\underline{k=0}$ $r_{11}^0 = \epsilon+0$ $r_{12}^0=1$ $r_{21}^0=\phi$ $r_{22}^0 = \epsilon+0+1$

$\underline{k=1}$

$r_{11}^1 = r_{11}^0(r_{11}^0)^* r_{11}^0 + \cancel{r_{11}^0} = \epsilon+0(\epsilon+0)^*\epsilon+0 = 0^*$

$r_{12}^1 = r_{11}^0(r_{11}^0)^* r_{12}^0 + \cancel{r_{12}^0} = \epsilon+0(\epsilon+0)^*1 = 0^*1$

$r_{21}^1 = r_{21}^0(r_{11}^0)^* r_{11}^0 + r_{21}^0 = \phi$

$r_{22}^1 = r_{21}^0(r_{11}^0)^* r_{12}^0 + r_{22}^0 = \phi + (\epsilon+0+1) = \epsilon+0+1$

$\underline{k=2}$

$r_{12}^2 = r_{12}^1(r_{22}^1)^* r_{22}^1 + \cancel{r_{12}^1} = (0^*1)(\epsilon+0+1)^*(\epsilon+0+1)$

$=(0^*1)(0+1)^*$

3. 

$k=0$ $\quad r_{11}^0 = \epsilon + b \quad r_{12}^0 = a \quad r_{21}^0 = a \quad r_{22}^0 = \epsilon + b$

$k=1$

$r_{11}^1 = r_{11}^0 (r_{11}^0)^* r_{11}^0 + r_{11}^0 = (\epsilon+b)((\epsilon+b))^*(\epsilon+b) = b^*$

$r_{12}^1 = r_{11}^0 (r_{11}^0)^* r_{12}^0 + r_{12}^0 = (\epsilon+b)(\epsilon+b)^* a = b^* a$

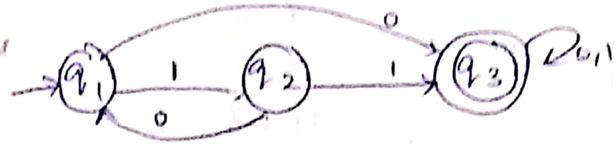
$r_{21}^1 = r_{21}^0 (r_{11}^0)^* r_{11}^0 + r_{21}^0 = a(\epsilon+b)^*(\epsilon+b) = ab^*$

$r_{22}^1 = r_{21}^0 (r_{11}^0)^* r_{12}^0 + r_{22}^0 = a(\epsilon+b)^* a + (\epsilon+b) = ab^*a + (\epsilon+b)$

$k=2$

$r_{11}^2 = r_{12}^1 (r_{22}^1)^* r_{21}^1 + r_{11}^1 = b^* a (ab^*a + (\epsilon+b))^* ab^* + b^*$

$= b^* a (ab^*a + b)^* ab^* + b^*$

4. 

$r_{21}^0 = \phi \quad r_{32}^0 = \phi \quad r_{33}^0 = \epsilon + 0 + 1$

$k=0$ $\quad r_{11}^0 = \epsilon \quad r_{12}^0 = 1 \quad r_{13}^0 = 0 \quad r_{21}^0 = 0 \quad r_{22}^0 = \epsilon \quad r_{23}^0 = 1$

$k=1$

$r_{11}^1 = r_{11}^0 (r_{11}^0)^* r_{11}^0 + \cancel{r_{11}^0} = \epsilon(\epsilon)^*\epsilon = \epsilon^*\epsilon = \epsilon$

$r_{12}^1 = r_{11}^0 (r_{11}^0)^* r_{12}^0 + \cancel{r_{12}^0} = \epsilon(\epsilon)^* 1 = 1$

$r_{13}^1 = r_{11}^0 (r_{11}^0)^* r_{13}^0 + \cancel{r_{13}^0} = \epsilon(\epsilon)^* 0 = 0$

$r_{21}^1 = r_{21}^0 (r_{11}^0)^* r_{11}^0 + \cancel{r_{21}^0} = 0(\epsilon)^*\epsilon = 0$

$r_{22}^1 = r_{21}^0 (r_{11}^0)^* r_{12}^0 + r_{22}^0 = 0(\epsilon)^* 1 + \epsilon = 01 + \epsilon$

$r_{23}^1 = r_{21}^0 (r_{11}^0)^* r_{13}^0 + r_{23}^0 = 0(\epsilon)^* 0 + 1 = 00 + 1$

$r_{31}^1 = r_{31}^0 (r_{11}^0)^* r_{11}^0 + \cancel{r_{31}^0} = \phi$

$r_{32}^1 = r_{31}^0 (r_{11}^0)^* r_{12}^0 + r_{32}^0 = \phi$

$r_{33}^1 = r_{31}^0 (r_{11}^0)^* r_{13}^0 + r_{33}^0 = \phi + (\epsilon + 0 + 1) = \epsilon + 0 + 1$

$k=2$

$r_{11}^2 = r_{12}^1 (r_{22}^1)^* r_{21}^1 + r_{11}^1 = 1(01+\epsilon)^* 0 + \epsilon = 1(01)^*0 + \epsilon$

$r_{12}^2 = r_{12}^1 (r_{22}^1)^* r_{22}^1 + \cancel{r_{12}^1} = 1(01+\epsilon)^*(01+\epsilon) = 1(01)^*$

$r_{13}^2 = r_{12}^1 (r_{22}^1)^* r_{23}^1 + r_{13}^1 = 1(01+\epsilon)^*(00+1) + 0 = 1(01)^*(00+1)+0$

$r_{21}^2 = r_{22}^1 (r_{22}^1)^* r_{21}^1 + \cancel{r_{21}^1} = (01+\epsilon)(01+\epsilon)^* 0 = (01)^*0$

$r_{22}^2 = r_{22}^1 (r_{22}^1)^* r_{22}^1 + \cancel{r_{22}^1} = (01+\epsilon)(01+\epsilon)^*(01+\epsilon) = (01)^*$

$r_{23}^2 = r_{22}^1 (r_{22}^1)^* r_{23}^1 + \cancel{r_{23}^1} = (01+\epsilon)(01+\epsilon)^*(00+1) = (01)^*(00+1)$

$r_{31}^2 = r_{32}^1 (r_{22}^1)^* r_{21}^1 + r_{31}^1 = \phi(01+\epsilon)^* 0 + \phi = \phi$

$r_{32}^2 = r_{32}^1 (r_{22}^1)^* r_{22}^1 + \cancel{r_{32}^1} = \phi(01+\epsilon)^*(01+\epsilon)^* = \phi$

$r_{33}^2 = r_{32}^1 (r_{22}^1)^* r_{23}^1 + r_{33}^1 = \phi\, \cdots + (\epsilon+0+1) = (\epsilon+0+1)$

Scanned with CamScanner

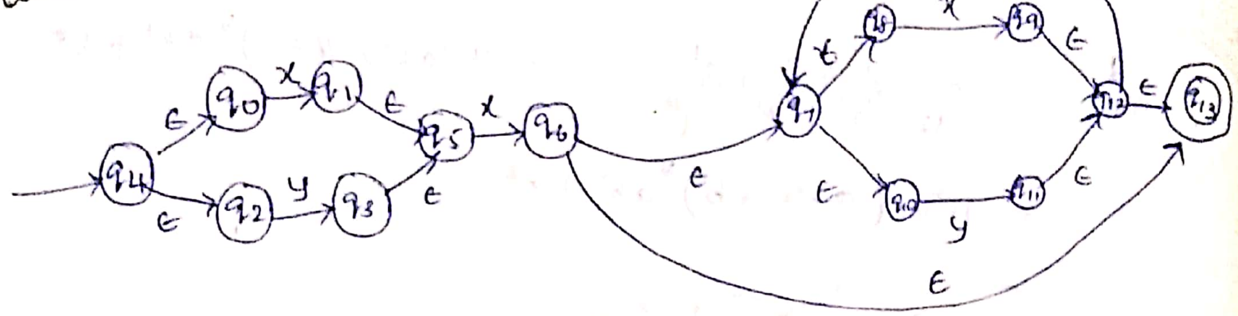
$\underline{k=3}$

$r_{13}^{3} = r_{13}^{2}(r_{33}^{2})^{*} r_{33}^{2} + \cancel{r_{13}^{2}}$

$= 1(01)^{*}(00+1)+0 \quad (\epsilon+00+1)^{*}(\epsilon+0+1)$

$= 1(01)^{*}(00+1)+0(0+1)^{*}$

Construct minimized DFA from RE $(x+y)x(x+y)^{*}$ and trace for the string $w = xxyx$



ϵ-closure $(q_4) = \{q_4, q_0, q_2\} = A$

$\delta(A,x) = \delta(\{4,0,2\},x) = \epsilon\text{-closure}(1) = \{1,5\} = B$

$\delta(A,y) = \delta(\{4,0,2\},y) = \epsilon\text{-closure}(3) = \{3,5\} = C$

$\delta(B,x) = \delta(\{1,5\},x) = \epsilon\text{-closure}(6) = \{6,7,8,10,13\} = D^{*}$

$\delta(B,y) = \delta(\{1,5\},y) = \epsilon\text{-closure}(\phi) = E$

$\delta(C,x) = \delta(\{3,5\},x) = \epsilon\text{-closure}(6) = \{6,7,8,10,13\} = D^{*}$

$\delta(C,y) = \delta(\{3,5\},y) = \epsilon\text{-closure}(\phi) = E$

$\delta(D,x) = \delta(\{6,7,8,10,13\},x) = \sim(9) = \{9,12,13,7,8,10\} = F^{*}$

$\delta(D,y) = \delta(\{6,7,8,10,13\},y) = \sim(11) = \{11,12,13,7,8,10\} = G^{*}$

$\delta(E,x) = \delta(\phi,x) = E$

$\delta(E,y) = \delta(\phi,y) = E$

$\delta(F,x) = \delta(\{7,8,9,10,12,13\},x) = \sim(9) = F^{*}$

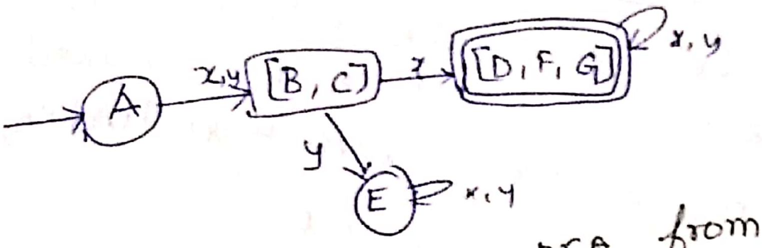
$\delta(F,y) = \delta(\{7,8,9,10,12,13\},y) = \sim(11) = G^{*}$

$\delta(G,x) = \delta(\{7,8,10,11,12,13\},x) = \sim(9) = F^{*}$

$\delta(G,y) = \delta(\{7,8,10,11,12,13\},y) = \sim(11) = G^{*}$

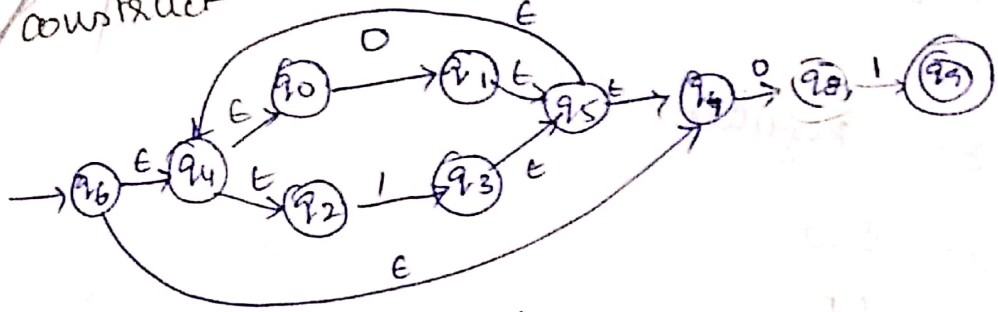
δ	x	y
A	B	C
B	D	E
C	D	E
D*	F	G
E	E	E
F*	F	G
G*	F	G

δ	x	y
→ A	[B, C]	[B, C]
[B, C]	[D, F, G]	E
* [D, F, G]	[D, F, G]	[D, F, G]
E	E	E



2, 8, 10, 15

Construct minimized DFA from RE $(0+1)^* 01$



$\epsilon(q_6) = \{q_6, 7, 4, 0, 2\} = A$

$\delta(A, 0) = \delta(\{0,2,4,6,7\}, 0) = \epsilon(1, 8) = \{1,5,4,0,2,7,8\} = B$

$\delta(A, 1) = \delta(\{0,2,4,6,7\}, 1) = \epsilon(3) = \{3,5,4,0,2,7\} = C$

$\delta(B, 0) = \delta(\{0,1,2,4,5,7,8\}, 0) = \epsilon(1, 8) = B$

$\delta(B, 1) = \delta(\{0,1,2,4,5,7,8\}, 1) = \epsilon(3, 9) = \{3,5,4,0,2,7,9\} = D^*$

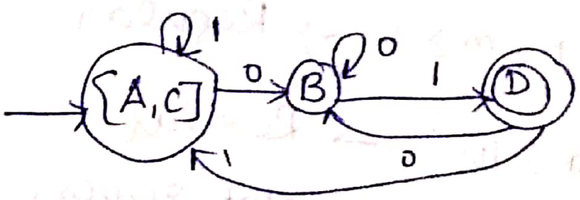
$\delta(C, 0) = \delta(\{0,2,3,4,5,7\}, 0) = \epsilon(1, 8) = B$

$\delta(C, 1) = \delta(\{0,2,3,4,5,7\}, 1) = \epsilon(3) = C$

$\delta(D, 0) = \delta(\{0,2,3,4,5,7,9\}, 0) = \epsilon(1, 8) = B$

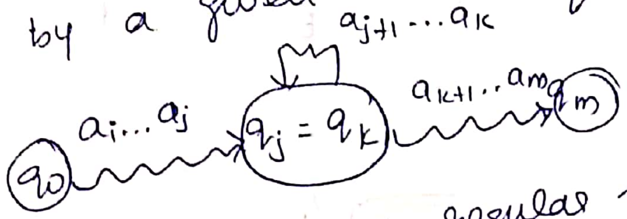
$\delta(D, 1) = \delta(\{0,2,3,4,5,7,9\}, 1) = \epsilon(3) = C$

δ	0	1
→ A	B	C
B	B	D
C	B	C
* D	B	C



Pumping Lemma

Pumping lemma is a powerful tool for proving certain language regular or not. Development of algorithms, languages accepted by a given FA is finite/infinite



Let L be a regular set. There is a constant n such that if z is any word in L and $|z| \geq n$, $z = uvw$ such that $|uv| \leq n$, $|v| \geq 1$ for all $i \geq 0$, uv^iw is in L

1. $0^n \mid n \geq 1$ Regular

$x = uvw$

$|uv| \leq n$

$|v| \geq 1$

for all $i \geq 0$ ~~the~~ $uv^iw \in L$

$\boxed{\text{when } L \text{ is Regular?}}$

(i) No dependency in power

(ii) If dependent, limit should be bound

(iii) Power should be ~~prime~~ AP

$uvw = a^n$ $z = uvw$

$\therefore a\, a^{n-2}\, a$

$|v| = n-2$

$|v| \geq 1$ true

$|uv| \leq n$ true

2. $a^n b^m \mid n, m \geq 1 \rightarrow$ Regular

3. $a^n b^n \mid n \leq 10^{10} \rightarrow$ Regular

4. $a^n b^n \mid n \geq 1 \rightarrow$ Not regular

5. $ww^R \mid |w| = 2 \rightarrow$ Regular

v w

aabb

aabbb

6. $ww^R \mid w \in (a,b)^*$ → Not Regular

7. $ww \mid w \in (a,b)^*$ → Not Regular

8. $a^n b^m c^k \mid n,m,k \geq 1$ → Regular

9. $a^i b^{2j} \mid i,j \geq 1$ → Regular

10. $a^i b^{4j} \mid i,j \geq 1$ → Regular

11. $a^n \mid n$ is even Arithmetic progression → Regular

12. $a^n \mid n$ is odd AP → Regular

13. $a^n \mid n$ is prime Not AP → not Regular

14. $a^{n^2} \mid n \geq 1$, not AP → not Regular

15. $a^{2^n} \mid n \geq 1$, not AP → not regular

16. $L = \{0^n 1 0^n \mid n \geq 1\}$ → Not Regular

UNIT II

GRAMMAR

→ Study of ~~human languages~~ languages.
→ Defines programming languages
→ string processing applications
→ Arithmetic expression → nesting of balanced { }
→ Block structure

A context free grammar is a finite set of $\boxed{\text{Variables}}$ (non terminals) each of which represents a language. Languages represented by variables are described recursively by primitive symbols called $\boxed{\text{Terminals}}$. Rules relating variables are called $\boxed{\text{productions}}$

(V, T, P, S)