8th August, 2023

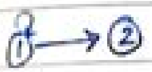
Finite State Machines

Set : collection of unique data (unordered)

9th August, 2023

Relation

R S T



R

$\phi = \{\}$

$\lambda / \varepsilon = \{\varepsilon\}$

↑ string length 0

$\Sigma = \{a, b\}$ => languages of a and b

alphabet

w / strings = a, b, aa, ab, ba, bb, aaa, aab, bba

ε / λ = length is 0

$|ab| = 2$ (length of string)

Power $\Sigma^{i = 0 \ldots \infty}$

$\Sigma^0 = \varepsilon / \lambda$

$\Sigma^1 = a, b$

$\Sigma^2 = aa, ab, ba, bb$

Σ^* / Kleen's star

$*$ => 0 or more

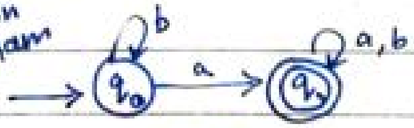
$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots\ldots$

Σ^+ / Kleen's Plus

$+ \Rightarrow$ 1 or more

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3$$

Transition diagram



Language / L = { strings of a's and b's with atleast one a }

$$\underset{\text{FSM}}{M} = (Q, \Sigma, \delta, q_0, F/A)$$

same for DFA, NFA, λ-NFA

- Q = finite set of states
- Σ = finite set of alphabets
- q_0 = start state (only one). $q_0 \in Q$
- F/A = finite set of final / accepting states (more than 1) $F \subseteq Q$

δ = transition

$$\underset{\text{state}}{Q} \times \underset{\text{alphabet}}{\Sigma} \text{ to } \underset{\text{state}}{Q}$$

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_0$

$\delta(q_1, a) = q_1$

$\delta(q_1, b) = q_1$

Transition Table

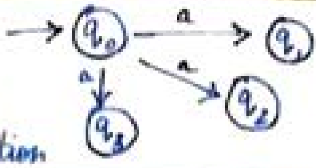
δ	a	b
$\rightarrow q_0$	q_1	q_0
$* q_1$	q_1	q_1

* Deterministic Finite Automata / state / Machine (DFA, DFSM)

$\Downarrow$ exactly one transition for every alphabet from every state

$Q = \{q_0, q_1\}$ $\quad Q \times \Sigma = \{(q_0, a), (q_0, b), (q_1, a), (q_1, b)\}$

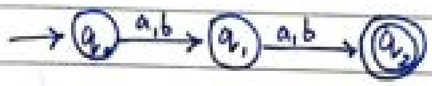
$\Sigma = \{a, b\}$



→ Pattern Recognition

Q1. Construct a DFA to accept alphabet a and b of string length greater than or equal to 2.

1) $\Sigma = \{a, b\}$
2) $w = \{aa, ab, ba, bb, aba, bab, \ldots\}$
3) minimum length string 2 ⇒ ab, ba, aa, bb
4) Draw skeleton DFA



5) Identify remaining transitions

2M

δ	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_2	q_2
q_2	q_2	q_2

$M = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
q_0 is start state
$F = \{q_2\}$

Q2. Construct DFA for $\Sigma = \{a, b\}$ to accept atleast 2 a's.

$\Sigma = \{a, b\}$
$w = \{aa, aba, aab, baa, bbbb\}$
min length string 2 ⇒ aa

skeleton DFA

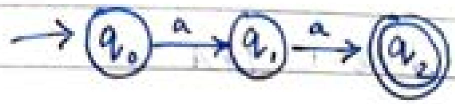
Remaining

q.3. Construct DFA for $\Sigma = \{a, b\}$ to accept exactly 2 a's.

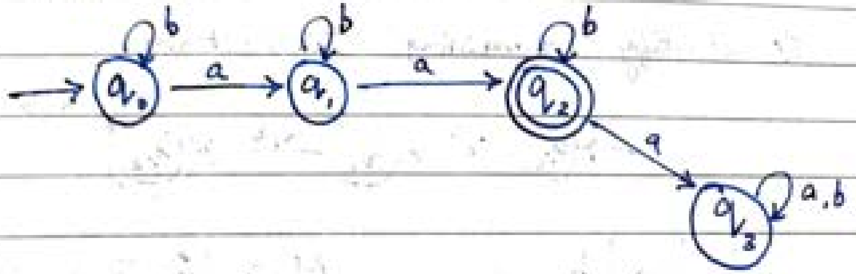
$\Sigma = \{a, b\}$

$w = \{aa, aba, aab, baa, bbaa, aabbb \ldots\}$

Skeleton DFA



final DFA



q.4.

less than 2 a's

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

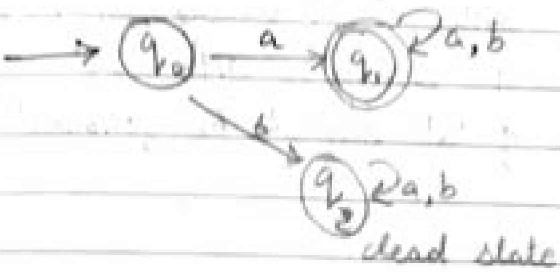
$w = \{ab, bbb, b, a, bab, bbbabbbb$

min length string = 1

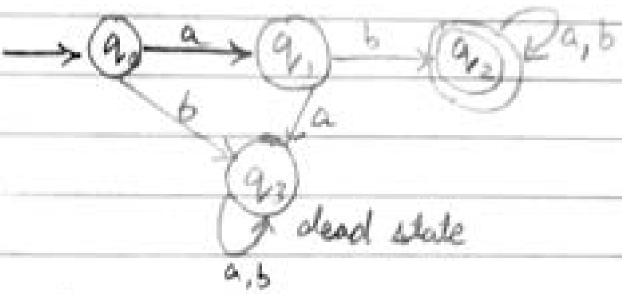
Construct a DFA for $\Sigma = \{a, b\}$

5. strings starting with 'a'
6. strings starting with 'ab'
7. strings ending with 'a'
8. strings ending with 'ab'
9. substring 'ab'

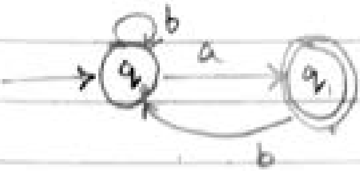
5. $w = \{a, ab, aba, abaaa, abbb \ldots$



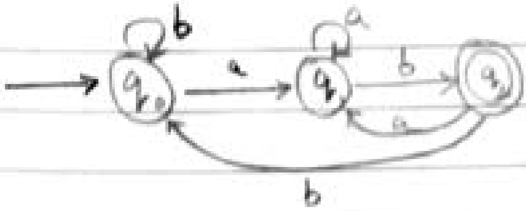
6. $w = \{ab, aba, abb, abaaa, abbbb, ababb \ldots \}$



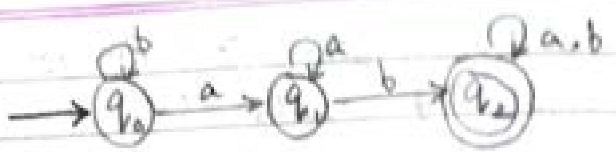
7. $w = \{a, ba, aa, bbba, aaaa \ldots \}$



8.



9.

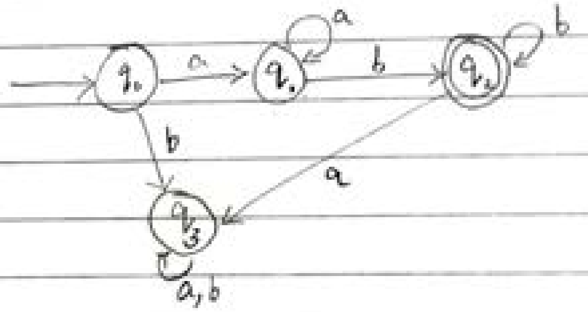


11th August, 2023

q. Construct DFA over $\Sigma = \{a, b\}$

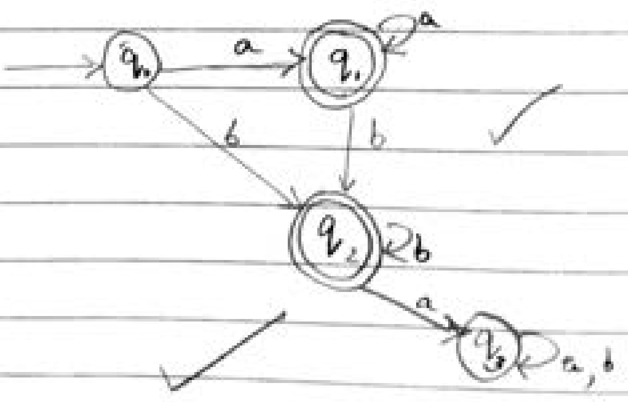
10. $L = \{ a^n b^m \mid n, m \geq 1 \}$
11. $L = \{ a^n b^m \mid m, n \geq 0 \}$
12. strings starting with 'a' & ending with 'b'
13. strings starting & ending with 'ab'
14. every occurrence of 'a' should be followed by 'bb'.

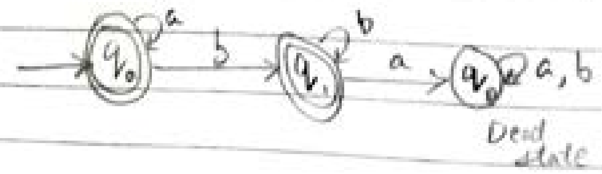
10. $w = \{ ab, a^2 b, ab^2, a^3 b^3 \ldots \}$



11. $w = \{ \lambda, a, b, aaa, bbb, ab, a^2 b, ab^2, \ldots$

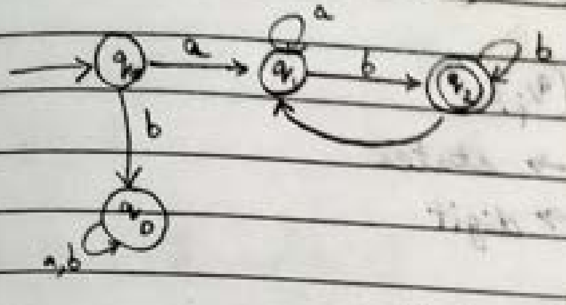
$n = 0$
$m = 0$



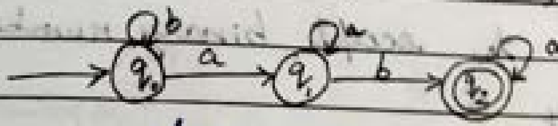


(teacher's soln)

12. $w = \{ab, abab, abaaab, abbbbab, abababab\}$

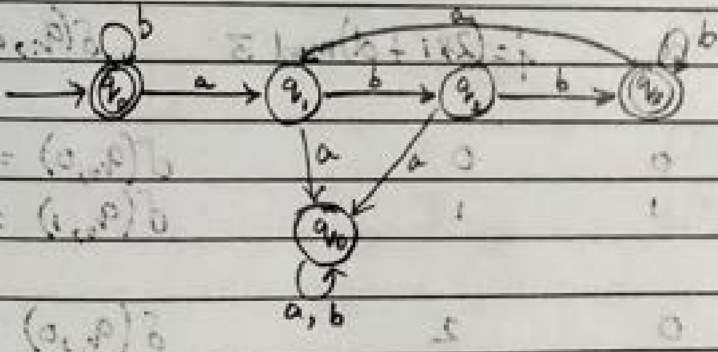


13. $w = \{ab, aba^n, a^n ab, b^n ab, abb^n, ab, ab(a^n b^n)^n ab$



here even string babb will also be accepted, this DFA is for substring ab (part of NFA).

14.



→ Types of DFA

1. Pattern Recognition
2. Divisible by K
3. Modulo K counter

II → Divisible by K

$$\delta(q_i, d) = (q_j)$$

$i, j \rightarrow$ states

$d \rightarrow$ digit

$$j = (r * i + d) \bmod K$$

(r ↑ radix)

q: Construct DFA to accept binary numbers divisible by 3

$r = 2 \qquad k = 3$

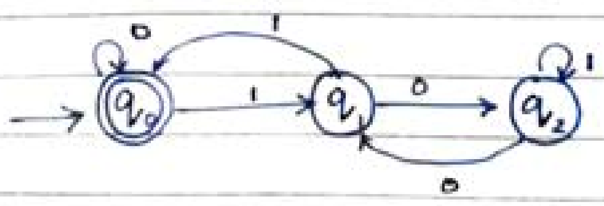
$d \Rightarrow \Sigma = \{0, 1\}$

$i = \{0, 1, 2\}$

required

i	d	$j = (2*i + d) \bmod 3$	$\delta(q_i, d) = q_j$
0	0	0	$\delta(q_0, 0) = q_0$
	1	1	$\delta(q_0, 1) = q_1$
1	0	2	$\delta(q_1, 0) = q_2$
	1	0	$\delta(q_1, 1) = q_0$
2	0	1	$\delta(q_2, 0) = q_1$
	1	2	$\delta(q_2, 1) = q_2$

required



0

0011

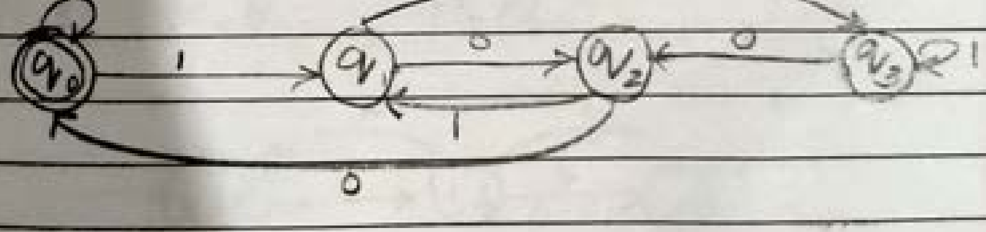
0110

q.2 Construct DFA to
by 4

$r = 2$ $\quad$ $K = 4$

$d \Rightarrow \Sigma = \{0, 1\}$
$i = \{0, 1, 2, 3\}$

i	d	$j = (2 * i + d) \bmod 4$	$\delta(q_i, d) = q_j$
0	0	0	$\delta(q_0, 0) = q_0$
	1	1	$\delta(q_0, 1) = q_1$
1	0	2	$\delta(q_1, 0) = q_2$
	1	3	$\delta(q_1, 1) = q_3$
2	0	0	$\delta(q_2, 0) = q_0$
	1	1	$\delta(q_2, 1) = q_1$
3	0	2	$\delta(q_3, 0) = q_2$
	1	3	$\delta(q_3, 1) = q_3$



q. Construct a DFA to accept base 3 divisible by 3.

$\lambda = 3 \qquad K = 3$

$d \Rightarrow \Sigma = \{0, 1, 2\}$

* for a's keep the counter horizontally
* for b's keep the counter vertically

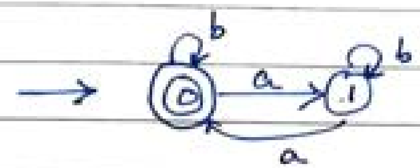
III → Modulo K counter

Q. Construct DFA to accept even no. of a's and even no. of b's $\Sigma = \{a, b\}$

$L = \{ n_a(w) \bmod 2 = 0 \ \& \ n_b(w) \bmod 2 = 0 \}$

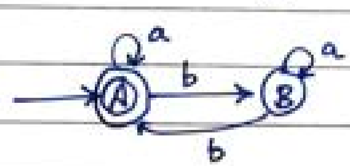
⇒ final state = 00

1) M_1



& ⇒ cross product

2) M_2



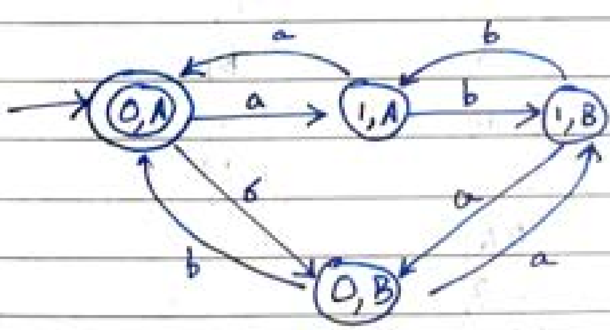
$M = M_1 \times M_2$

$Q_1 \times Q_A = \{ (0\ A), (0\ B), (1\ A), (1\ B) \}$

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

$q_0 \Rightarrow 0 \times A = (0, A)$

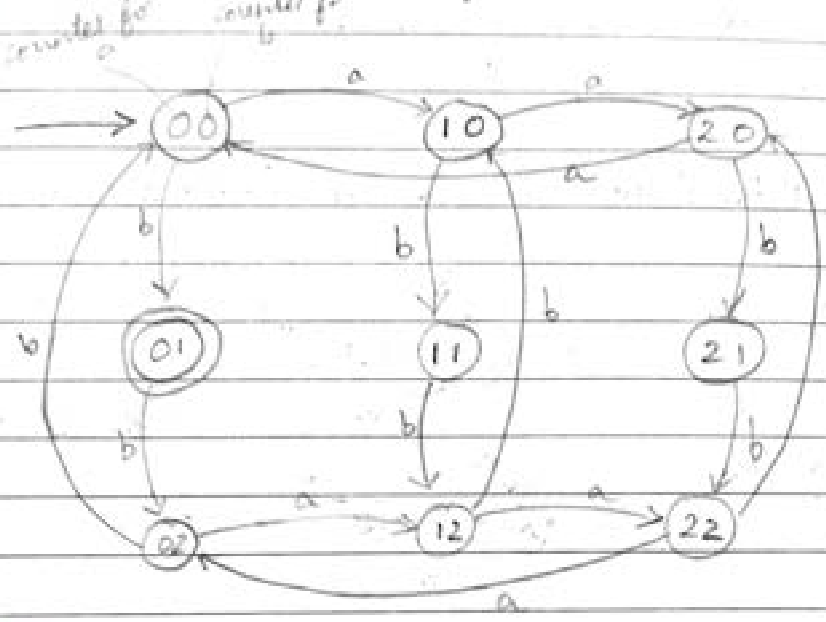
	a	b
→(0, A)	(1, A)	(0, B)
(0, B)	(1, B)	(0, A)
(1, A)	(0, A)	(1, B)
(1, B)	(0, B)	(1, A)



q.2 $n_a(w) \bmod 3 = 0$ & $n_b(w) \bmod 3 = 1$

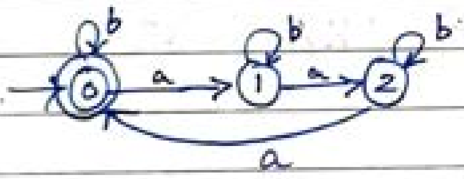
0, 1, 2

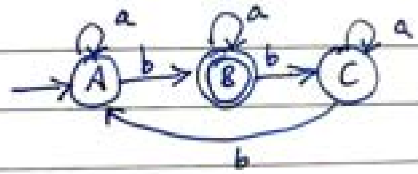
$\Rightarrow$ final state = 01



16th August, 2023

$n_a(w) \bmod 3 = 0$ & $n_b(w) \bmod 3 = 1$

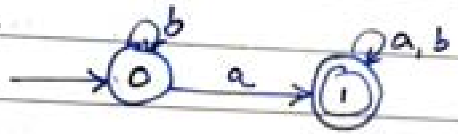


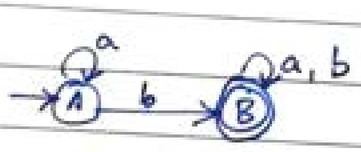


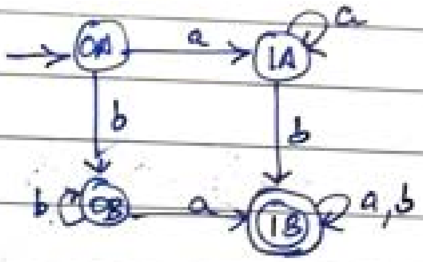
	a	b
→ (0 A)	(1 A)	(0 B)
(0 B)	(1 B)	(0 C)
(0 C)	(1 C)	(0 A)
(1 A)	(2 A)	(1 B)
(1 B)	(2 B)	(1 C)
(1 C)	(2 C)	(1 A)
(2 A)	(0 A)	(2 B)
(2 B)	(0 B)	(2 C)
(2 C)	(0 C)	(2 A)

q.3. Construct DFA over {a, b} where strings should accept atleast one 'a' & atleast one 'b'.

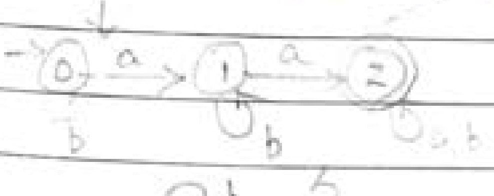
{ab, ba, aba, bab, aaab, bbba . . . }

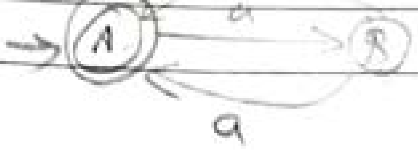




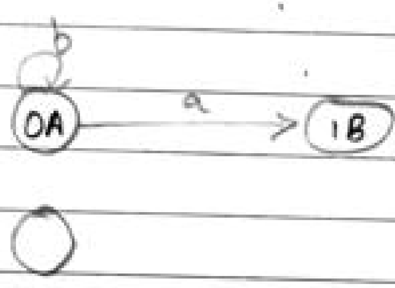


q.4. atleast 2 a's & ends with even no. of a's



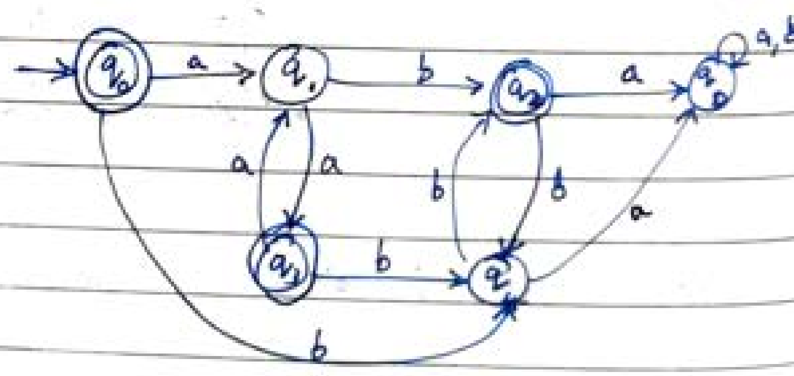


	a	b
→(0 A)	(1 B)	(0 A)
(0 B)	(1 A)	(0 A)
(1 A)	(2 B)	(1 A)
(1 B)	(2 A)	(1 A)
(2 A)	(2 B)	(2 A)
(2 B)	(2 A)	(2 A)



q5. $L = \{a^n b^m \mid (n+m) \bmod 2 = 0 \ \& \ n, m \geq 0\}$

n	m	
0	0	λ
0	>0	bb
>0	0	aa



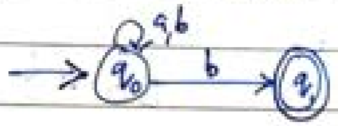
q6: $n_a(w) \bmod 3 = 2$ & $n_b(w) \bmod 3 = 1$

Non-Deterministic Finite Automata.

$$M = (Q, \Sigma, \delta, q_0, A)$$
$$\delta : Q \times \Sigma \text{ to } 2^Q$$

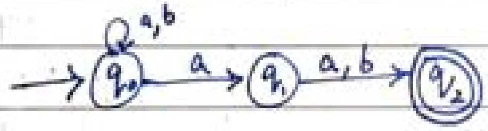
for every state, for every alphabet there can be 0/more transitions.

eg: L = of a's b's where strings are ending with b

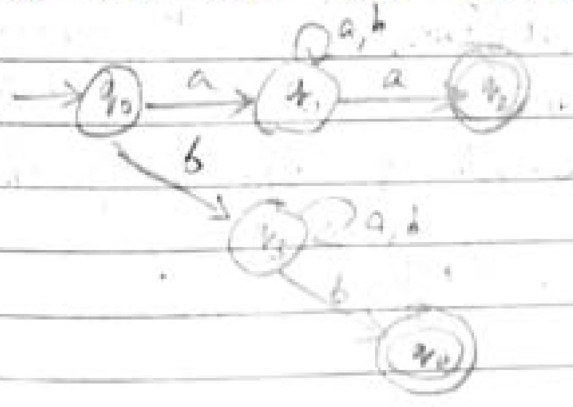


q: L of strings a's and b's, where the second symbol from RHS is 'a'.

a/b a a/b



q: start and end with same symbol



a .... a
b .... b

→ NFA to DFA conversion

A) Subset - Conversion Method

1) Identify states of DFA

eg:- $Q_{DFA} = \{\phi, \{q_0\}, \{q_1\}, \{q_0 q_1\}\}$ Power set

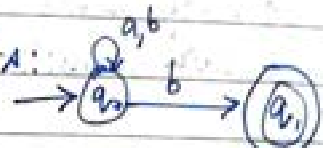
$\Sigma = \{a, b\}$

$q_0 \Rightarrow$ start of NFA is start of DFA $\Rightarrow q_0$

$F \Rightarrow \{q_0, q_0 q_1\}$

$\delta \Rightarrow$

NFA:

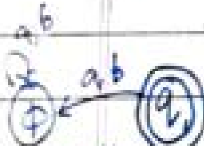


for DFA:

	a	b
ϕ	ϕ	ϕ
$\rightarrow q_0$	q_0	$\{q_0 q_1\}$
q_1	ϕ	ϕ
$q_0 q_1$	q_0	$\{q_0 q_1\}$

for NFA:

	a	b
$\rightarrow q_0$	q_0	$q_0 q_1$
$* q_1$	ϕ	ϕ



unreachable state

DFA :

reachable state

also specify all tuples

$M = (Q = \{q_0 q_1\}$

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

q_0 is start state

$F = \{\{q_0 q_1\}\}$

$\delta =$ T.T & T.D)

18th August, 2023

A) Subset Conversion Method

1. $Q_{DFA} = \text{Powerset}(Q_{NFA})$
2. $q_{0\,DFA} = q_{0\,NFA}$
3. $F_{DFA} =$ Every subset of Q_{DFA} having F_{NFA}
4. $\Sigma_{DFA} = \Sigma_{NFA}$
5. $\delta((q_0 q_1), a) = (\delta(q_0, a) \cup \delta(q_1, a))$

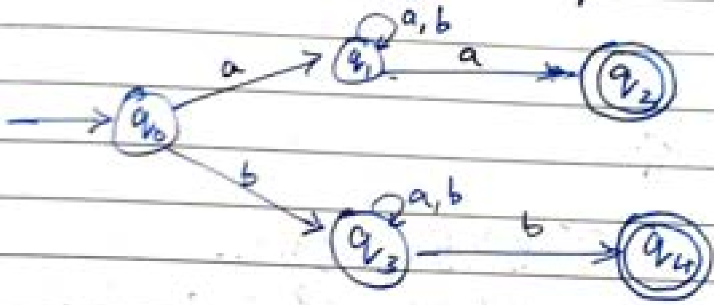
5) Lazy Evaluation Method – start with start state then write new states

1. $\Sigma_{DFA} = \Sigma_{NFA}$
2. start state $q_{0\,DFA} = q_{0\,NFA}$
3. Identify transition
   a. start state
   b. every new state identified.

Q. 2 starts & ends with same alphabet

lazy evaluation method

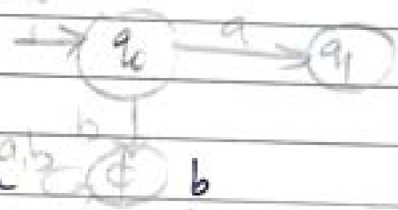
NFA:



T.T

		a	b
→	q_0	q_1	q_3
	q_1	$q_1 q_2$	q_1
*	q_2	ϕ	ϕ
	q_3	q_3	$q_3 q_4$
*	q_4	ϕ	ϕ

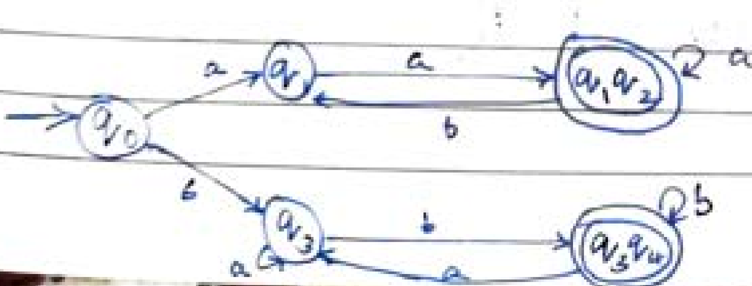
	a	b
q_0	q_1	ϕ
q_1	$q_1 q_2$	q_1
$q_1 q_2$	$q_1 q_2$	q_1
ϕ	ϕ	ϕ



DFA:

T.T

	δ	a	b
→	$\{q_0\}$	$\{q_1\}$	$\{q_3\}$
	$\{q_1\}$	$\{q_1, q_2\}$	$\{q_1\}$
	$\{q_3\}$	$\{q_3\}$	$\{q_3 q_4\}$
*	$\{q_1 q_2\}$	$\{q_1 q_2\}$	$\{q_1\}$
*	$\{q_3 q_4\}$	$\{q_3\}$	$\{q_3 q_4\}$



Q3 second symbol from RHS is a.

NFA:

T.D:

T.T:

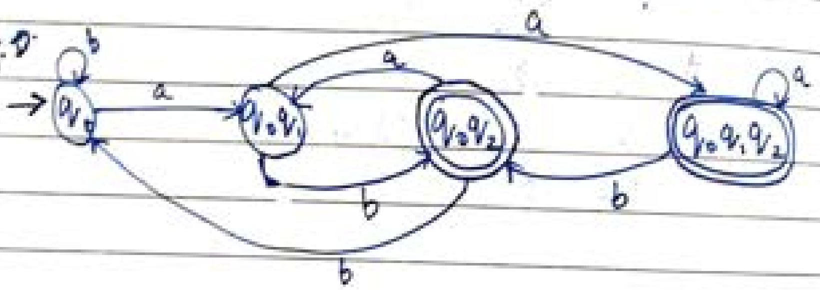
	a	b
q_0	$q_0 q_1$	q_0
q_1	q_2	q_2
q_2	ϕ	ϕ

DFA:

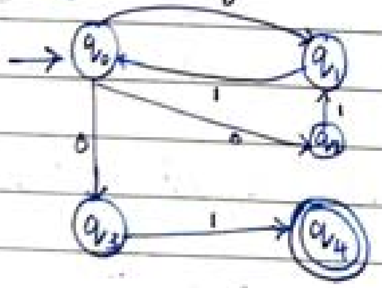
T.T:

	a	b
$\rightarrow \{q_0\}$	$\{q_0 q_1\}$	$\{q_0\}$
$\{q_0 q_1\}$	$\{q_0 q_1 q_2\}$	$\{q_0 q_2\}$
$* \{q_0 q_2\}$	$\{q_0 q_1\}$	$\{q_0\}$
$* \{q_0 q_1 q_2\}$	$\{q_0 q_1 q_2\}$	$\{q_0 q_2\}$

T.D:



Q4 Convert given NFA to DFA

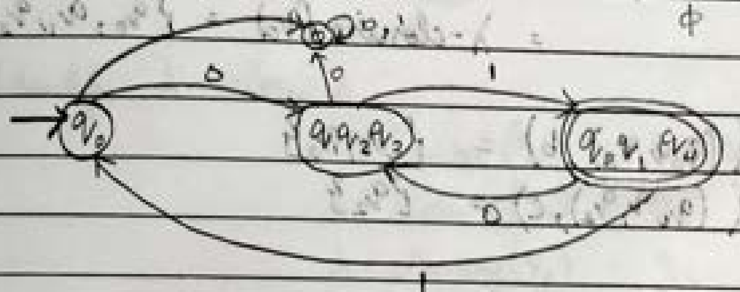


T.T

δ	0	1
$\rightarrow q_0$	$q_1 q_2 q_3$	ϕ
q_1	ϕ	q_0
q_2	ϕ	q_1
q_3	ϕ	q_4
$*q_4$	ϕ	ϕ

T.T

δ	0	1
$\rightarrow q_0$	$q_1 q_2 q_3$	ϕ
$q_1 q_2 q_3$	ϕ	$q_0 q_1 q_4$
$*q_0 q_1 q_4$	$q_1 q_2 q_3$	q_0

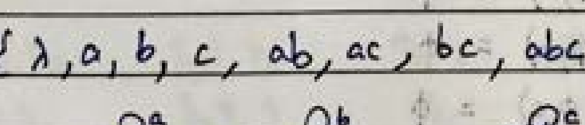


λ - NFA / ε - NFA

$\delta : Q \times (\Sigma \cup \lambda)$ to 2^Q

q.1. Construct λ-NFA over alphabets a, b, c where strings of 0 or more 'a''s followed by strings of 0 or more 'b''s followed by strings of 0 or more 'c''s are accepted.

{λ, a, b, c, ab, ac, bc, abc . . .



E-closure / ε-closure / λ-closure
without giving any I/P which states are reachable.

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

1) start state DFA = λ-closure start state of NFA
= λ-closure (q_0) = $\{q_0, q_1, q_2\}$ = A

2) Identify Transitions

start state on a: $\delta((q_0 q_1 q_2), a) = \lambda\text{-closure}(\underbrace{\delta(q_0, a)}_{q_0} \cup \underbrace{\delta(q_1, a)}_{\phi} \cup \underbrace{\delta(q_2, a)}_{\phi})$
$= \lambda\text{-closure}(q_0) = \{q_0, q_1, q_2\}$

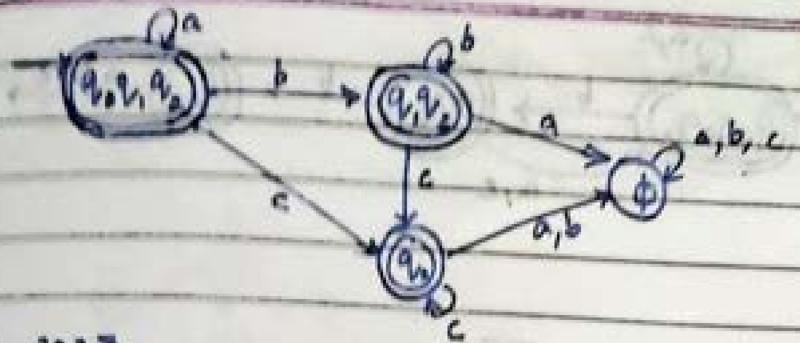
start state on b: $\delta((q_0 q_1 q_2), b) = \{q_1, q_2\}$
on c: $\delta((q_0 q_1 q_2), c) = \{q_2\}$

for $\{q_1 q_2\}$ on I/P a, b, c:
$\delta(\{q_1, q_2\}, a) = \phi$
$\delta(\{q_1, q_2\}, b) = \{q_1, q_2\}$
$\delta(\{q_1, q_2\}, c) = \{q_2\}$

for $\{q_2\}$ on I/P a, b, c:
$\delta(q_2, a) = \phi$
$\delta(q_2, b) = \phi$
$\delta(q_2, c) = \{q_2\}$

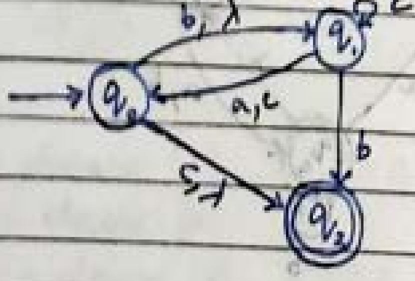
for ϕ on I/P a, b, c:
$\delta(\phi, a) = \phi$
$\delta(\phi, b) = \phi$
$\delta(\phi, c) = \phi$

DFA
TD



22nd August, 2023

Q3 Convert the given λ-NFA to DFA



i) Identify λ-closure

$\lambda\text{-closure}(q_0) = q_0 q_1 q_2$

$\lambda\text{-closure}(q_1) = q_1$

$\lambda\text{-closure}(q_2) = q_2$

TT NFA

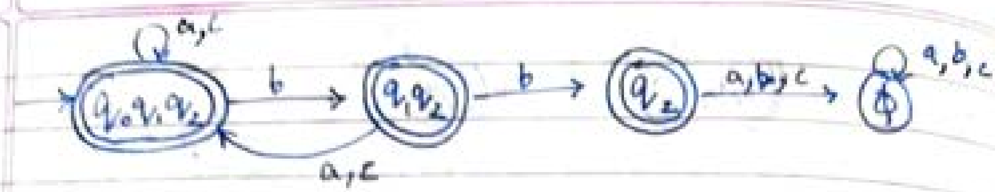
	λ-closure	a	b	c
$\to q_0$	$q_0 q_1 q_2$	ϕ	q_1	q_2
q_1	q_1	q_0	q_2	$q_0 q_1$
$*\, q_2$	q_2	ϕ	ϕ	ϕ

TT DFA

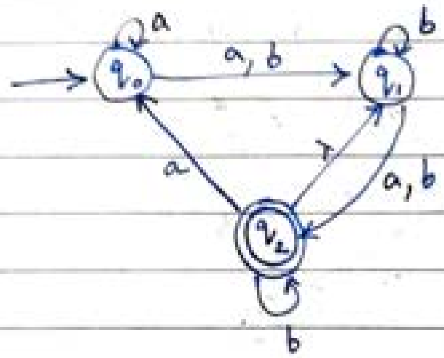
	a	b	c
$\to * q_0 q_1 q_2$	$q_0 q_1 q_2$	$q_1 q_2$	$q_0 q_1 q_2$
$*\, q_1 q_2$	$q_0 q_1 q_2$	q_2	$q_0 q_1 q_2$
$*\, q_2$	ϕ	ϕ	ϕ
ϕ	ϕ	ϕ	ϕ

$q_1 q_1 q_2$ on 'a' is q_0, λ-closure of q_0 is $q_0 q_1 q_2$

T.D DFA



q3. λ-NFA to DFA



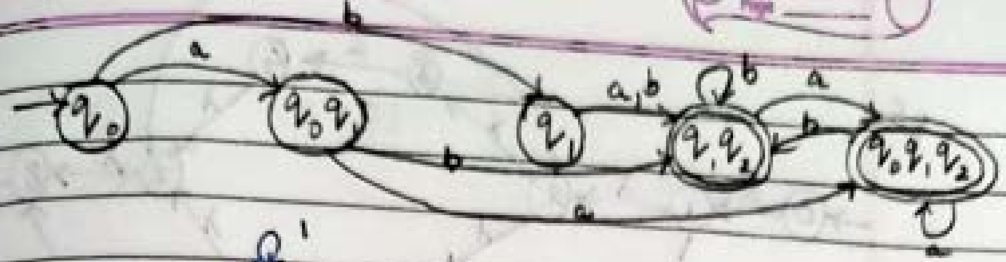
λ-closure $(q_0) = q_0$
λ-closure $(q_2) = q_1 q_2$
λ-closure $(q_1) = q_1$

TT. NFA

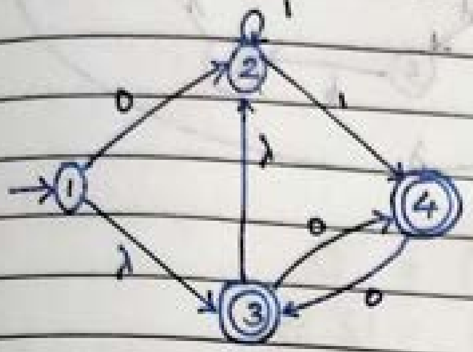
	λ-closure	a	b
$\rightarrow q_0$	q_0	$q_0 q_1$	q_1
q_1	q_1	q_2	$q_1 q_2$
$* q_2$	$q_1 q_2$	q_0	q_2

TT DFA

	a	b
$\rightarrow q_0$	$q_0 q_1$	q_1
$q_0 q_1$	$q_0 q_1 q_2$	$q_1 q_2$
q_1	$q_1 q_2$	$q_1 q_2$
$* q_1 q_2$	$q_0 q_1 q_2$	$q_1 q_2$
$* q_0 q_1 q_2$	$q_0 q_1 q_2$	$q_1 q_2$



Q4.



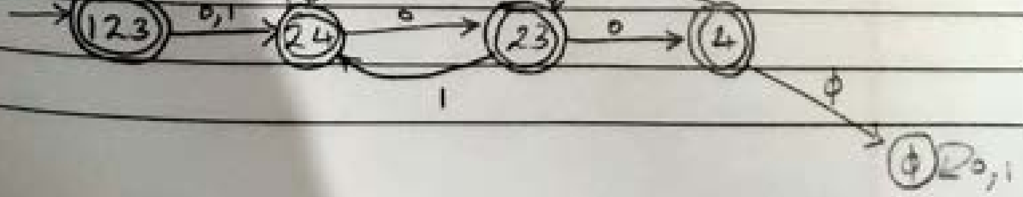
λ- closure (1) = 132
λ- closure (2) = 2
λ- closure (3) = 23
λ- closure (4) = 4

TT NFA

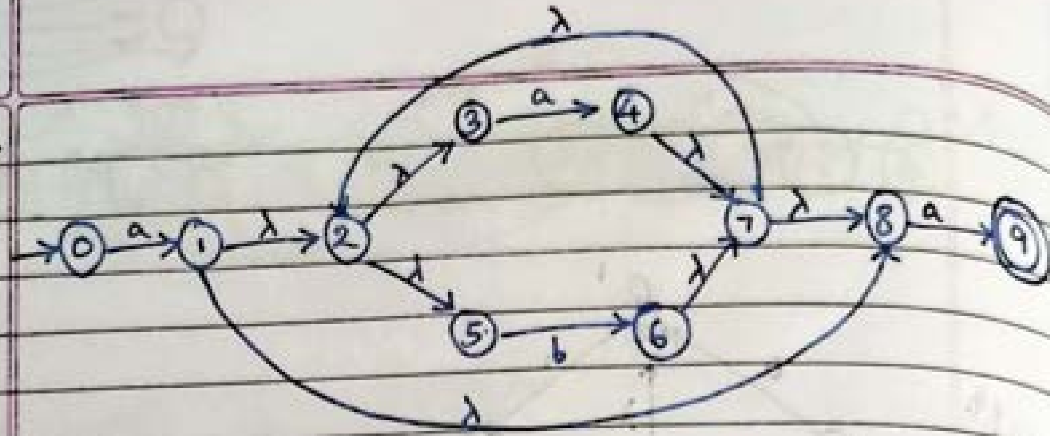
	λ-closure	0	1
→1	132	2	ϕ
2	2	ϕ	24
* 3	23	4	ϕ
* 4	4	3	ϕ

TT DFA

	0	1
→132	24	24
24	23	24
* 23	4	24
* 4	23	ϕ



Q5.



λ-closure(0) = 0

(1) = 12358

(2) = 235

(3) = 3

(4) = 478235

(5) = 5

(6) = 678235

(7) = 78235

(8) = 8

(9) = 9

TT NFA

	λ-closure	a	b
→ 0	0	1	φ
1	12358	φ	φ
2	235	φ	φ
3	3	4	φ
4	478235	φ	φ
5	5	φ	6
6	235678	φ	φ
7	23578	φ	φ
8	8	a	φ
9	9	φ	φ

row: 1 – n
col : 0 – (n – 1)

23rd August, 2023

Minimization of DFA

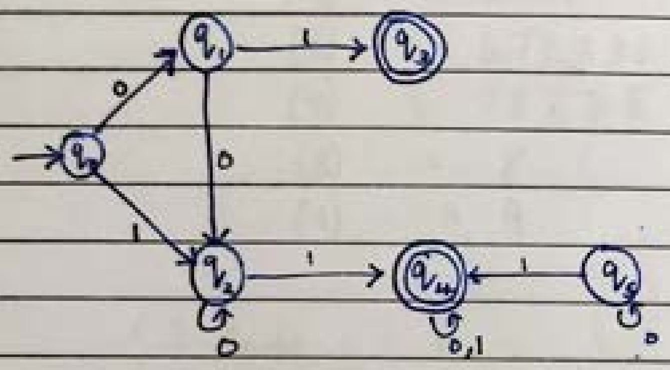
1) DFA: may not be unique.

2) Minimized DFA is unique

3) Identify Distinguishable & non-distinguishable state
→ Table Filling algorithm.

4) Distinguishable & non-distinguishable states will be final states of DFA.

q1.



Sol 1) Remove dead state and unreachable state

δ	0	1
$\rightarrow q_0$	q_1	q_2
q_1	q_2	q_3
q_2	q_2	q_4
$* q_3$	ϕ	ϕ
$* q_4$	q_4	q_4

2) Table :

a)

q_1	X			
q_2	X			
$* q_3$	X	X	X	
$* q_4$	X	X	X	
	q_0	q_1	q_2	q_3^*

cross cells with final & non-final cells.

b)

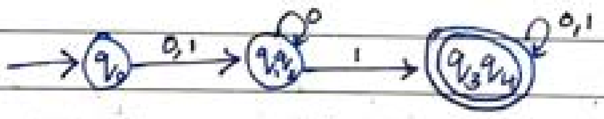
	0	1
$q_0 q_1$	$q_1 q_2$	$q_2 q_3$
$q_0 q_2$	$q_1 q_2$	$q_2 q_4$
$q_1 q_2$	q_2	$q_3 q_4$
$q_3 q_4$	q_4	q_4

identify transitions for uncrossed cells.
If transitions are going to crossed cells then put a cross for that pair (black pen crosses)

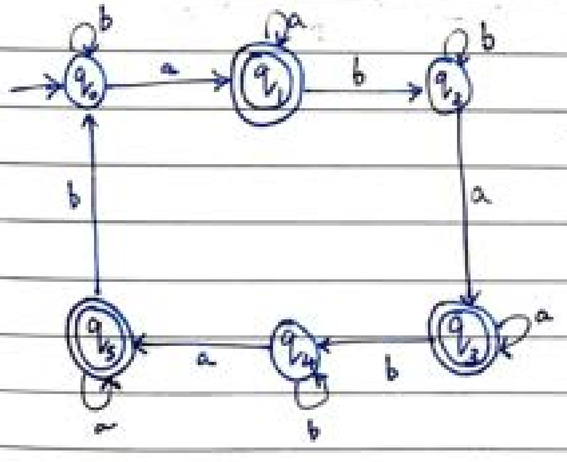
c) distinguishable states : q_0
non-distinguishable states : $(q_1 q_2)$ $(q_3 q_4)$
(uncrossed cells)

3) TT for minimized DFA

	0	1
$\rightarrow q_0$	$q_1 q_2$	$q_1 q_2$
$q_1 q_2$	$q_1 q_2$	$q_3 q_4$
$* q_3 q_4$	$q_3 q_4$	$q_3 q_4$



q2.



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

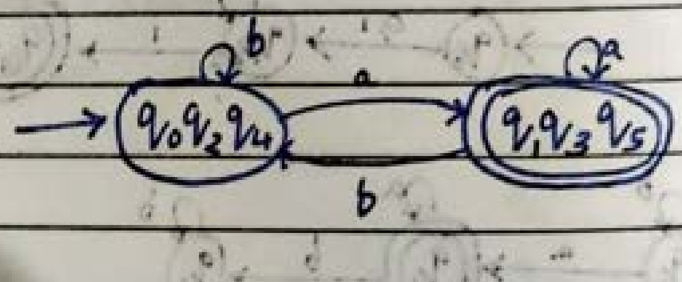
q_1	X				
q_2		X			
* q_3	X		X		
q_4		X		X	
* q_5	X		X		X
	q_0	q_1 *	q_2	q_3 *	q_4

	a	b
q_0q_2	q_1q_3	q_0q_2
q_0q_4	q_1q_5	q_0q_4
q_1q_5	q_1q_5	q_0q_2
q_1q_3	q_1q_3	q_2q_4
q_2q_4	q_3q_5	q_2q_4
q_3q_5	q_3q_5	q_0q_4

(q_0q_2) (q_0q_4) (q_2q_4)

(q_1q_3) (q_1q_5) (q_3q_5)

⇒



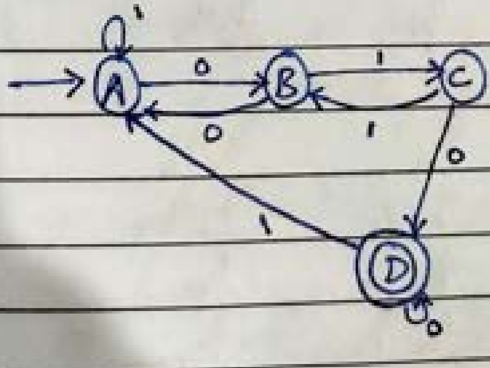
q3

	0	1
→A	B	A
B	A	C
C	D	B
*D	D	A
E	D	F
F	G	E
G	F	G
H	G	D

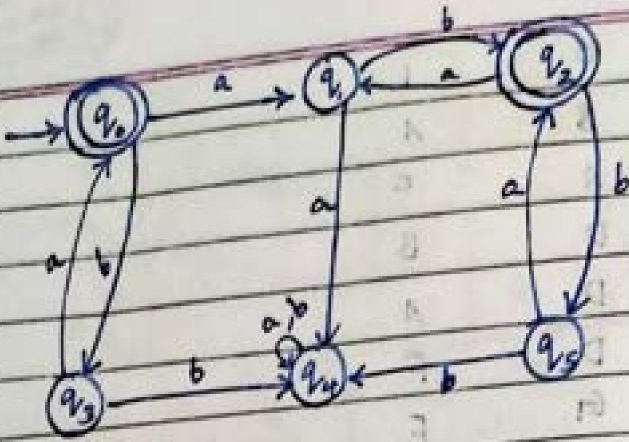
non-reachable

B	X						
C	X	X					
*D	X	X	X				
E	X	X	X	X			
F	X	X	X	X	X		
G	X	X	X	X	X	X	
H	X	X	X	X	X	X	X
	A	B	C	D*	E	F	G

	0	1
×AC	BD×	AB×
×AB	AB×	AC×
×BC	AD×	BC×



q4.



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

q_1	X				
$*q_2$		X			
q_3	X	X	X		
dead q_4	X	X	X	X	
q_5	X	X	X		X
	q_0 *	q_1	q_2 *	q_3	q_4 dead

Cannot be clubed with anything

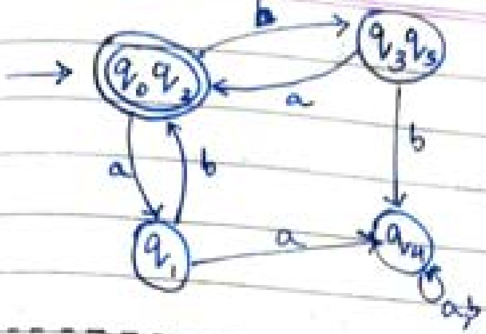
⇓

anything that contains q_4 is X

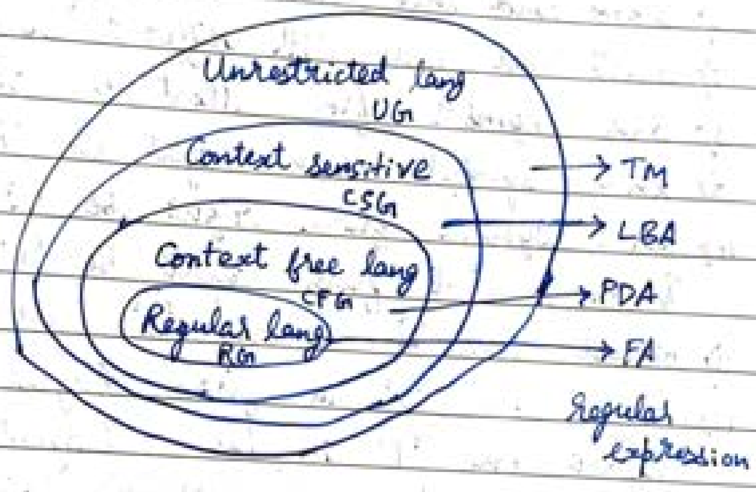
⇓

q_4 alone is valid

	a	b
$q_0 q_2$	q_1	$q_3 q_5$
X q_1, q_4	q_4	$q_2 q_4$
X q_1, q_5	q_2, q_4	$q_2 q_4$
X $q_3 q_4$	$q_0 q_4$	q_4
$q_3 q_5$	$q_0 q_2$	q_4



END OF UNIT-1 (acc to old 5 units)



ϕ

ε / λ

a

a^+

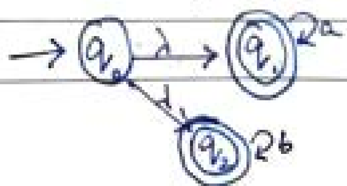
a^*

$a+b$ ⇒ either a or b

$a.b$ ⇒ a followed by b

$(a+b)^*$ ⇒ 

$(a.b)$ ⇒ 

$a^* + b^*$ ⇒ 

25th August, 2025

Regular Expressions (RegEx)

Write RegEx over a and b

1) $|w| = 0$: λ / $(a+b)^0$
2) $|w| = 2$: $(a+b)(a+b)$ / $(a+b)^2$ / $(aa+ab+ba+bb)$
3) ends with ab : $(a+b)^* ab$
4) starts with ab : $ab(a+b)^*$
5) contains ab :
6) starts and ends with same symbol : $[(a(a+b)^*a)+(b(a+b)^*b)]$
7) starts and ends with different symbol : $[(a(a+b)^*b)+(b(a+b)^*a)]$
8) 3rd symbol from start be 'a' : $(a+b)^2 a (a+b)^*$
9) 2nd symbol from right be 'b' : $(a+b)^* b (a+b)$
10) 3rd symbol from right be 'a' & 4th symbol from right be 'b' : $[(a+b)^* ba (a+b)^2]$
11) atleast one 'a' & one 'b' : $[((a+b)^* a (a+b)^* b (a+b)^*) + ((a+b)^* b (a+b)^* a (a+b)^*)]$
12) atleast one 'a' in first 3 symbols & atleast one 'b' in last 3 symbols
: $[(a(a+b)(a+b))+((a+b)a(a+b))+((a+b)(a+b)a)]\,(a+b)^*\,[(b(a+b)(a+b))+((a+b)b(a+b))+((a+b)(a+b)b)]$
13) $|w| \leq 2$: $(\lambda + a + b + a + ab + ba + bb)$ / $(a+b+\lambda)(a+b+\lambda)$
$(a+b)^0 + (a+b)^1 + (a+b)^2$ / $(a+b+\lambda)^2$
14) $|w| \geq 2$: $(a+b)^2 (a+b)^*$
15) $n_a(w) = 2$: $b^* a b^* a b^*$
16) $n_a(w) \geq 2$: $(a+b)^* a (a+b)^* a (a+b)^*$
17) $n_a(w) \leq 2$: $b^*(a+\lambda) b^* (a+\lambda) b^*$
18) $n_a(w) \bmod 2 = 0$: $(b^* a b^* a b^*)^* + b^*$
{a 0 b}
19) $n_a(w) \bmod 2 = 1$: $(b^* a b^*)(b^* a b^* a b^*)^*$
20) atleast one 'a' or one 'b' : $(a+b)^* (a+b) (a+b)^*$

→ valid adhaar no. must satisfy foll. conditions:

i) it should have 12 digits
ii) it should not start with 0/1.
iii) it should not contain any special characters
iv) it should have white space after 4 digits.

Sol

[2-9] ⇒ only 1 digit will be selected
{2} ⇒ power / repetition
\d = [0-9]
\s = space

. ⇒ single char without newline
? ⇒ 0/1
eg: ab?c → ac if ? = 0
abc if ? = 1
^abc ⇒ string starting with a
[^abc] ⇒ string won't contain a/b/c.
$ ⇒ matches char with end of line.

a{0, } = a^*
a{1, } = a^+
a{2, 3} = min len = 2, max len = 3

adhaar : ^[2-9]{1}\d{3}\s\d{4}\s\d{4}$

29th August, 2023

q2. PAN card : i) 10 characters long

ii) first 5 chrs should be any upper case alphabet

iii) next 4 chars should be any no. from 0 to 9

iv) last (tenth) char. should be

v)

$$\wedge[A-Z]\{5\}\backslash d\{4\}[A-Z]\{1\}\$$$

q3. Indian mobile number

i) 10 digit number

ii) first digit should contain no. b/w 6 to 9

iii) rest 9 digit can be any no. b/w 0 to 9

iv) mobile no. can have 11 digits also by including 0 in the starting

v) mobile no. can have 13 digits also by including +91 at the starting

^(0|"+91")?[6-9]{1}\d{9}$

OR

9876543210

q4. dates in the year 2020

* it is a leap year
* let the format be DD-MM-YY

- months with 31 days: Jan, Mar, May, July, Aug, Oct, Dec [1, 3, 5, 7, 8, 10, 12]
- months with 30 days: Apr, June, Sep, Oct [4, 6, 9, 11]
- month with 29 days: Feb [2]

* Year 2020

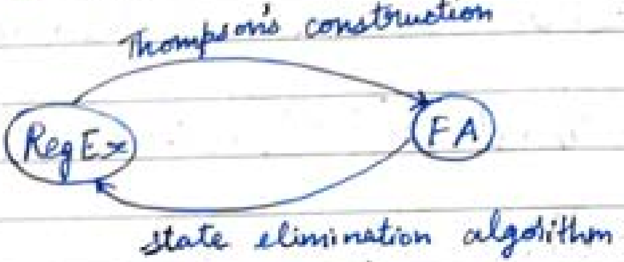
(
(0[1-9])|[10-31] "-" (0[13578]|1[02])
|
(0[1-9])|[10-30] "-" (0[469]|11)
|
(0[-9])|[10-29] "-" "02"
) "-" "20"

q5. email address : personal_info@domain. The length of the personal_info may be upto 64 characters and domain name may be upto 253 characters.

personal-info contains

30th August, 2023

RegEx to F.A



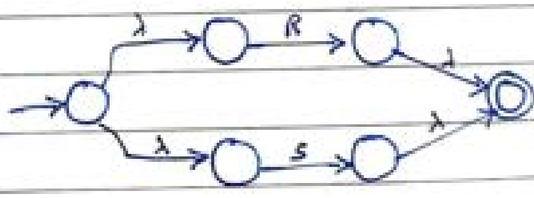
→ Thompson's Construction

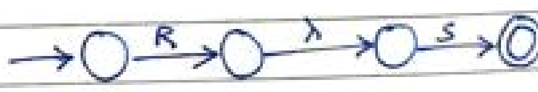
RegEx — F.A

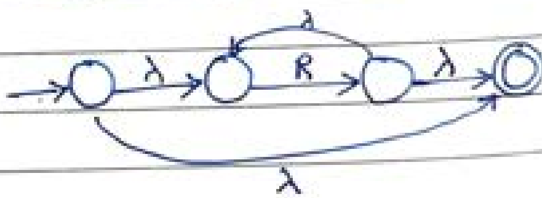
ϕ 

λ 

a 

R+S
(either R or S) 

R.S 

R^* 

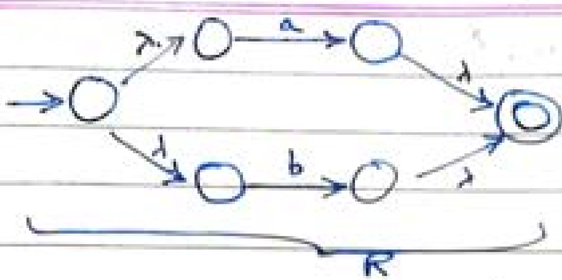
q1. $(a+b)^*.a$

↓

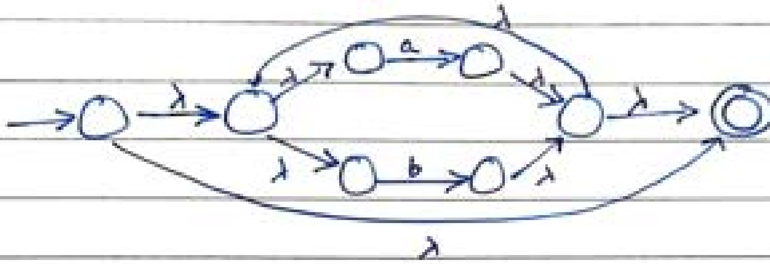
a 

b 

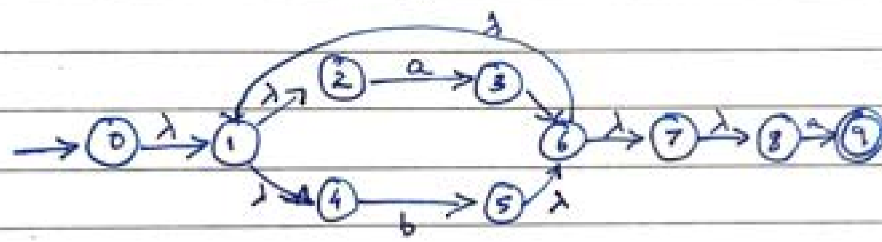
$(a+b)$



$(a+b)^*$



$(a+b)^* \cdot a$



q2. $(ab + ba)^*$

q3. $(aa)^* (bb)^*$

q4. $(aa)^* a + (bb)^* b$

q5. $a^* + b^* + c^*$

q6. $(a+\lambda)(1+\lambda)(1+2)^* 0 (2+1)^*$

q7. $(1+01)^* 00 (1+10)^*$

→ F.A to RegEx

State Elimination Algorithm

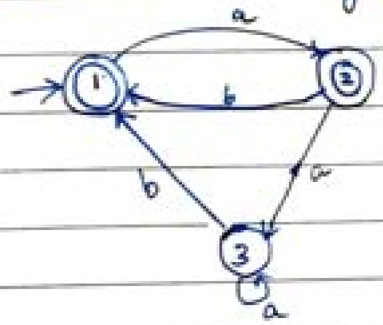
1. Identify unreachable states, if any remove it.

2. Identify if there are any incoming transitions on start state, if so create new state and connect it to old start state using λ.

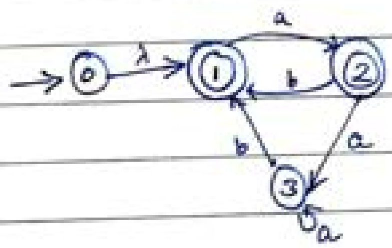
3. Identify if there are any multiple final states or if there are any outgoing transitions from final state, if so create new final state and connect old final state to new final state using λ.

4. Remove any arbitrary intermediate state and identify regular expression for that state. Repeat this step till only start and final state are remaining in FA

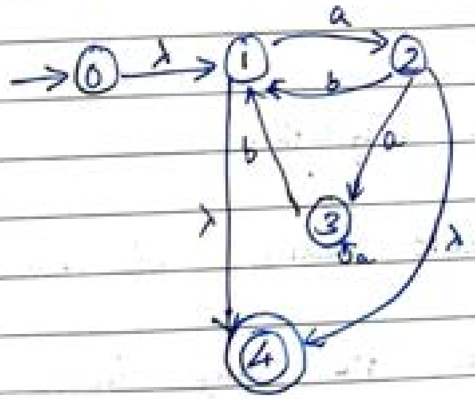
Q.1. Convert FA to RegEx



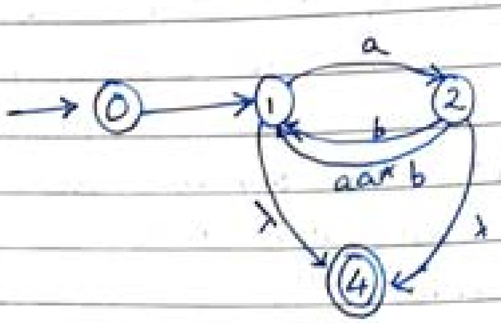
Sol. 2.



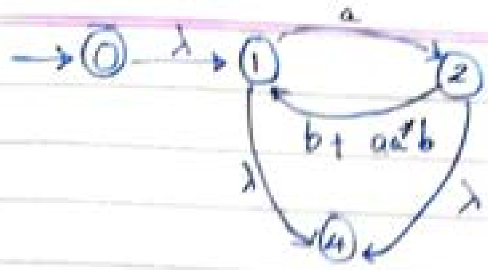
3.



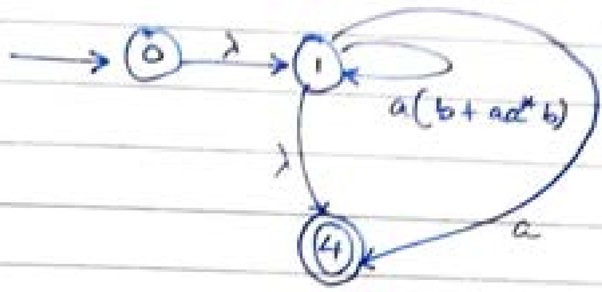
4.

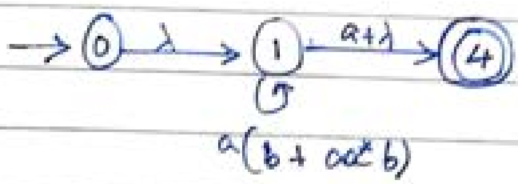


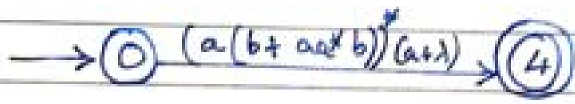
1-2-1
OR
1-2-4



5.





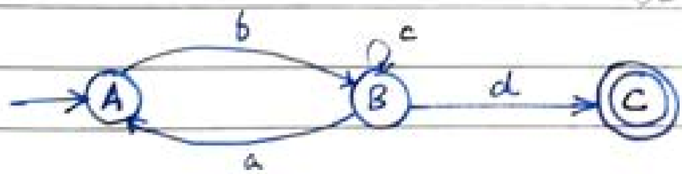


RegEx : $(a(b + aa^*b))^* (a + \lambda)$

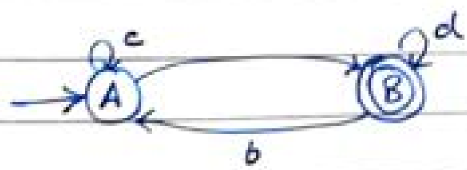
$(ab + aaa^*b)^* (a + \lambda)$

---

q2.

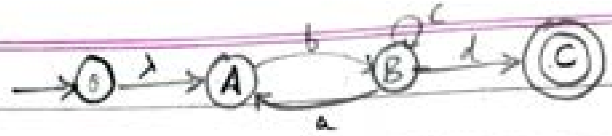


q3.

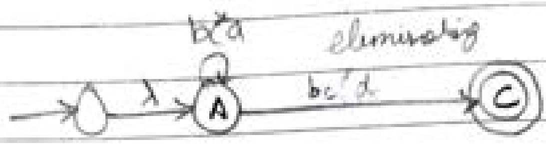


q4.

Sol 2.

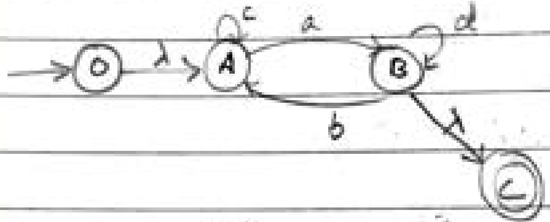


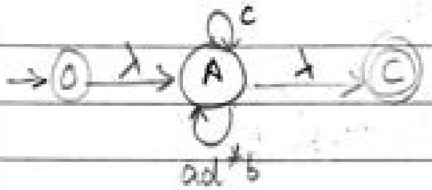
eliminating B



RegEx : $(bc^*a)^*(bc^*d)$

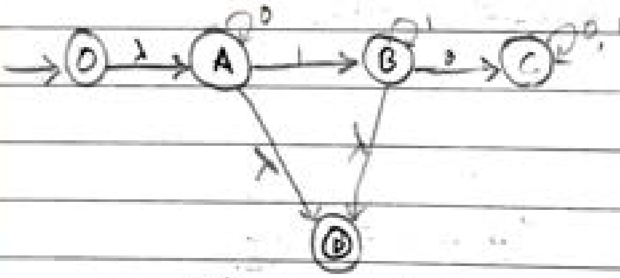
Sol 3.

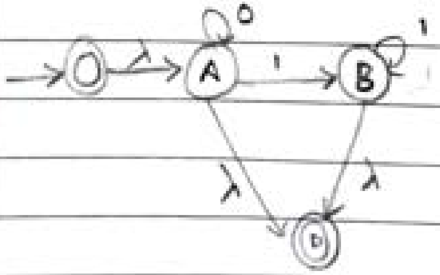




RegEx : $(c + ad^*b)^*$

Sol 4.





$0^*(11^* + \lambda)$

A - B
B - B
B - D

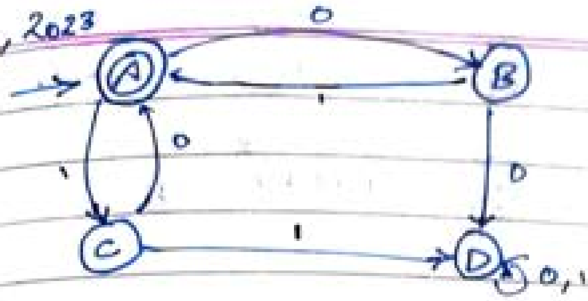
$11^* \Rightarrow 1^+$

$1^+ + \lambda \Rightarrow 1^*$

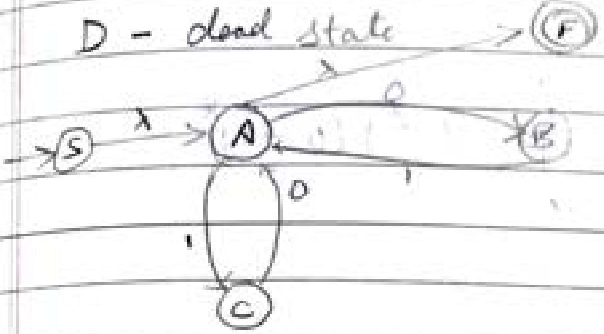
0^*1^*

1st September, 2023

Q5.



D - dead state

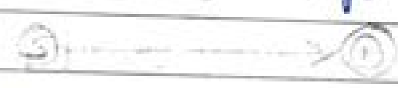


eliminating B

AB 0

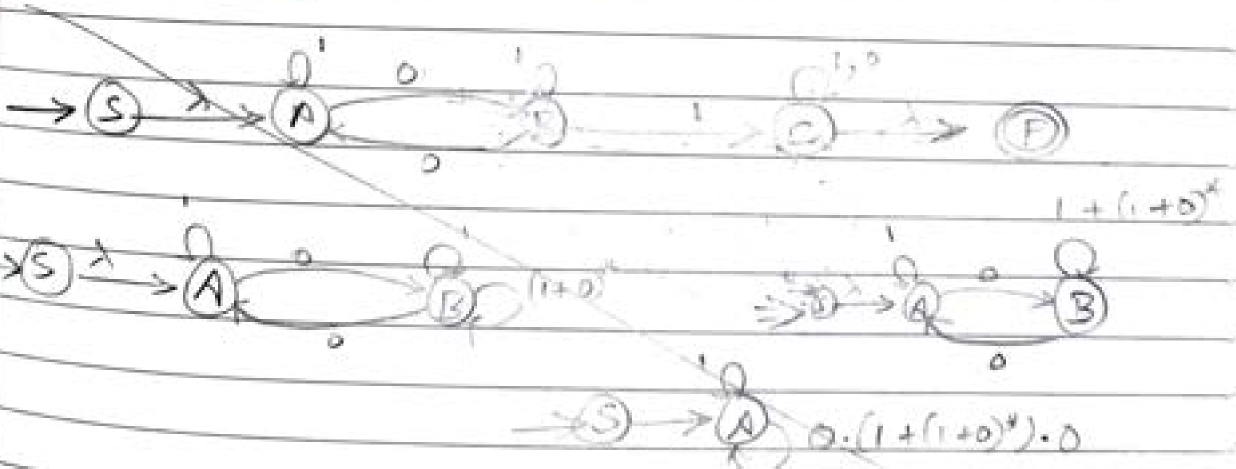
BA 1

$(01+10)^*$ - RegEx



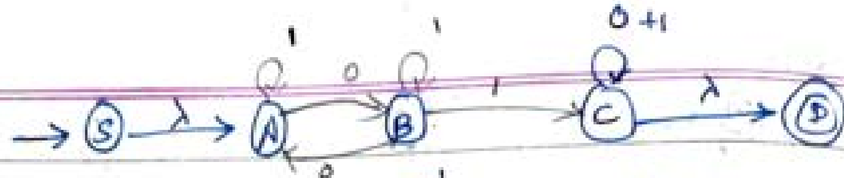
Sol.

Q6.



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

Sol.



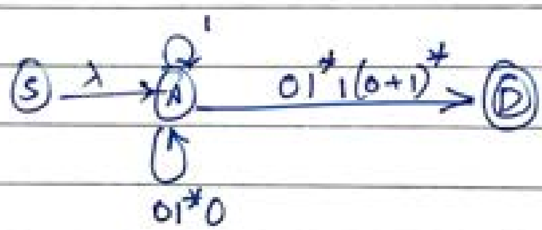
elim. order CBA

(1)

A – B	0	
B – B	1^*	$\Big\}$ $01^*1(0+1)^*$
B – D	$1\cdot(0+1)^*$	

(2)

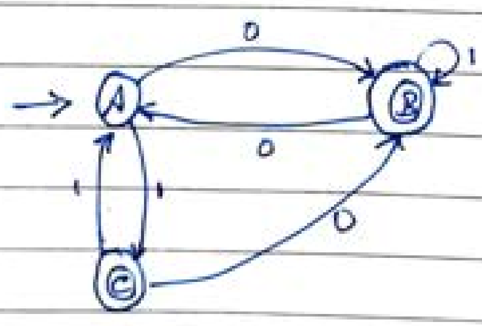
A – B	0	
B – B	1^*	$\Big\}$ 01^*0
B – A	0	

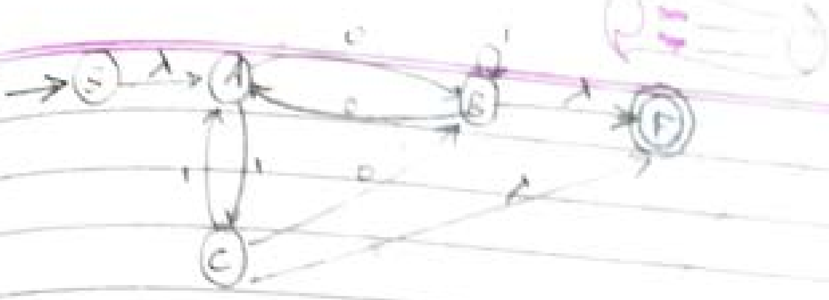


$$\Rightarrow (1+01^*0)^*\,01^*1\,(0+1)^*$$

$$\Rightarrow (1+01^*0)^*\,01^+\,(0+1)^*$$

q7.



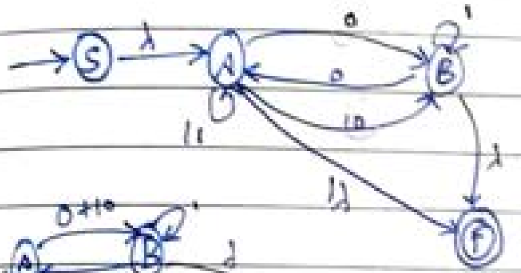


A-B
B-A } 01^*0

A-C
C-B

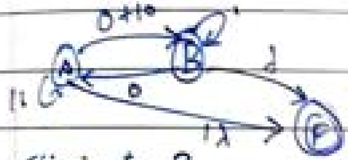
$(0+10)$

Eliminate C



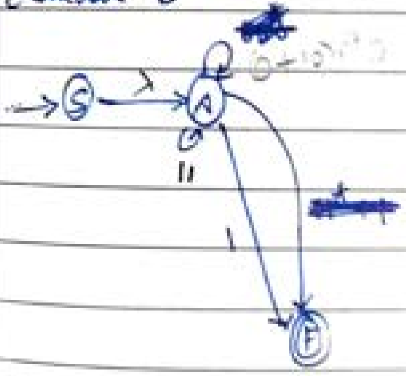
A-C
C-A } 11

A-C
C-F } 1λ



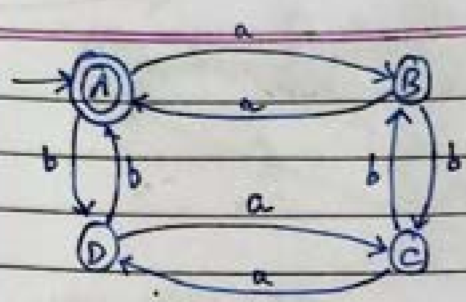
A-C
C-B } 10

Eliminate B



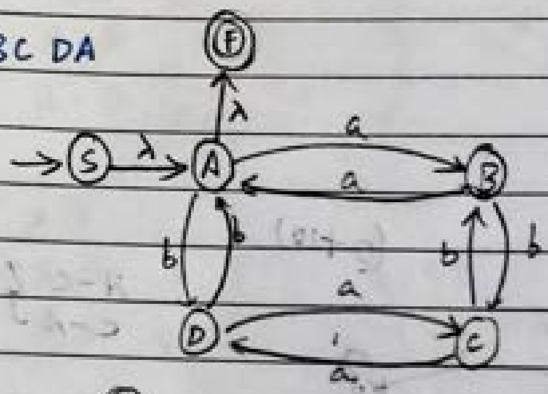
? • RegEx : $(((0+10)1^*0)+11)^*(0+10)1^* \;\; 1$

q8.



order of elimination : BCDA & BDCA

i) BC DA

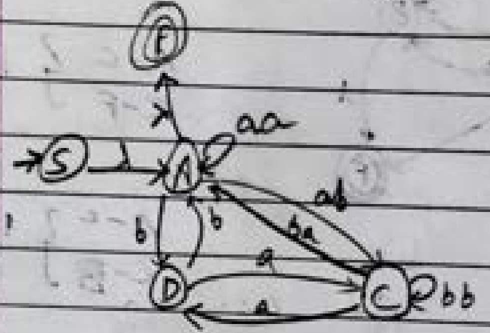


A-B, B-A } aa

A-B, B-C } ab

C-B, B-A } ba

C-B, B-C } bb

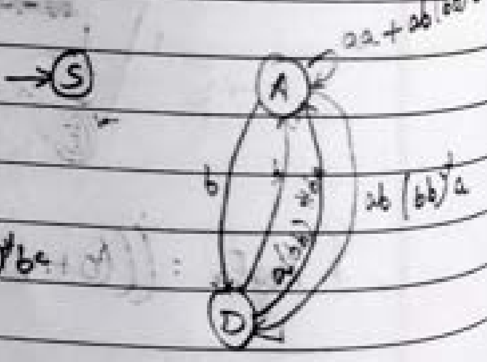


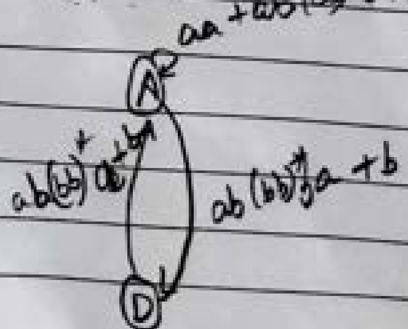
A-C, C-C, C-A } $ab(bb)^* ba$

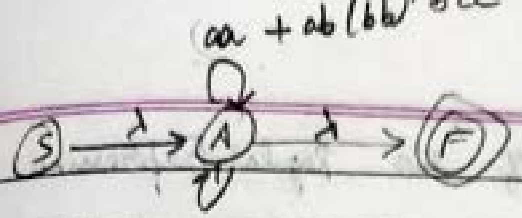
A-C, C-D } $ab(bb)^* a$

D-C, C-A } $a(bb)^* ba$

D-C, C-D } $a(bb)^* a$



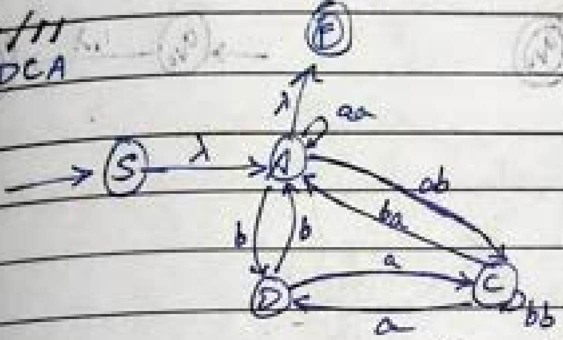




$(ab(bb)^*a+b)\cdot(ab(bb)^*ba+b)$

RegEx: $[(aa+ab(bb)^*ba)+(ab(bb)^*a+b)(ab(bb)^*ba+b)]^*$

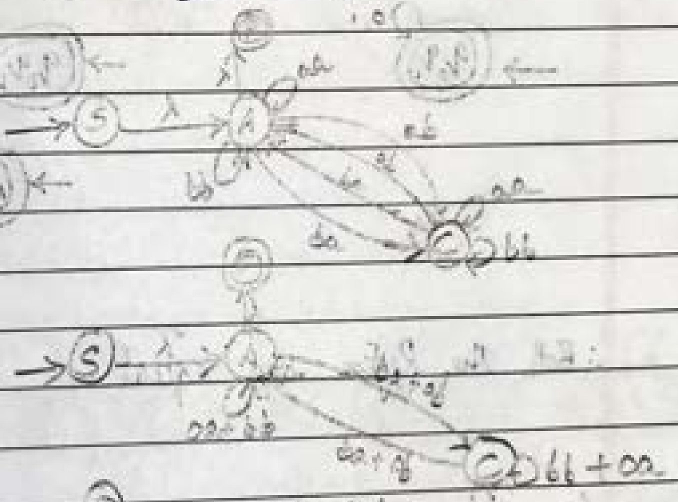
ii) BDCA

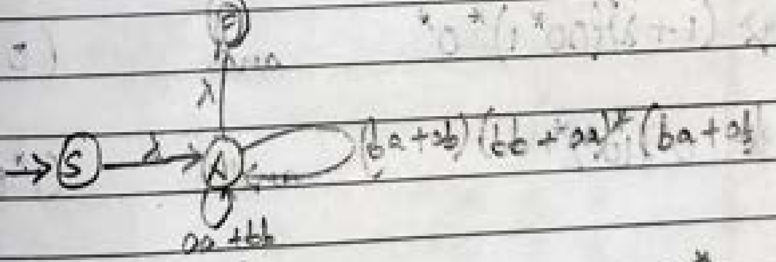


A-D } bb
D-A

A-D
D-C

A-D-C
C-D-A





RegEx: $[(ba+ab)(bb+aa)^*(ba+ab)+(aa+bb)]^*$

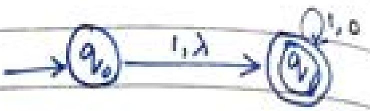
5th September, 2023

Equivalence of two Regular Expression

- → Formal
- → Informal

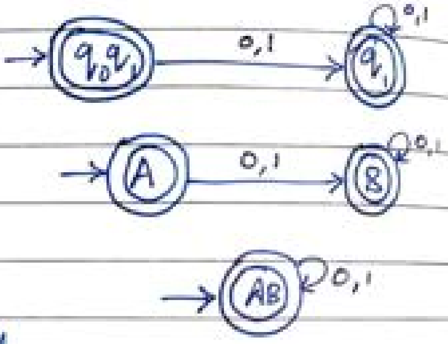
1. $(0+1)^*(0+\lambda)$ $\qquad\qquad$ $(1+\lambda)(1+0)^*$

NFA



TT

	λ	0	1
q_0	$q_0 q_1$	$q_0 q_1$	q_0
q_1	q_1	ϕ	ϕ



∴ Both the RegEx are equal

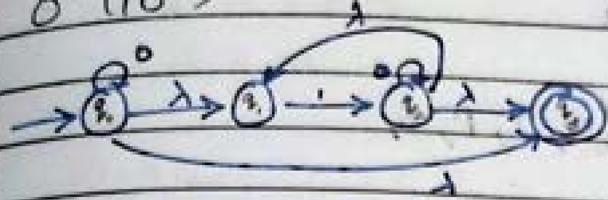
Q2. $(1+\lambda)(00^*1)^*0^*$ and $(0+\lambda)(11^*0)1^*$

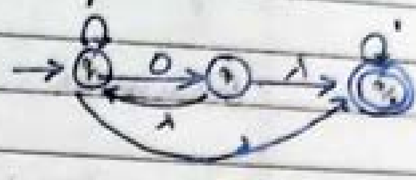
Q3. $0^*(10^*)^*$ and $(1^*0)^*1^*$

Sol 2. $(1+\lambda)(00^*1)^*0^*$ $\qquad\qquad$ $(0+\lambda)(11^*0)1^*$

$0^*(10^*)^*$ $\qquad\qquad$ $(1^*0)^*1^*$

NFA





	λ-closure	0	1
→ q_0	$q_0 q_1 q_3$	q_0	ϕ
q_1	q_1	ϕ	q_2
q_2	$q_1 q_2 q_3$	q_2	ϕ
* q_3	q_3	ϕ	ϕ

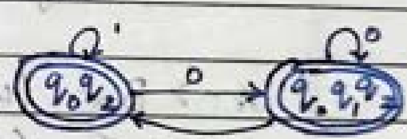
	λ-closure	0	1
q_0	$q_0 q_2$	q_1	q_0
q_1	$q_0 q_1 q_2$	ϕ	ϕ
q_2	q_2	ϕ	q_2

DFTT

	0	1
$q_0 q_1 q_3$	$q_0 q_1 q_3$	$q_1 q_2 q_3$
$q_1 q_2 q_3$	$q_1 q_2 q_3$	$q_1 q_2 q_3$

	0	1
$q_0 q_2$	$q_0 q_1 q_2$	$q_0 q_2$
$q_0 q_1 q_2$	$q_0 q_1 q_2$	$q_0 q_2$

DFA



	0	1
→* A	A	B
* B	B	B

	0	1
→* A	B	A
* B	B	A

* B $\boxed{}$ (AB) $\qquad$ * B $\boxed{}$ (AB)

A * $\qquad$ A *

	0	1
AB	AB	AB

	0	1
AB	AB	AB

=

6th September, 2023

Regular Grammar

Grammar $G = (V, T, P, S)$

$V \rightarrow$ Variable / Non Terminal - uppercase : S A B exps id

$T \rightarrow$ Terminal - lowercase, numbers : a b ... z, 0-9, +, -

$P \rightarrow$ Production / Rules - $\alpha \rightarrow \beta$ $\alpha, \beta \in (V \cup T)^*$

$S \rightarrow$ start symbol

* sentence : only terminals

Linear grammar	Non Linear grammar
only 1 non-Terminal	any no. of non-Terminal

Type 3 Regular
- → Right Linear
- → Left Linear (LLG)

$A \rightarrow Bw$ (LLG) (non-terminal on left)

$A \rightarrow wB$ (RLG) (non-terminal on right)

$A \rightarrow w$

Type 2: Context Free Grammar

$A \rightarrow \alpha$

$\alpha \in (V \cup T)^*$

Type 1: Context Sensitive Grammar

Type 0: Unrestricted

$\alpha \rightarrow \beta$

$\alpha \in (V \cup T)^+$

$\beta \in (V \cup T)^*$

Q1. Construct regular grammar for $L = \{a\}$

$G = (V, T, P, S)$

$V = \{S\}$

$T = \{a\}$

$P = \{S \rightarrow a\}$

$S \rightarrow$ start symbol

q2. $L = \{a^n \mid n \geq 0\}$

$\lambda, a, aa, aaa \ldots$

$S \to aS$
$S \to \lambda$

q3 $L = \{a^n \mid n \geq 1\}$

$a, aa, aaa \ldots$

$S \to aS$
$S \to a$ } $S \to aS \mid a$

q4. RegEx $(a+b)^*$

$S \to aS \mid bS \mid \lambda$

q5. RegEx $(a+b)^+$

$S \to aS \mid bS \mid a \mid b$

q6. $(aa)^*$ $\quad S \to aaS \mid \lambda$

$(ab)^*$

$S \to abS \mid \lambda$

q7 $(aa)^* a$ $\quad S \to aaS \mid a$

q8 $a^* b^*$ $\quad P = \{$

a^* $\quad A \to aA \mid \lambda \mid bB$
b^* $\quad B \to bB \mid \lambda$
$\}$

$S \to aA \mid \lambda \mid bB$

q9. $b^* (a+b)^*$

$S \to bS \mid \lambda$
$A \to aA \mid bA \mid \lambda$

$\therefore S \to bS \mid \lambda \mid aA \mid bA$

q10. $\boxed{b^*} a \boxed{(a+b)^*}$

$A \to bA \mid \lambda$
$B \to aB \mid bB \mid \lambda$

$P = \{$
$A \to bA \mid aB$
$B \to aB \mid bB \mid \lambda$
$\}$ $\quad A \to B$
$\lambda \to$

$A \to bA \mid \cancel{\lambda}$
$B \to aB \mid bB \mid \lambda$

Finite Automata to Regular Grammar

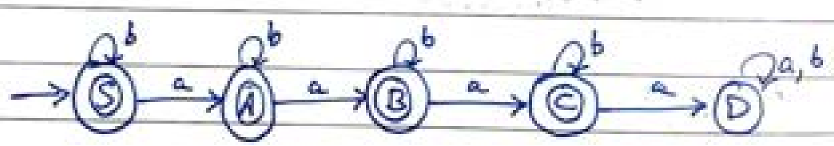
M	G
$Q \longleftrightarrow$	V
$\Sigma \longleftrightarrow$	T
δ	P
$q_0 \longleftrightarrow$	S
F	

0) Remove dead and unreachable states
1) Represent states as variables
2) Represent alphabets as Terminals
3) Start state as start symbol
4) Convert transition to production

$$\delta(q_0, a) = q_1 \quad \Rightarrow \quad q_0 \rightarrow a q_1$$

final state introduce production $q_f \rightarrow \lambda$

q.1.



$V = \{S, A, B, C\}$
$T = \{a, b\}$ S is start symbol

$\delta(q_0, a) = q_1$ $q_0 \rightarrow a q_1$ S, A, B, C final sta

$\delta(S, b) = S$	$S \rightarrow bS$	$S \rightarrow \lambda$
$\delta(S, a) = A$	$S \rightarrow aA$	
$\delta(A, b) = A$	$A \rightarrow bA$	$A \rightarrow \lambda$
$\delta(A, a) = B$	$A \rightarrow aB$	
$\delta(B, a) = C$	$B \rightarrow aC$	$B \rightarrow \lambda$

$\delta(B, b) = B$ $\quad$ $B \to bB$

$\delta(C, b) = C$ $\quad$ $C \to bC$ $\quad$ $C \to \lambda$

$P = \{$

$S \to aA \mid bS \mid \lambda$

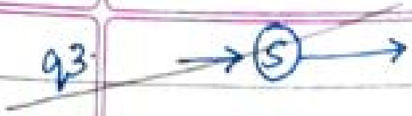
$A \to aB \mid bA \mid \lambda$

$B \to aC \mid bB \mid \lambda$

$C \to bC \mid \lambda$

$\}$

72

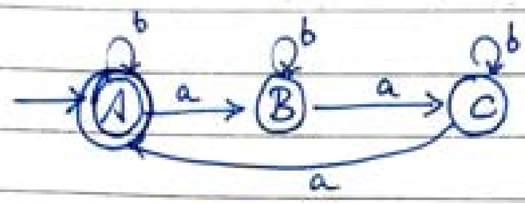
q3. 

Regular Grammar to Finite Automata

q1. $A \rightarrow aB \mid bA \mid b$ — if start symbol not mentioned first production is start symbol

$B \rightarrow aC \mid bB$

$C \rightarrow aA \mid bC \mid a$



implicitly identifying λ

$A \rightarrow bA \quad A \rightarrow \lambda \quad A \rightarrow b$

$C \rightarrow aA \quad C \rightarrow \lambda \quad C \rightarrow a$

$\therefore$ A is final state

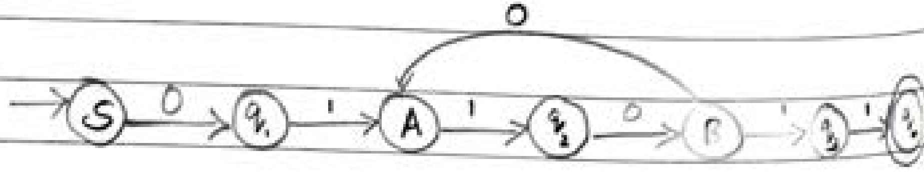
$B \rightarrow aC$ replacing C with λ — don't care no λ $B \rightarrow a$?

$C \rightarrow bC$ — $C \rightarrow b$

q2. $S \rightarrow 01A$

$A \rightarrow 10B$

$B \rightarrow 0A \mid 11$



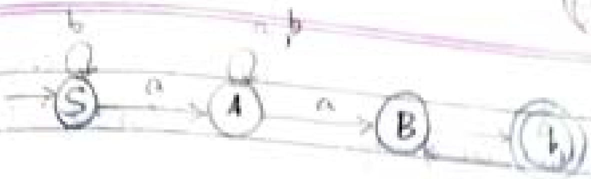
q3. $S \rightarrow bS \mid aA$

$A \rightarrow aA \mid bA$

$A \rightarrow aB$

$B \rightarrow bbB$

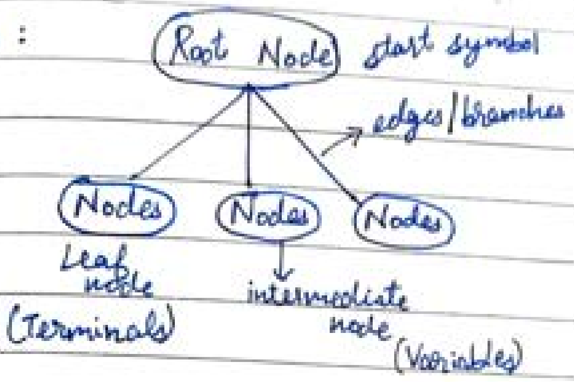
$B \rightarrow \lambda$



th September, 2023

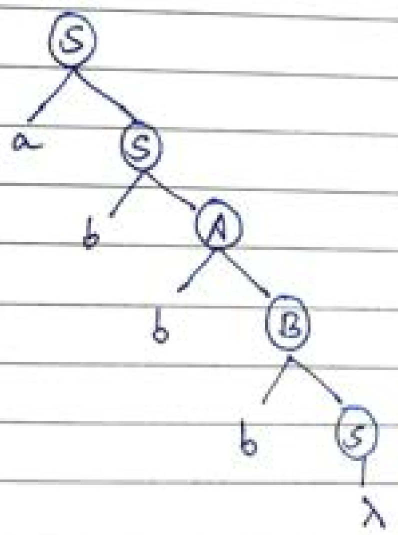
Aspects Of Grammar

ParseTree :



eg: $S \rightarrow aS \mid bA \mid \lambda$
$A \rightarrow aA \mid bB$
$B \rightarrow aB \mid bS$

abbb



Parsing : Process of determining if a string belongs to language or not.

- → Top - Down (easy)
- → Bottom - Up (hard)

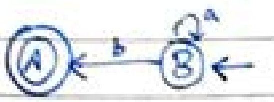
→ LLG
(constraint on ending)

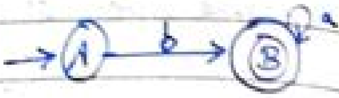
RLG
(constraint on starting)

$$A \rightarrow Bb$$
$$B \rightarrow Ba \mid \lambda$$

$\neq$

$$A \rightarrow bB$$
$$B \rightarrow aB \mid \lambda$$





9th September, 2023

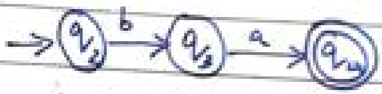
UNIT - 2

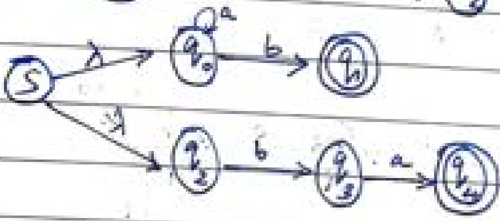
Properties Of Regular Lang
- → Closure
- → Decisive - Pumping Lemma

Context Free Grammar

Pushdown Automata
- → DPDA
- → NPDA

q1 $L_1 = \{a^n b\}$ $L_2 = \{ba\}$ P.T $L_1 \cup L_2$ is regular



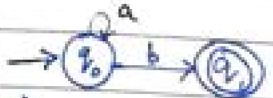


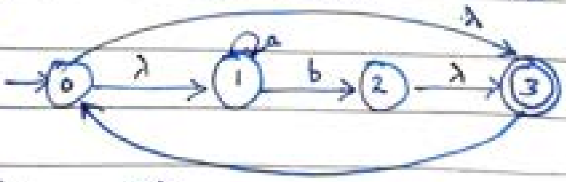
* Complement (N^c) ✗ NFA
✓ DFA

to find N^c draw DFA

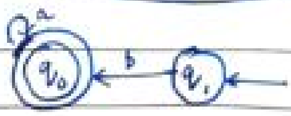
q2 $L_1 \cdot L_2$ is regular

q3 L^* is regular $L = \{a^n b\}$





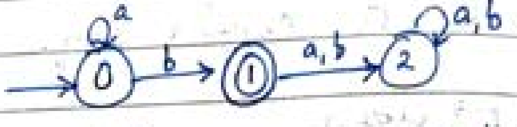
q4 L^R is regular



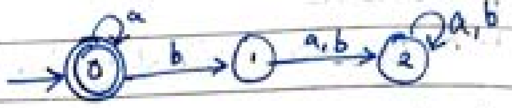
q5. L^c is regular $\quad L = \{a^n b\}$

make DFA

L :



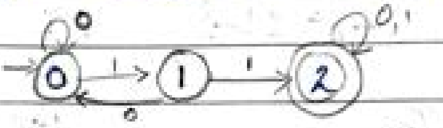
L^c :

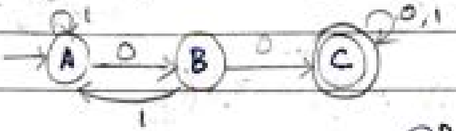


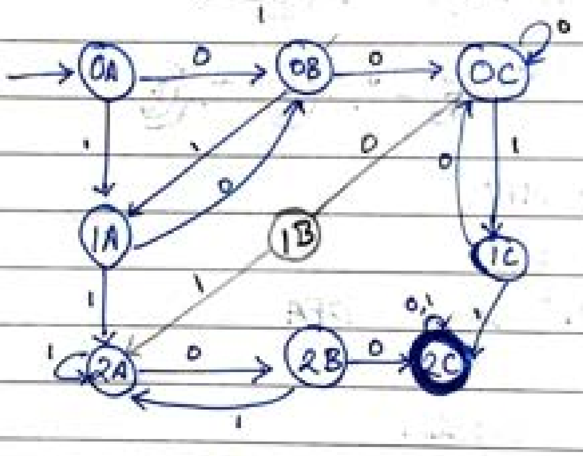
q6. $L1 = \{w \mid w$ contains '11' as substring$\}$

$L2 = \{w \mid w$ contains '00' as substring$\}$

P.T $L1 \cap L2$ is regular







12th September, 2023

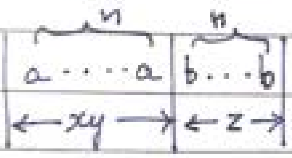
Pumping Lemma (n = no. of states / pumping constant)

Q1. P.T $L = \{a^n b^n \mid n \geq 0\}$ is not regular

Sol 1) Assume lang is regular
2) Assume no. of states as n
3) Select string $\in L$ such that $|w| \geq n$

$$w = a^n b^n \quad , |w| = 2n \geq n$$

4) Divide string into 3 parts xyz,
such that $|y| \geq 1$ & $|xy| \leq n$



$$\left.\begin{array}{l} x = a^{n-1} \\ y = a \end{array}\right\} a^n \qquad w = \underset{x}{a^{n-1}}\ \underset{y}{a}\ \underset{z}{b^n}$$
$$z = b^n$$

5) acc. to Pumping property if L is regular,
$w \in L$ such that $|w| \geq n$,
$w = xyz$ such that $|xy| \leq n$ & $|y| \geq 1$
then apply pumping property

$$xy^iz \in L \quad \text{for } i = 0, 1, 2 \ldots$$

$i = 0$

$$w = a^{n-1} b^n \notin L$$

This is contradiction to the statement
L is regular.
Hence Proved that $L = \{a^n b^n \mid n \geq 0\}$
is not a regular language.

q2. $L = \{wcw^R \mid w \in (a,b)^*\}$

let $w = a^n b^n c b^n a^n$

$|w| = 4n+1 \geq n$

$\underbrace{a^n b^n}\ c\ \underbrace{b^n a^n}$

$|y| \geq 1$

$|xy| \leq n$

$a^{n-1} a b^{n-1}\ b\ c\ b^n a^n$

$x = a^{n-1}$

$y = a$

$|y| \geq 1 \quad |xy| \leq n$

$i = 0$

$xy^iz = a^{n-1}(a)^0 b^n c b^n a^n$

$= a^{n-1} b^n c b^n a^n \notin L$

13th September, 2023

q3. $L = \{ww^R \mid w \in (a,b)^*\}$

let $w = a^n b^n b^n a^n$

$|w| = 4n \geq n$

$x = a^{n-1}$

$y = a$

$|y| \geq 1 \quad |xy| \leq n$

$xy^iz = a^{n-1}(a^i)\, b^n b^n a^n$

$i=0 \quad a^{n-1} b^n b^n a^n \quad \notin L$

Q4. $L = \{ a^{n^2} \mid n \geq 0 \}$

$|w| = n^2 \geq n$ $\quad$ | $aa \cdot a^{n^2-2}$

$a \cdot a^{n^2-3}\, a$

$x = a$ $\quad$ $x = \lambda$

$y = a$ $\quad$ (or) $\quad y = a$

$z = a^{n^2-2}$ $\quad$ $z = a^{n^2-1}$

$|y| \geq 1 \qquad |xy| \leq n$

$xy^iz = a\, a^i\, a^{n^2-2}$

$i=0 \; ; \quad a\, a^{n^2-2} = a^{n^2-1} \notin L$

Q5. $L = \{ a^n \mid n$ is composite number $\}$

$L^c = \{ a^n \mid n$ is prime number $\}$

13th September, 2023

Context Free Grammar (CFG)

→ can be linear or non-linear
→ can be ambigious or unambigious

$G = (V, T, P, S)$

variable Terminal Production start symbol

* left side should be only 1 non-terminal

→ Linear CFG

q1. Construct CFG for lang $L = \{a^n b^n \mid n \geq 0\}$

$S \to aSb \mid \lambda$

q2. $L = \{a^n b^n \mid n \geq 1\}$ CFG: $S \to aSb \mid ab$

q3. $L = \{w c w^R \mid w \in (a,b)^*\}$ CFG: $S \to aSa \mid bSb \mid c$

q4. $L = \{w w^R \mid w \in (ab)^* \mid (ba)^*\}$ CFG: $S \to abSba \mid baSab \mid \lambda$

q5. $L = \{w w^R \mid w \in (a,b)^*\}$ CFG: $S \to aSa \mid bSb \mid \lambda$

q6. $L = \{a^n w w^R b^n \mid w \in (a,b)^*, n \geq 0\}$ CFG: $S \to aSb \mid A$

$A \to aAa \mid bAb \mid \lambda$

q7. $L = \{a^n b^{n+1} \mid n \geq 0\}$ CFG: $S \to aSb \mid b$

q8. $L = \{a^{n+2} b^n \mid n \geq 1\}$ CFG: $S \to aSb \mid aaab$

$S \to aSb \mid aa$

q9. $L = \{a^n b^{2n} \mid n \geq 0\}$ CFG: $S \to aSbb \mid \lambda$

q10. $L = \{a^n b^{n-3} \mid n \geq 3\}$ CFG: $S \to aSb \mid aaa$

aaa a^3

$a\, a^3 b$

$a^2 a^3 b^2$

q11. $L = \{a^n b^m \mid n > m\}$ CFG: $S \to aSb \mid aS \mid a$

q12. $L = \{a^n b^m \mid n \neq m\}$ CFG:

$S \to aSb \mid A \mid B$
$A \to aA \mid a$
$B \to Bb \mid b$

q13. $L = \{a^n b^m \mid n = 2 + (m \bmod 3)\}$ CFG:

0, 1, 2
2a, 3a, 4a

b	a	
0 + 2	2a	aa
1 + 2	3a	aab
2 + 2	4a	aaaabb

$S \to aaA \mid aaabA \mid aaaabbA$
$A \to bbbA \mid \lambda$

m = 3, n = 2+0 $\boxed{aabbb}$ m = 4, n = 2+1 aaab$\boxed{bbb}$

q14. $L = \{a^n b^m \mid n \neq 2m\}$

m = 1 n ≠ 2
b ab aaab aaaab
m = 2 n ≠ 4
bb abb aabb aaabb
m = 3 n ≠ 6
bbb abbb aabbb aaabbb

$S \to aaSb \mid A \mid B \mid aC$
$A \to aA \mid a$
$B \to Bb \mid b$
$C \to bb \mid \lambda$

q15. $L = \{a^{n+2} b^m \mid m > n, n \geq 0\}$

n m
0 >0 $a^2 b b^4$
1 >1 $a a^2 b b b'$
2 >2 $aa\, a^2 b\, bbb'$

aab

$S \to aSb \mid aab \mid Sb$

q16. $L = \{a^n b^m c^m d^n \mid n, m \geq 1\}$

$S \to aSd \mid aAd$
$A \to bAc \mid bc$

$a^n b^m c^{n+m}$

q17. $L = \{a^n b^m c^k, k = n + m, n, m, k \geq 0\}$

$a^n b^m c^m c^n$ λ, ac, bc, abcc

$S \to aSc \mid A$
$A \to bAc \mid \lambda$

q18. $L = \{a^n b^m c^k, k = 2, m = 2n, n \geq 0\}$ $S \to AB$

$a^n b^{2n} c^2$

$A \to aAbb \mid \lambda$
$B \to cc$

$a^n b^m c^{n+2m}$ $\quad S \to aSc \mid A$

$a^n b^m c^{2m} c^n$ $\quad A \to bAcc \mid \lambda$

q19. $L = \{a^n b^m c^k,\ m, n \geq 0,\ k = n + 2m\}$

$S \to aaaaaaS \mid aa \mid aaa \mid aaaa \mid aaaaa$

q20. $L = \{|w| \bmod 3 \neq |w| \bmod 2,\ w \in (a/b)^*\}$

	mod 2	mod 3
a^0	0	0
a^1	1	1
a^2	0	2
a^3	1	0
a^4	0	1
a^5	1	2
a^6	0	0
a^7	1	1
a^8	0	2
a^9	1	0

$L = \{a^2, a^3, a^5, a^8, a^9, a^{10}, a^{11}\}$

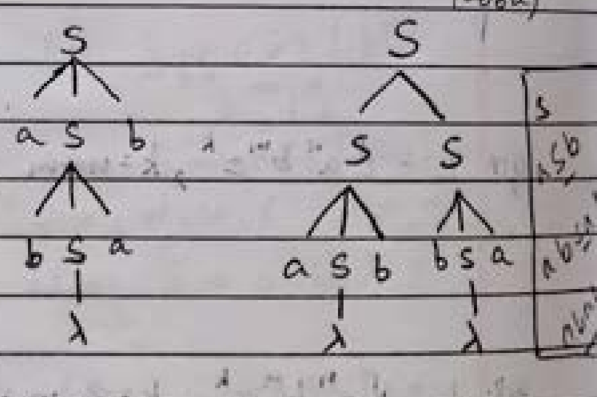
15th September, 2023

→ Non-Linear CFG

q1. $L = \{n_a(w) = n_b(w) \mid w \in (a, b)^*\}$

aabb
bbaa
abab
baba
baab
λ, ab, ba

(abba)



$S \to SS$
$S \to aSb \mid \lambda \mid bSa$

q2. $L = \{uvwv^R \mid |u| = |w| = 2,\ |v| > 1,\ u, v, w \in (a,b)^*\}$

$S \rightarrow AB$
$A \rightarrow aa \mid ab \mid ba \mid bb$ // length = 2
$B \rightarrow aBa \mid bBb \mid aa \mid bb$ // vv^R

$\Big\}\ uvv^R$

$\overset{A}{u}\,\overbrace{v\,\underset{A}{w}\,v^R}^{B}$

$S \rightarrow AB$
$A \rightarrow aa \mid ab \mid ba \mid bb$
$B \rightarrow aBa \mid bBb \mid aAa \mid bAb$

q3. $L = \{n_a(w) = n_b(w) + 1,\ w \in (a,b)^*\}$

$S \rightarrow AaA$
$A \rightarrow aAb \mid AA \mid bAa \mid \lambda$

q4. $L = \{n_a(w) = 2 \times n_b(w),\ w \in (a,b)^*\}$
λ, aab, baa, aba

aab baa
$S \rightarrow SS$

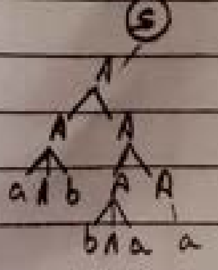
$S \rightarrow \lambda \mid SS$
$S \rightarrow aSaSb$
$S \rightarrow bSaSa$
$S \rightarrow aSbSa$

q5. $L = \{n_a(w) > n_b(w),\ w \in (a,b)^*\}$

$S \rightarrow AaA$
$A \rightarrow aAb \mid AA \mid bAa \mid \lambda \mid aA \mid Aa$

q6. $L = \{n_a(w) \neq n_b(w),\ w \in (a,b)^*\}$

$S \rightarrow AaA \mid BbB$
$A \rightarrow aAb \mid AA \mid bAa \mid \lambda \mid aA \mid Aa$
$B \rightarrow aBb \mid BB \mid bBb \mid \lambda \mid bB \mid Ba$



q7. $L = \{a^n b^n \cup a^n b^{2n}\}$

$S \rightarrow S_1 | S_2$
$S_1 \rightarrow aS_1a | \lambda$
$S_2 \rightarrow aS_2bb | \lambda$

q8. equal parenthesis

$S \rightarrow (S) | SS | \lambda | [S] | \{S\}$

q9. if condition

if (condition)
{
statements
}

Statement → if Cond^n then Statement
Statement → { statement }
Cond^n → b

• if
$S \rightarrow iCtS | a$
$C \rightarrow b$
• if else
$S \rightarrow iCtS | a$
$S \rightarrow iCtSeS$
$C \rightarrow b$

q10. while loop

while (cond)
{
statement
}

$S \rightarrow wCS | a$
$C \rightarrow b$

q11. Arithmetic Expression

$E \to id \mid E+E \mid E-E \mid E \times E \mid E/E \mid E\%E \mid (E)$

→ Declaration statement

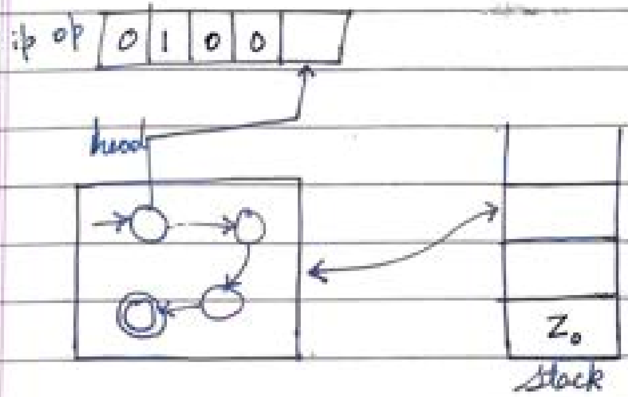
int id, id;

$D \to TL;$

$T \to$ int | float | char

$L \to id \mid id, L$

Pushdown Automata : FA + stack



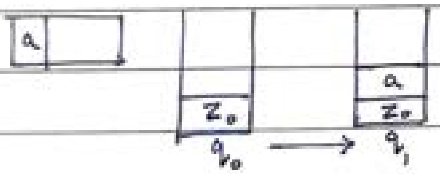
$Q \times \Sigma \times \Gamma$ to $Q \times \Gamma$

δ(cur state, ip alp, TOS)
= next state $\times \Gamma$

→ acceptance PDA : Final state / empty stack

→ Transition / Operation

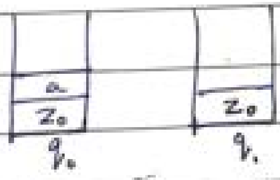
Push



$\delta(q_0, a, Z_0) = (q_1, aZ_0)$

new TOS   old TOS

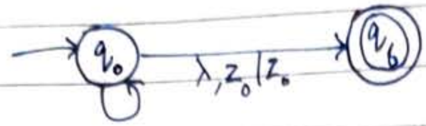
Pop



$\delta(q_0, a, a) = (q_1, \lambda)$

20th September, 2023

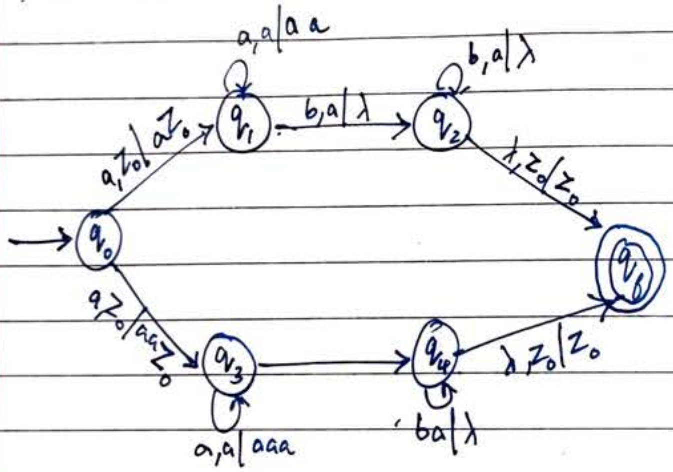
5. $n_a(w) = n_b(w)$



$b, a | \lambda$
$a, b | \lambda$
$b, b | bb$
$a, a | aa$
$b, Z_0 | bZ_0$
$a, Z_0 | aZ_0$

NPDA

6. $L = \{a^n b^n \mid n \geq 1 \cup a^n b^{2n} \mid n \geq 1\}$



27th September, 2023

Model of PDA

Transition → current state, current I/P symbol, current TOS

Operation → Push
→ Pop

λ-NFA + stack

PDA := $(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

- Q : states of λ-NFA
- Σ : input alphabet
- Γ : stack symbols
- δ : transition func.
- q_0 : start state
- z_0 : initial stack top symbol
- F : final / accepting states

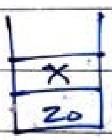
DPDA

$\delta: Q \times (\Sigma \cup \lambda) \times \Gamma \rightarrow Q \times \Gamma^*$

NPDA

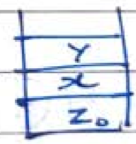
$\rightarrow 2^{Q \times \Gamma^*}$

→ Push operation



$$q \xrightarrow{(a, X \mid YX)} p$$

current ip symb, current ToS, new ToS, prev ToS



eg: $(a, X \mid ZYX)$

→ Pop operation

only 1 symb can be popped at a time

$$q \xrightarrow{(a, X \mid \lambda)} p$$

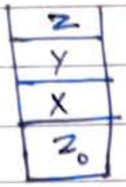
→ Instantaneous Description (ID) (Snapshot)

(q, w, u) (current state, leftover i/p, stack contents)

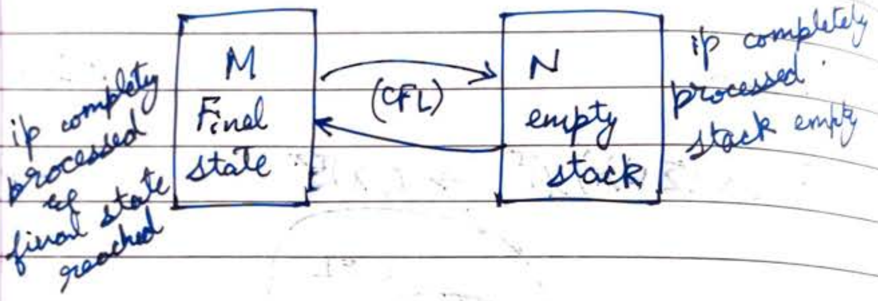
$\to (Z\,Y\,X)$

TOS

eg:

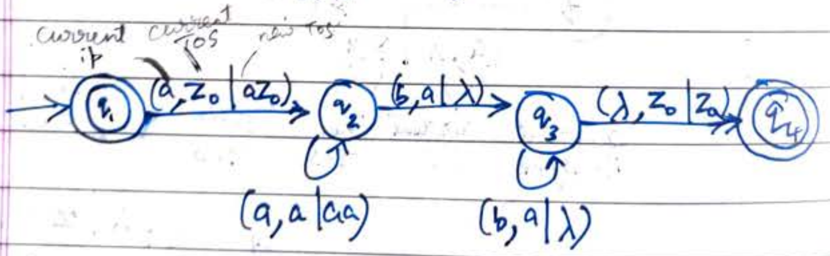


→ Acceptance PDA



eg 1. $a^n b^n,\ n \geq 0 \to \lambda$

$\underbrace{aa}_{\text{Push all } a\text{'s}}\ \underbrace{bb}_{\text{pop an 'a' for each } b}$



Transition function

$\delta(q_1, a, Z_0) = (q_2, aZ_0)$

current state, current i/p, current TOS; Transit to this state, new TOS

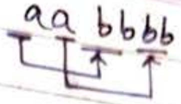
$\delta(q_2, a, a) = (q_2, aa)$

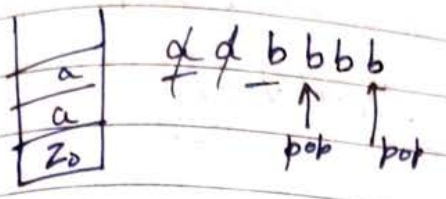
$\delta(q_2, b, a) = (q_3, \lambda)$

$\delta(q_3, b, a) = (q_3, \lambda)$

$\delta(q_3, \lambda, Z_0) = (q_4, Z_0)$

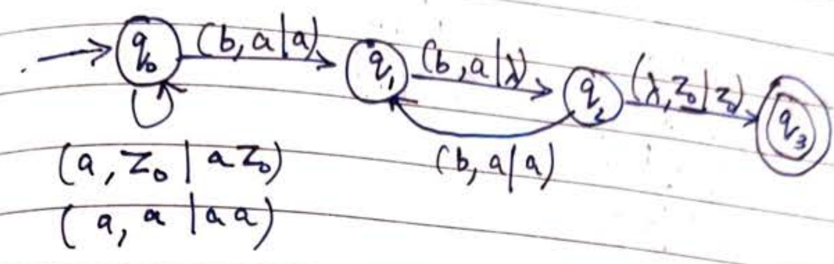
2. $a^n b^{2n}$, $n \geq 1$



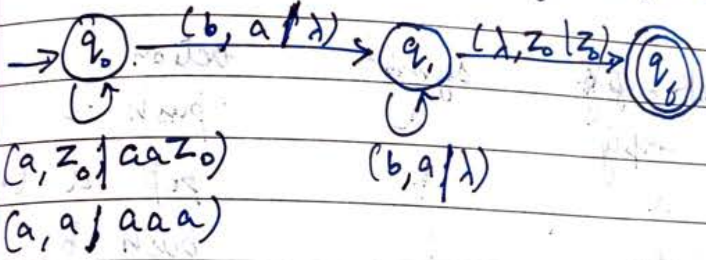


poping alt b s

Sol i)



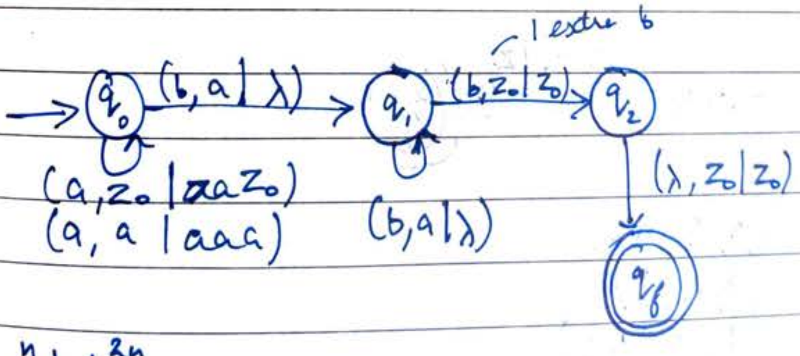
ii) a a b b b b ; for every a push 2 a s.



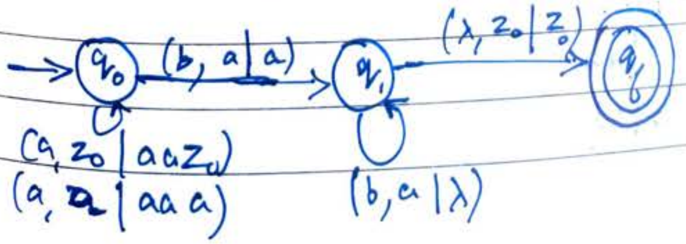
3. $a^n b^{2n+1}$; $n \geq$ [illegible]

$\Downarrow$

$\underbrace{a^n b^{2n}}_{\text{old soln}} \underbrace{b}_{\text{extra}}$

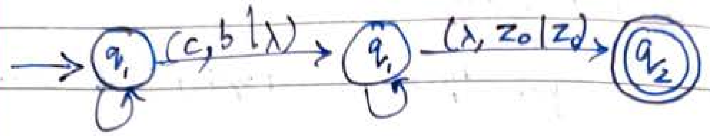


4. (g) $a^n b\, b^{2n}$



5. $a^n b^m c^{m+n}$

no. of cs = no. of as + no. of bs



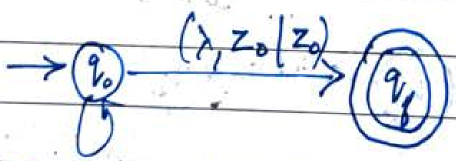
$(a, z_0 | a z_0)$ $\quad$ $(c, b | \lambda)$
$(a, a | aa)$ $\quad$ $(a, a | \lambda)$
$(b, a | ba)$
$(b, b | bb)$

6. $n_a(w) = n_b(w)$

a's & b's can occur in any order

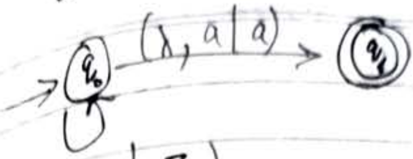
stack ~~empty~~	symbol	action
empty	a	push a
a	b	pop a (match)
empty	b	push b
b	a	match pop b
a	a	push a
b	b	push b

* push symbol if there is no chance of popping



$(a, z_0 | a z_0)$
$(b, a | \lambda)$
$(b, z_0 | b z_0)$
$(a, b | \lambda)$
$(a, a | aa)$
$(b, b | bb)$

$n_a(w) > n_b(w)$



1.

$(a, Z_0 \mid aZ_0)$
$(b, a \mid \lambda)$
$(b, Z_0 \mid bZ_0)$
$(a, b \mid \lambda)$
$(a, a \mid aa)$
$(b, b \mid bb)$

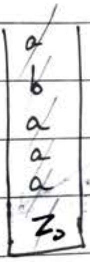
→ NDPA : same symbol, same TOS Two diff actions

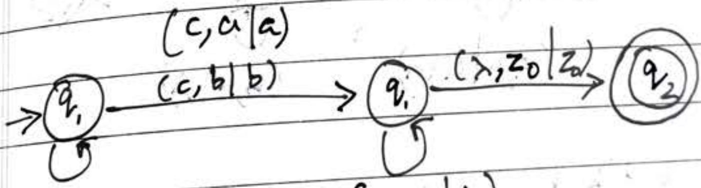
1. $L = \{wcw^R : w \in \{a, b\}^*, |w| \geq 1\}$

DPA

$\underbrace{ab}_{w}\ c\ \underbrace{ba}_{w^R}$

Push
$\boxed{aaaba}$ c abaaa

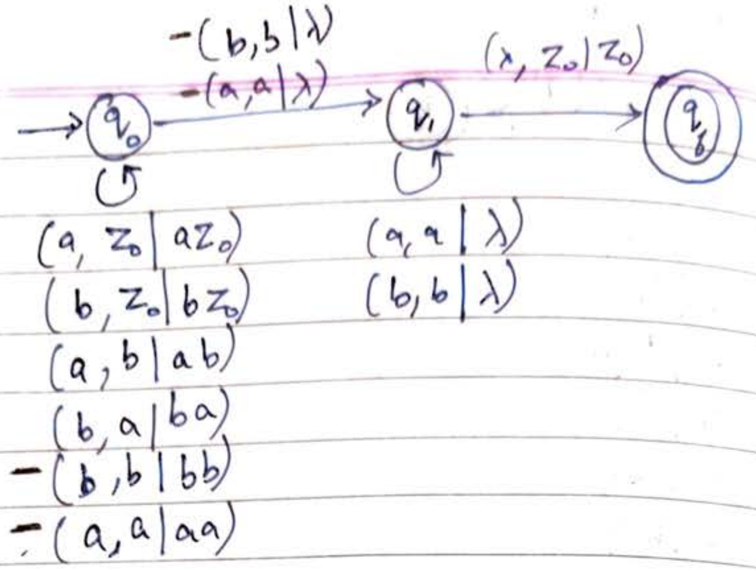




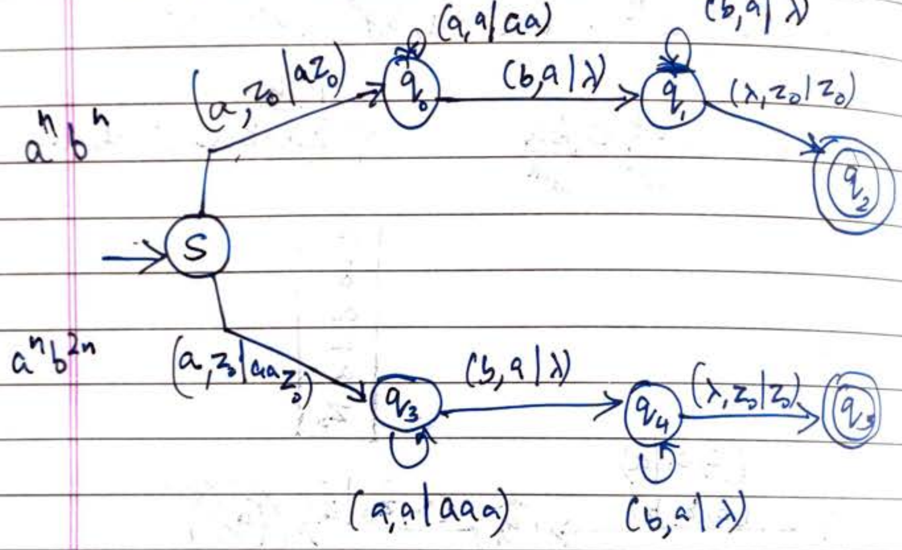
$(a, Z_0 \mid aZ_0)$
$(b, Z_0 \mid bZ_0)$
$(a, a \mid aa)$
$(b, b \mid bb)$
$(a, b \mid ab)$
$(b, a \mid ba)$

$(a, a \mid \lambda)$
$(b, b \mid \lambda)$

2. ww^R NPDA



3. $L = \{a^n b^n \cup a^n b^{2n}, n \geq 1\}$



4. $L = \{a^i b^j c^k d^l ;\ i = k \text{ or } j = l,\ i, k, j, l \geq 1$