THEORY OF COMPUTATION

Date / /
Page

Remarks

7-12 Ques.

F.A. & R.L. - 60%

PDA & CFL - 20%

LBA & CSL - 0%

TM & REL } 20%
Undecidability }
Complexity Theory }

TOC:

1. It is the mathematical study of compating M/C's and its capabilities, means observation before go for practical.

2. It is the study of F.L. & Automata theory.

F.L.:

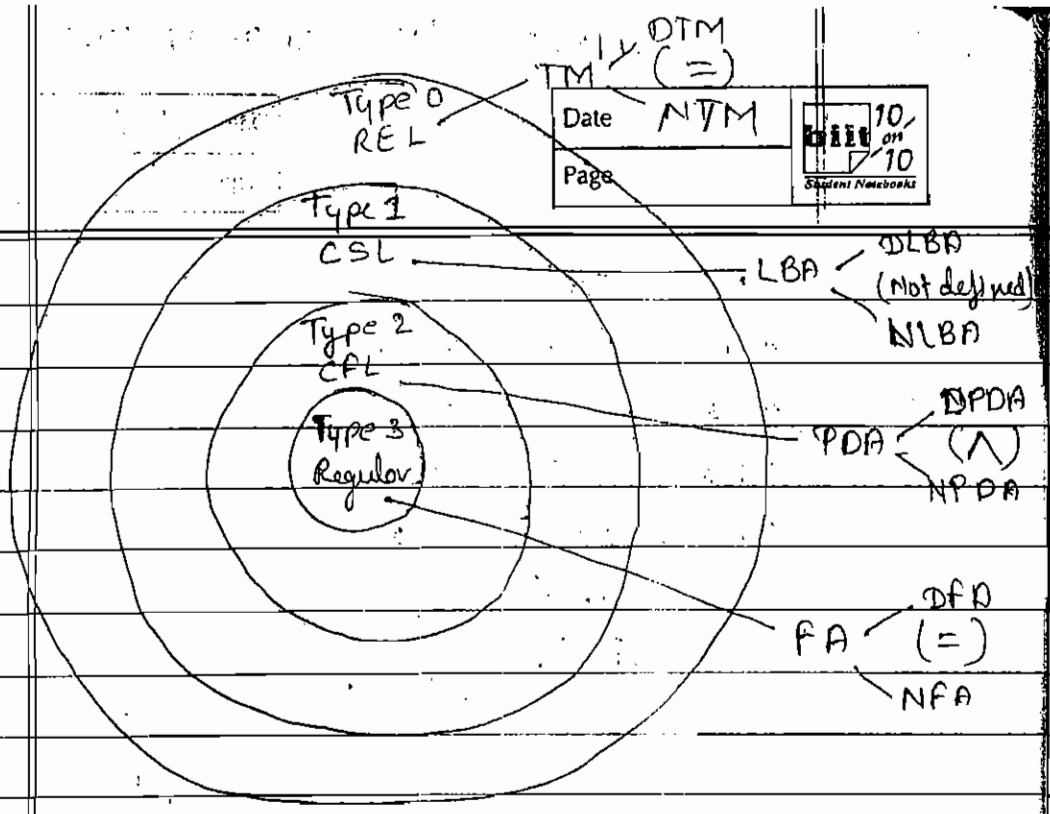
Collection of strings where these strings are formed based on some conditions.

Type 3 FL → Regular Lang. : FA.

Type 2R → Context Free Lang : PDA

Type 1R → Context Sensitive L. : Linear Bounded A.

Type 0 FL → Recursive Enumerable : T.M.



Expressive Power:

No. of lang. accepted by a particular M/C.

$FA \overset{(1)}{<} PDA \overset{(2)}{<} LBA \overset{(3)}{<} TM \overset{(4)}{}$

M/C accepts the lang. in two ways – Deterministic & Non-D (more than 1 soln.).

E.P. of DLBA & NLBA is undecidable.

$\Rightarrow L_1 = \{a^n b^m \mid n, m \geq 1\}$

$L_2 = \{a^n b^n \mid n \geq 1\}$

$L_3 = \{a^n b^n c^n \mid n \geq 1\}$

$L_4 = \{a^m b^n c^b d^n \mid n \geq 1\}$

DLBA (Not defined)
NLBA

PDA — DPDA (∧)
NPDA

DFA (=)
NFA

- FA fails to recognize lang. L_2 bcoz L_2 requires memory element (comparison b/w symbols of lang.).
  Hence to recognize lang. L_2 the suitable M/C is PDA.

- PDA fails to recognize lang. which requires >1 memory element (>1 comparison).

- To recognize such kind of lang. suitable M/C is TM.

- TM capable of recognizing all F.L.

Alphabet (Σ): Any finite non-empty set of symbols.

String: Any finite sequence of symbols over the given alphabet.

Prefix of String: Any seq. of leading symbols over the given ~~string~~ alphabet is known as POS.
For string of length n, POS is (n+1).
TOC ⇒ ∈, T, TO, TOC

Suffix of String: Any seq. of trailing symbols over the given ~~string~~ alphabet.
TOC ⇒ ∈, C, OC, TOC

Date / /
Page

Substrings: Any consecutive seq. of symbols over the given string.

TOC ⇒ T, O, C, TO, OC, TOC, ϵ

No. of substrings = $\frac{n(n+1)}{2} + 1$

Language: Any set of strings over the given alphabet.

L = {ϵ} - finite lang.

Grammer: Set of rules used to describe string of a lang.

G = (V, T, P, S)

Finite Automata:

Mathematical model which contains finite no. of states and x-tions defined b/w them.

FA

- Compilers (Lang. Recognizers) — w/o o/p
  - DFA
  - NFA
  - ϵ NFA
- Digital ckt. (o/p generators) — with o/p
  - Mealy M/C
  - Moore M/C

Representation or Notation of F.A.

- x-tional diagrams
- Transitional diagrams, Table

Date / /
Page

DFA:

It is FA in which from each & every state on every I/P symbol exactly one Transition should exist.

DFA = $(Q, \Sigma, \delta, q_0, F)$

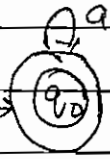
Q: Finite no. of states
Σ: I/P alphabet
δ: X-tion fun^n.
q_0: initial state
F: set of final states

$\delta : Q \times \Sigma \to Q$

Acceptance Mechanism:

By reading string x, from left to right if DFA enters into final state, given string is accepted, otherwise not accepted. If there is no final state in DFA, empty lang. will be accepted by DFA.

Ques: Minimal DFA accepts all strings of as & bs including ϵ.



Date / /

Page

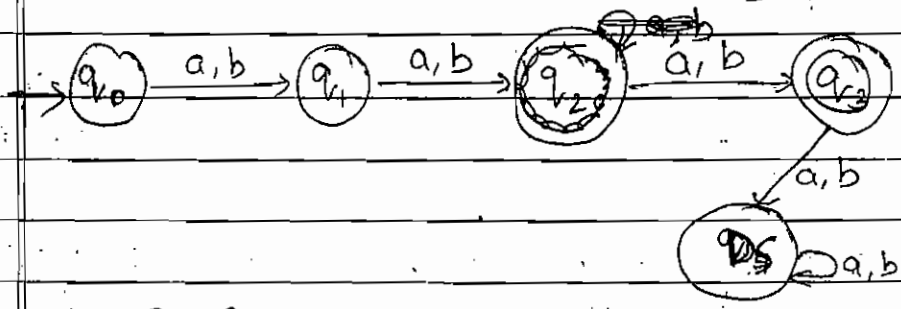
Date 04/24/11
Page

LENGTH OF STRING:

→ Construct minimal DFA that accepts all strings of As & Bs where length of each string is exactly 3.

$2 \times 2 \times 2 = 8$ possibilities. $\Sigma = (a, b)$

$L = \{aab, aba, abb, bab, aaa, bbb\}$

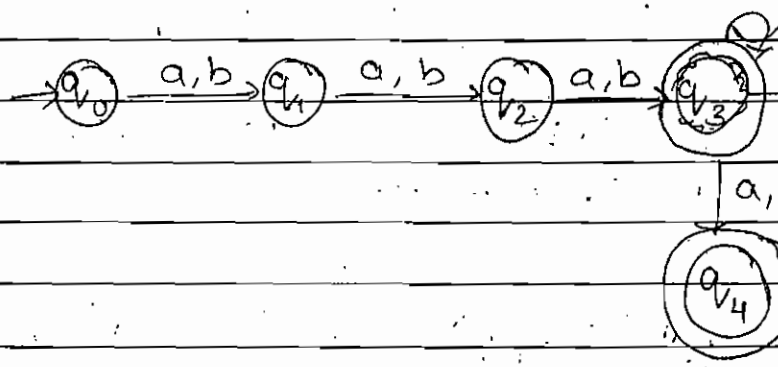


NO. OF STATES:

⇒ The minimal DFA that accepts all strings of As & bs where length of string is exactly n, requires $(n+2)$ no. of states.

→ $\Sigma = (a, b)$

$L = \{aaaa, bbbb, ---\}$. length of string is at least 4.

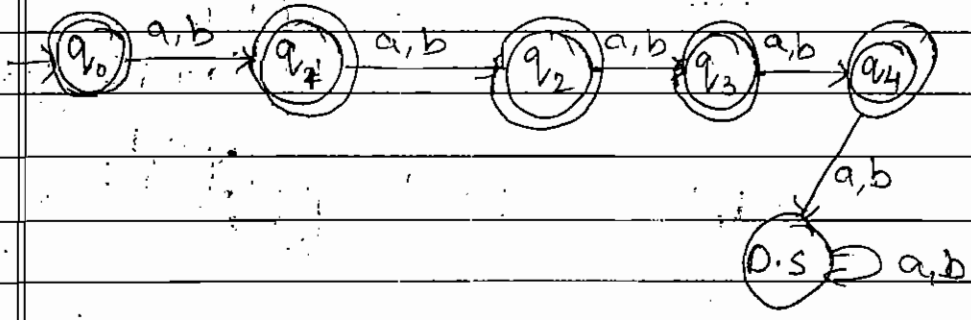


⇒ The minimal DFA that accepts all strings of as & bs where length of string is at least n, req. $(n+1)$ no. of states.

Date / /
Page

→ $\Sigma = (a, b)$ length of string is at most 4.

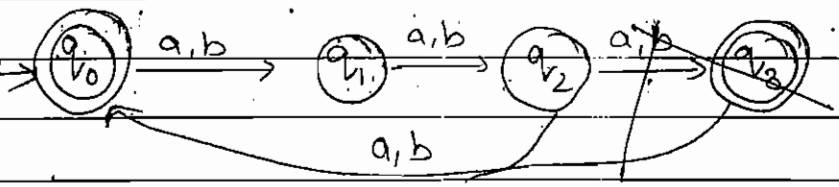
$L = \{0, 1, 2, 3, 4\}$



⇒ If length of string is at most n,
no. of states = n+2

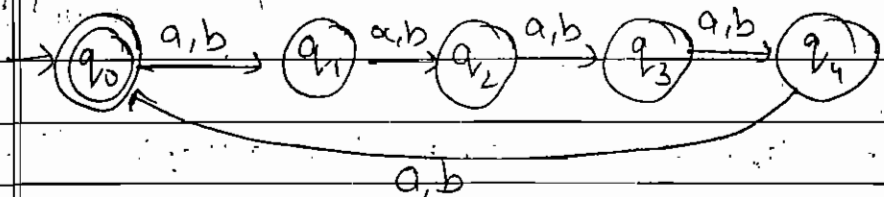
→ $\Sigma = (a, b)$ where LOS is divisible by 3.

$L = \{0, 3, 6, 9 \ldots\}$



→ $\Sigma = (a, b)$ LOS divisble by 5

$L = \{0, 5, 10, 15, \ldots\}$



⇒ If length of string is divisible by n, no. of states = n.

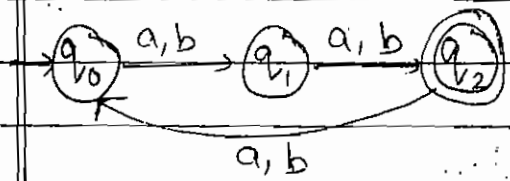
Date / /
Page

→ $\Sigma = (a, b)$ where LOS is ~~quadrant~~ Congruent to 2 mod 3. (equal)

$L = \{2, 5, 8, 11, 14 \ldots\}$
$\{3, 3, 3\}$

2 mod 4 means remainder is 2.

3 mod 5 mean remainder is 3.

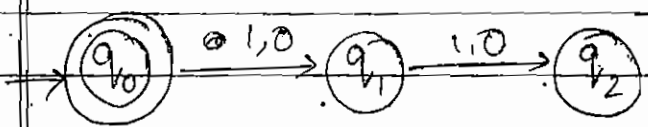


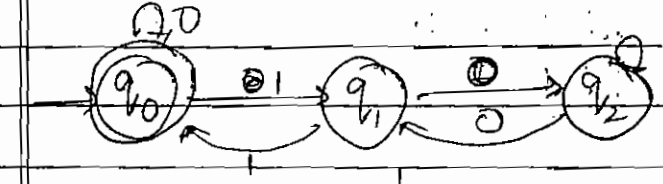
If LOS is congruent to M mod N, (equal)
no. of states = N.

NUMBER SYSTEM:

→ $\Sigma = (1, 0)$ accepts all binary nos, which are divisible by 3.

$L = \{0, 3, 6, 9 \ldots\}$
$\{0, 11, 110, 1001\}$





	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

Date / /
Page

→ For problems related to no. system, method is -

① Find no. of states
which is equal to no. of possible remainder. i.e. remainders for any no. divided by 3 are 0, 1, 2.

② Find final state
Initial state because, remainder is 0. 1st

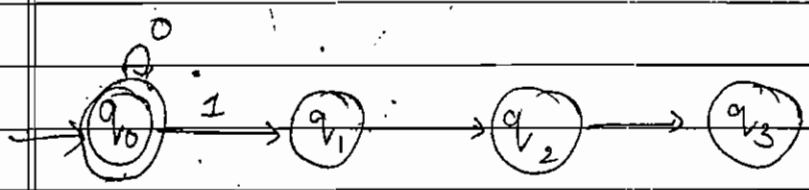
③ Transitions
- Better solve it through transition table.

→ $\Sigma = (0,1,2)$ that accepts all base 3 nos. which are divisible by 4.

$L = \{$ $\}$

No. of states = 4 0, 1, 2, 3

Final state =



	0	1	2
q_0	q_0	q_1	q_2
q_1	q_3	q_0	q_1
q_2	q_2	q_3	q_0
q_3	q_1	q_2	q_3

→ $\Sigma = (0, 1, 2, \ldots 9)$: accepts all decimal nos which are divisible by 6.

no. of states = 6

final = q_0

	0	1	2	3	4	5	6	7	8	9
q_0	q_0	q_1	q_2	q_3	q_4	q_5	q_0	q_1	q_2	q_3
q_1	q_4	q_5	q_0	q_1	q_2	q_3				
q_2										
q_3										
q_4										
q_5										

Date / /
Page

⇒ The DFA that accepts all base m nos: which are divisible by n requires n no. of states.

→ Construct DFA that accepts all binary nos. which are divisible by

(i) either 2 or 3.
(ii) by 2 but not by 3.
(iii) 3 but not by 2.
(iv) not by 2 or 3.

(i) no. of states = LCM(2, 3) = 6
{0, 1, 2, 3, 4, 5}

final states = {nos. divisble by 2} ∪ {nos. divisible by 3}
= {0, 2, 3, 4} = q_0, q_2, q_3, q_4

Transitions same as earlier.

(ii) no. of states = 6

final state = {nos. divisble by 2 by not by 3}
= (2, 4) q_2, q_4

(iii) no. of states = 6
final state = q_3

(iv) no. of states = 6
final states = {1, 5} = q_1, q_5

Date / /

Page

→ Construct DFA that accepts all base 4 nos. which are divisible by either 2 or 3 but not by 5. 0, 1, 2, 3

no. of states = LCM (2, 3, 5)

= 30

{0, 1, 2 - - - 29}

final states = {0, 2, 4, 6 - -} ∪ {0, 3, 6, 9} − {0, 10, 15, 20, 25}

= {2, 3, 4, 6, - - }

→ How many nos. of states in DFA that accepts all hexadecimal nos. which are divisible by 3 by not by 5 and 7.

no. of states = LCM (3, 5, 7)

= 105

Date / /
Page

→ How many no. of DFA's are possible with 2 states x & y for input alphabet (0,1).

2 × 16

(x) (y)

	0	1	
2 x	2	2	possibilities
2 y	2	2	

4 × 16 = 64 + 64 = 128

possibilities on symbols ⇒ 0 (x) & (x) —0→ (y)
(x) & (x) —1→ (y)

possibilities on states ⇒ (x) or x

→ How many no. of FAs are possible with 3 states X, Y, Z over the i/p alphabet (a, b, c), where X is always initial state.

		a	b	c
2	X	3	3	3
2	Y	3	3	3
2	Z	3	3	3

3^9

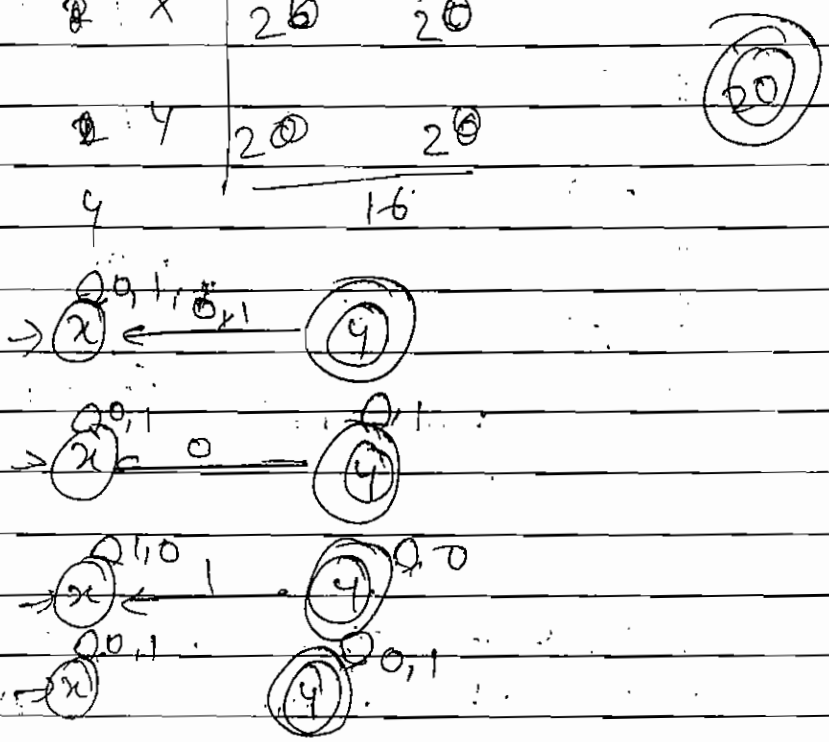
Date / /
Page

⇒ NPDA is default PDA.

→ How many no. of DFA are possible with 2 states X & Y, over the I/P alphabet (a,b) which accepts empty lang. only, where X is always initial state.

	a	b
→ X	2	2
Y	2	2

4 16



⇒ How many DFA are possible with 2 states x & y over I/P alphabet (0,1) which accepts complete lang., where x is always initial state.

	0	1
→ X	2	2
Y	2	2

4 + 4 = 20

Date / /
Page

→ How many nos. of FA are possible with N states $(Q_1 \text{ --- } Q_n)$ for the $\Sigma = \{1, 2, 3 \text{ --- } m\}$, where Q_1 is always initial.

	1	2	3	—	m
Q_1	n	n	n	—	n
Q_2					
Q_n					

$$n^{n \times m}$$

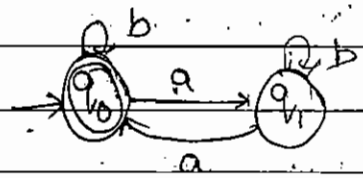
$$2^n \times n^{n \times m}$$

no. of FA's = $2^n \times n^{n \times m}$

→ (i) Construct Min^n. DFA that accepts $\Sigma = (a, b)$ where no. of As are even in every string.

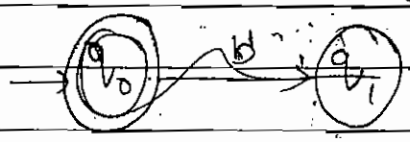
$\Sigma = (a, b)$

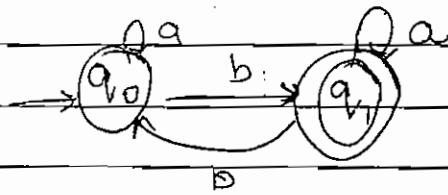
$L = \{ac, aaaab, \text{ --- }\}$



(ii) no. of b's odd

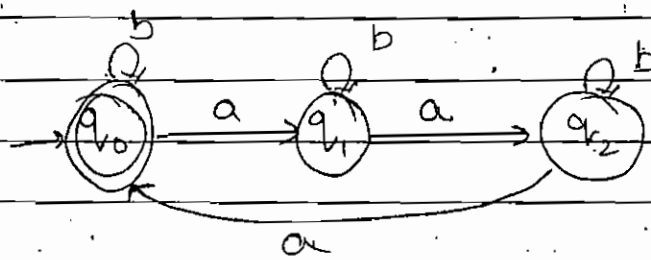
$L = \{1, 3, \ldots\}$





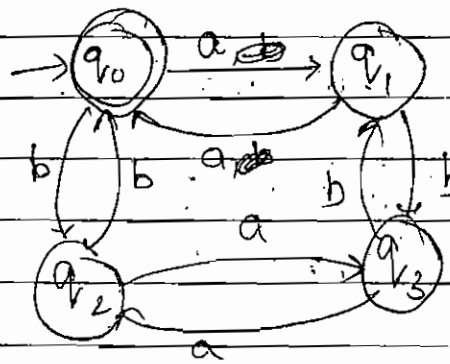
→ $\Sigma = (a, b)$ no. of as are divisible by 3 in each string.

$L = \{0, 3, 6 - -\}$



→ $\Sigma = (a, b)$ where no. of a's & no. of bs are even.

$L = \{0, 2, 4, 6 - -\}$



Date / /
Page

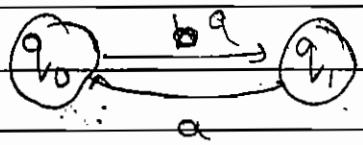
→ (i) a is even b is odd.

$L = \{1, 3, 5, \ldots\}$

(ii) a is odd, b is even.

$\{1, 3, 5 \ldots$

(iii) a is odd, b is odd.

(i) 

q_2 is final

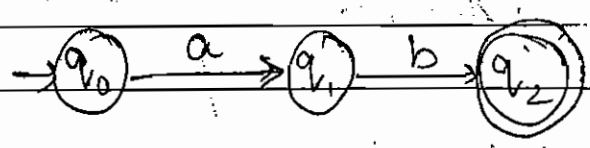
(ii) q_1 is final

(iii) q_3 is final

→ Cons:- DFA for lang, $L = \{a^n b^n \mid n \geq 1\}$.

$\Sigma = (a, b)$

$L = \{ab, aabb, aaabbb, \ldots\}$

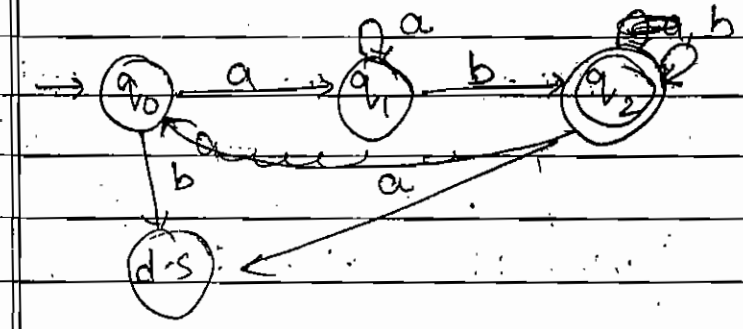


FA is not possible.

→ $L = \{a^n b^m \mid n, m \geq 1\}$

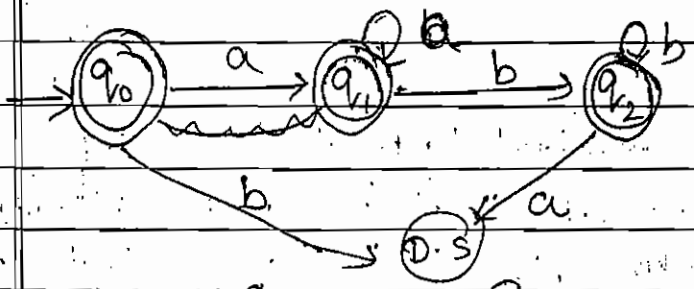
$L = \{ab, aab, \ldots\}$

Date / /
Page



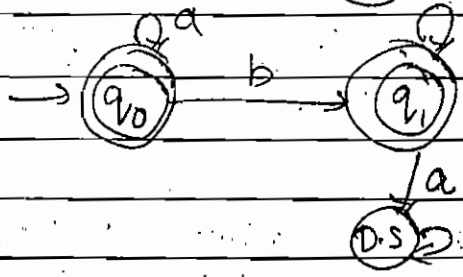
→ $L = \{ a^n b^m \mid n, m \geq 0 \}$

$L = \{ 0, a, b, ab, \text{---} \}$



$\geq 1 \}$.

$b = \text{---} \}$?



$k > 2$

n -

aa

→ Min^m no. of states needed in DFA to recognize the Lang. $L = \{ a^{nk} \mid k \geq 0 \text{ \& } n \text{ is +ve int.} \}$

A) $n+1$ B) $k+1$ C) 2^{k+1} D) 2^{n+1}

Soln $L = \{ a, aa, a \text{---} \}$.

check for $k=1$

$k=2$

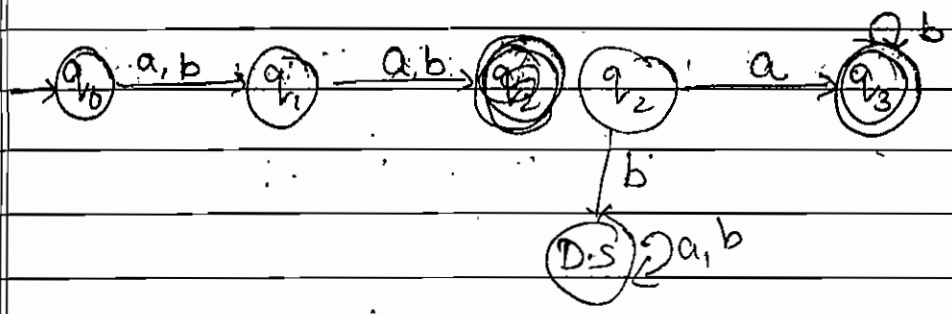
$k=3$

Date / /

Page

→ $\Sigma = (a, b)$ where 3rd input symbol is a while reading the string from LHS.

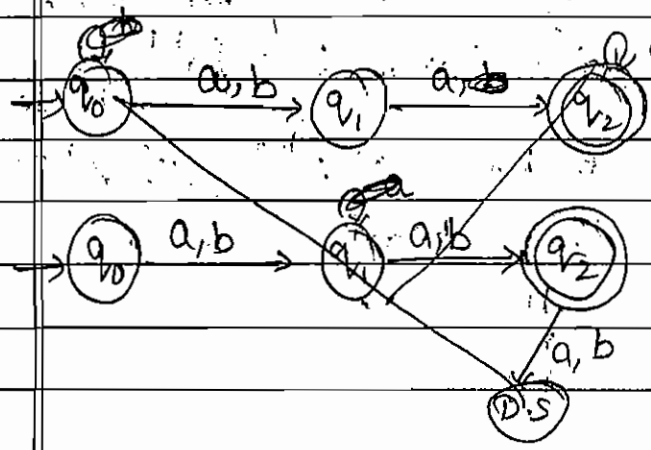
$L = \{aaa, bba, aba, baa, \ldots\}$



⇒ The minimal DFA that accepts all string of a & b where nth i/p symbol is a while reading string from LHS, requires (n+2) states.

→ $\Sigma = (a, b)$ where 2nd I/P symbol is 'a' while reading string from R.H.S.

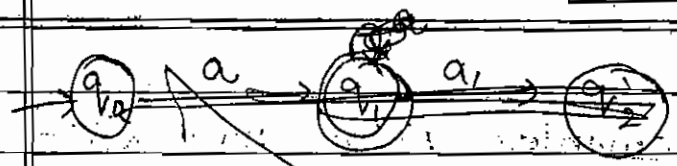
$L = \{aa, ab, aab, \ldots\}$



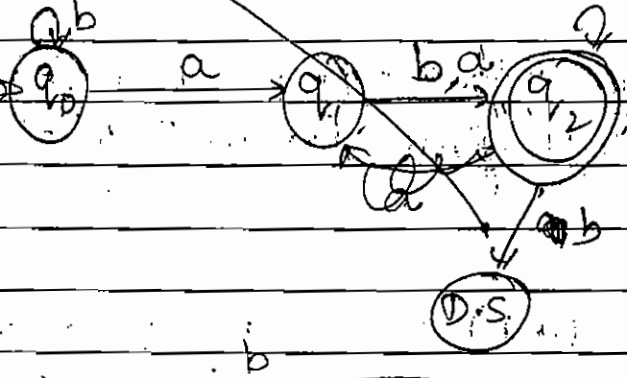
aaab

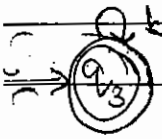
bbaa

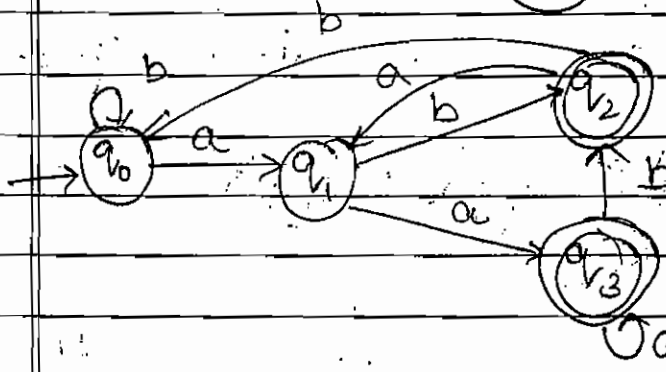
Date / /
Page



symbol is
Carry the
LHS.







all
i/p
string
states.

⇒ The minimal DFA that accepts all string of As & Bs where n^{th} I/P symbol is a while reading the string from RHS requires (2^n) states.

symbol
ding
RHS.

⇒ The complexity of constructing DFA is very high for some languages.

⇒ whenever DFA designing is difficult, for such kind of lang we have to construct NFA.

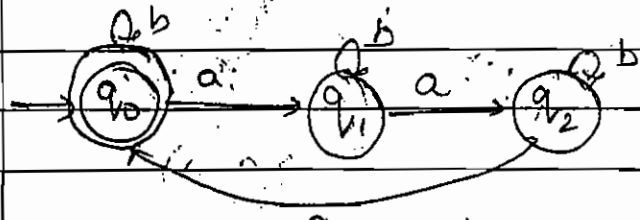
Date / /
Page

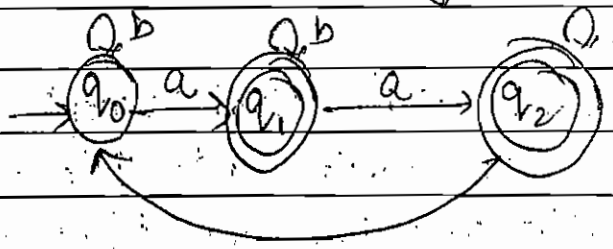
Complemented Finite Automata:-

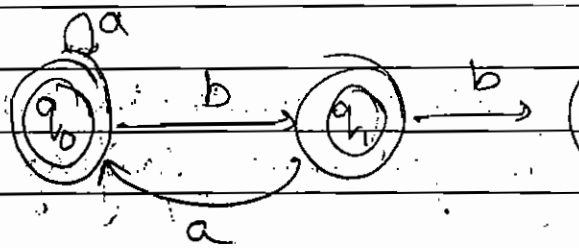
By interchanging final & non-final states of given FA we'll get complemented FA.

→ $\Sigma = (a, b)$ no. of a's are not divisible by 3.

0, 3, 6



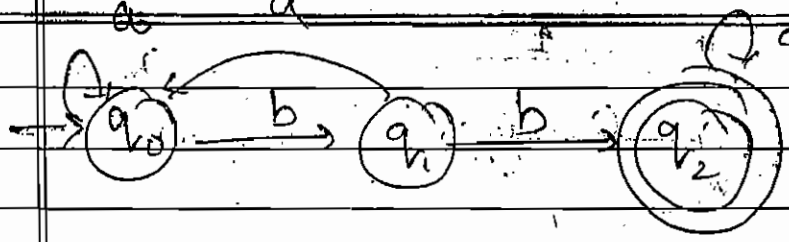




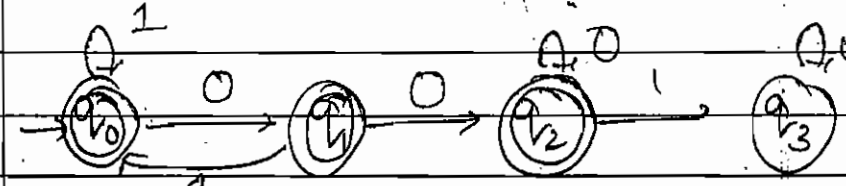
If u complement above DFA, then the lang. accepted by that DFA is:-

A) starting with a or b
B) ending with a or b
C) contains atleast 2 b's.
D) NOT

Date / /
Page



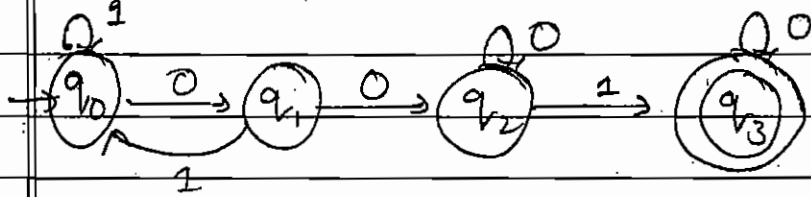
final implemented



not

3- ~~If~~ Complement above FA. Let S denotes a 7bit binary string in which 1st, 1st, 4th & last bits are 1, then how many no. of strings are there in lang. ; accept by above complemented DFA from S.

A) 16 b) 8 , c) 7 D) 9



1 _ _ 1 _ _ 1

2 x 2 2 x 2

Total string = 2 x 2 x 2 x 2 = 16

so FA, that

| 0 0 | 0 0) or | 0 0 | 0 0)

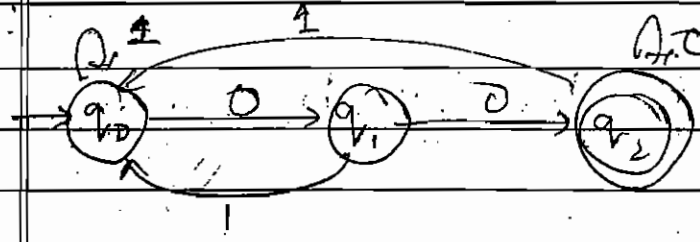
0 | 0 |

1 0 1 0

1 1 1 1

4 + 4 − 1 (in case of set repetition is not allowed)

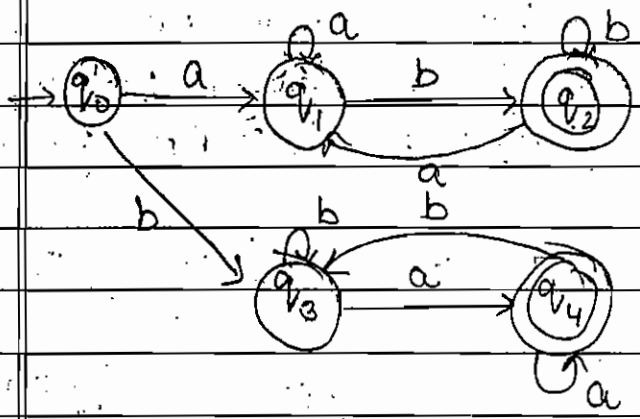
= 7



(c)

→ $\Sigma = (a, b)$ where 1st & last symbols are diff.

$L = \{ab, ba, abb, baa, \ldots\ abbab \ldots\}$



c, d

Date / /
Page

Nondeterministic FA:

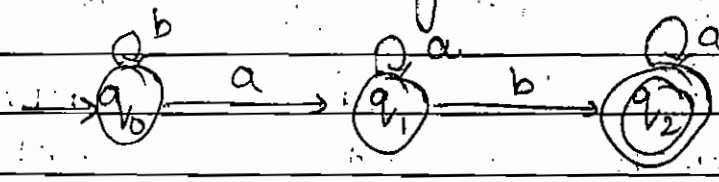
In NFA, from each and every state on every I/P symbol, ~~exactly one transition exists~~ there may be zero or more than zero transitions exist.

* All DFAs are NFAs.

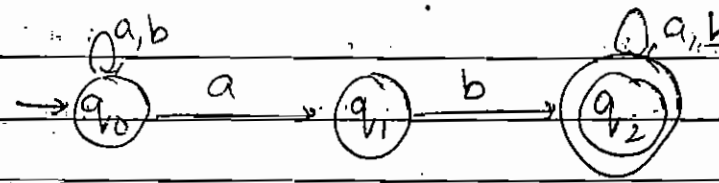
$NFA \equiv (Q, \Sigma, \delta, q_0, F)$

Q : finite no. of states
Σ : I/P alphabet
δ : transition func. $\delta : Q \times \Sigma = P(Q)$ or 2^Q
q_0 : initial state
F : set of final states

→ Construct NFA that accepts all string of as & bs where each string contain ab as substring.



: DFA



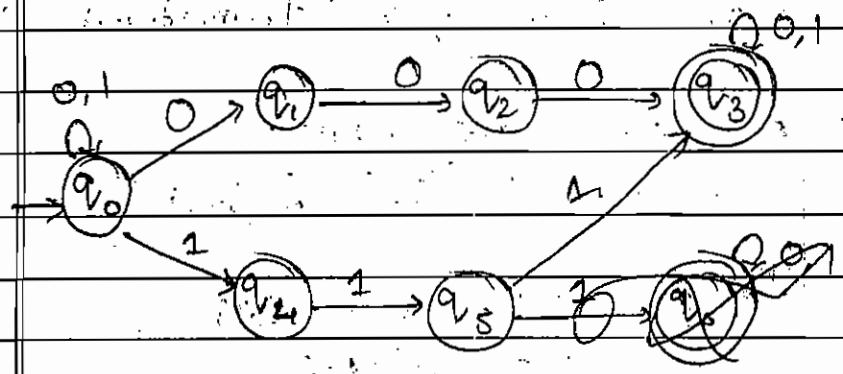
→ $\Sigma = (0,1)$

Date / /
Page

tribit = 000 or 111

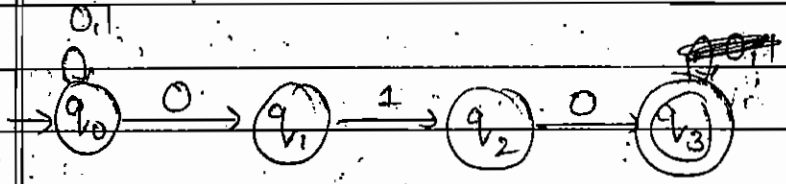
→ $\Sigma = (0,1)$ where each string contains tribit as substring.

$L = \{000, 111, 10001, \ldots\}$



→ $\Sigma = (0,1)$ where each string ends with 010.

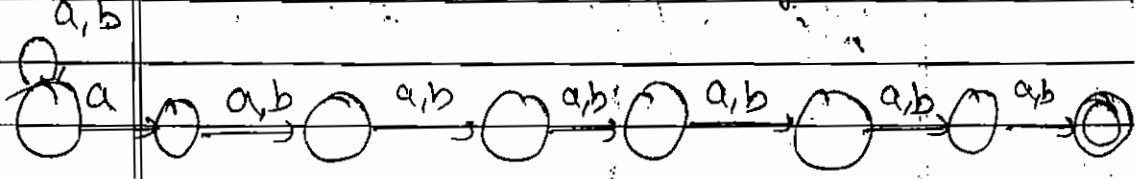
$L = \{010, 1010, 0010, \ldots\}$



	$w \in L$	DFA	NFA
→ (1)	$\|w\| = n$	$n+2$	$n+1$
(2)	$\|w\| \geq n$	$n+1$	$n+1$
(3)	$\|w\| \leq n$	$n+2$	$n+1$
(4)	$\|w\| \bmod n = 0$	n	n
(5)	$\|w\| = m \bmod n$	n	n
(6)	n^{th} I/P from L.H.S.	$n+2$	$n+1$
(7)	n^{th} I/P from RHL	2^n	$n+1$

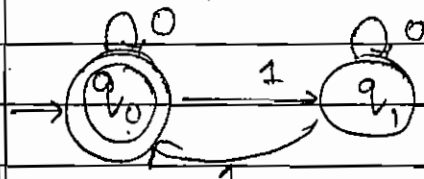
st ring as

→ $\Sigma = (a, b)$ where n^{th} i/p symbol is a while reading string from RHS.



⇒ for n^{th} i/p symbol from R.H.S, no of states = n+1

string ends → Construct an equivalent DFA for following NFA.



NFA:

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

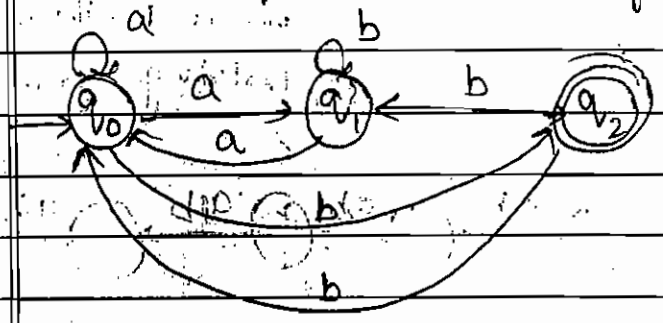
DFA:

	0	1
q_0	q_0	q_1
q_1	q_1	$[q_0 q_1]$
$[q_0 q_1]$	$[q_0 q_1]$	$[q_0 q_1]$

Date / /

Page

→ Construct equivalent DFA for NFA.



NFA:

	a	b
q_0	$\{q_0, q_1\}$	q_2
q_1	q_0	q_1
q_2	∅	$\{q_0, q_1\}$

DFA:

	a	b
q_0	$[q_0 q_1]$	q_2
q_1	q_0	q_1
q_2	∅	$[q_0 q_1]$
$[q_0 q_1]$	$[q_0 q_1]$	$\{q_1, q_2\}$
$[q_1 q_2]$	∅	$[q_0 q_1]$

Date / /
Page

DFA:

	a	b
q_0	$[q_0 q_1]$	q_2
$[q_0 q_1]$	$[q_0 q_1]$	$[q_1 q_2]$
q_2 ~~$[q_2 q_2]$~~	DS	$[q_0 q_1]$
$[q_1 q_2]$	q_0	$[q_0 q_1]$
DS	DS	DS

→ NFA

	0	1
q_0	q_1	$\emptyset$
q_1	$\{q_0, q_1\}$	$\emptyset$

DFA

	0	1
q_0	q_1	DS
~~DS~~	~~DS~~	~~DS~~
q_1	$[q_0 q_1]$	DS
$[q_0 q_1]$	$[q_0 q_1]$	DS
DS	DS	DS

→ NFA

	0	1
q_0	q_0	q_0
q_1	q_1	q_1
q_2	q_2	q_2

DFA

	0	1
q_0	q_0	q_0

Q ⇒ While converting from NFA to DFA, no. of states possible in DFA are is 1 in the best case and 2^n in the worst case.

→

	0	1
p	{p, q}	p
q	{r, s}	t
r	{p, r}	t
*s	∅	∅
*t	∅	∅

DFA:

X	0	1
p	[pq]	p
q	[rs]	t
[pq]	[pqrs]	[pt]
t	DS	DS
[pt]	DS	DS
~~[pqrs]~~		

	0	1
p	[pq]	p
[pq]	[pqrs]	[pt]
[pqrs]	[pqrs]	[pqrs]
[pt]	DS	DS

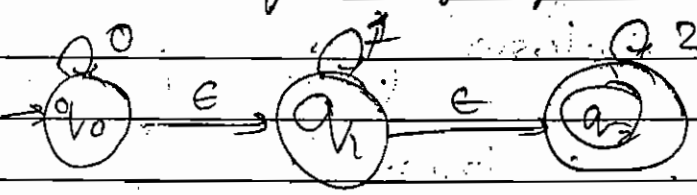
NFA,
re is
ein

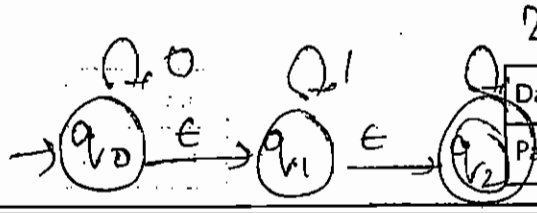
ε-NFA:-

$(Q, \Sigma, \delta, q_0, F)$

$\delta: Q \times \Sigma \cup \{\epsilon\} \to P(Q)$ or 2^Q

→ Construct eq. NFA for following ENFA.

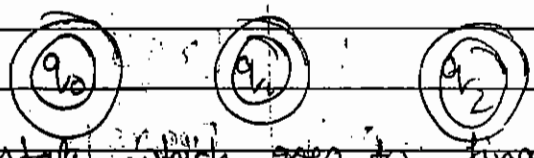




Date / /

Page

NFA:



$\Rightarrow$ All states which goes to final states using ϵ, will be final state in NFA.

ϵ-closure $(q_0) = \{q_0, q_1, q_2\}$

i.e. all the possible states reachable through ϵ from q_0

ϵ-closure $(q_1) = \{q_1, q_2\}$

ϵ-closure $(q_2) = \{q_2\}$

$$\delta(q_0, a) = \epsilon\text{-closure}\left(\delta(\epsilon\text{-closure}(q), a)\right)$$

$$\delta(q_0, 0) = \epsilon\text{-closure}\left(\delta(\epsilon\text{-closure}(q_0), 0)\right)$$

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

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

$$= \epsilon\text{-closure}\left(q_0 \cup \phi \cup \phi\right)$$

$$= \epsilon\text{-closure}(q_0)$$

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

$\delta(q_0, 1) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0), 1))$

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

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

$= \{q_1, q_2\}$

$\delta(q_0, 2) = \{q_2\}$

$\delta(q_1, 0) = \emptyset$

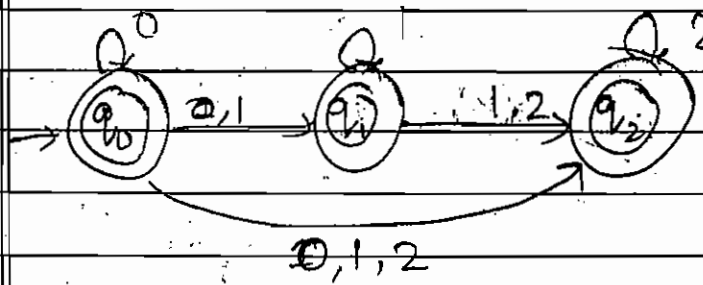
$\delta(q_1, 1) = \{q_1, q_2\}$

$\delta(q_1, 2) = q_2$

$\delta(q_2, 0) = \emptyset$

$\delta(q_2, 1) = \emptyset$

$\delta(q_2, 2) = \{q_2\}$



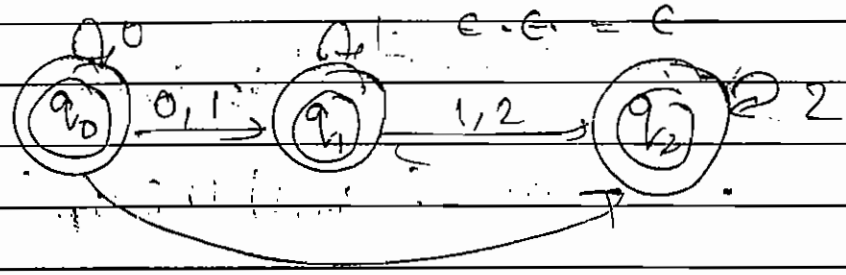
Date / /
Page

2nd Approach:

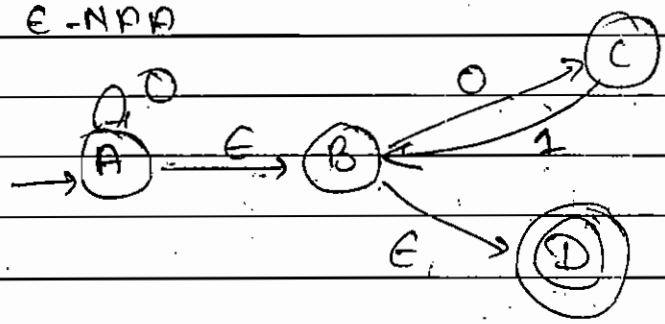
$0 \cdot \epsilon = 0$

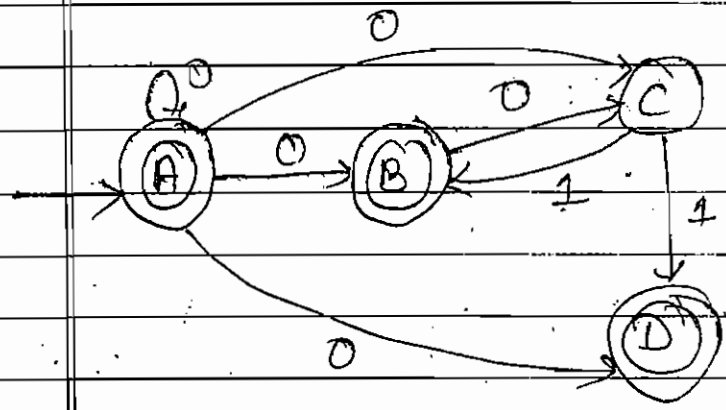
$\epsilon \cdot 0 \cdot \epsilon = 0$

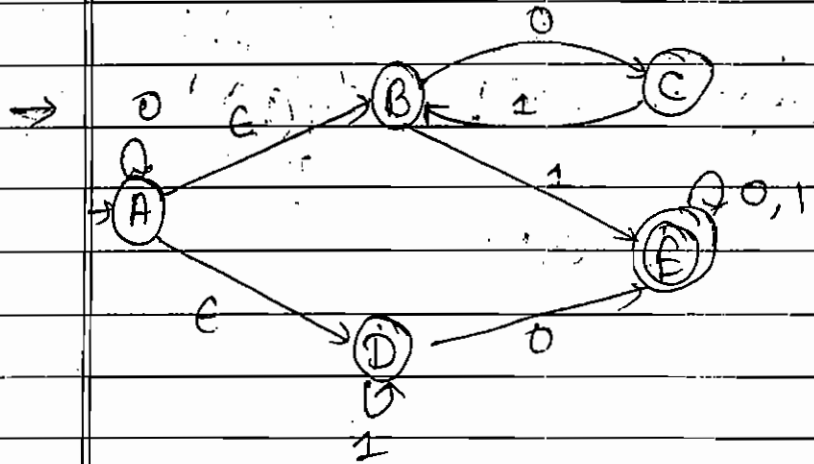
$\epsilon \cdot \epsilon = \epsilon$

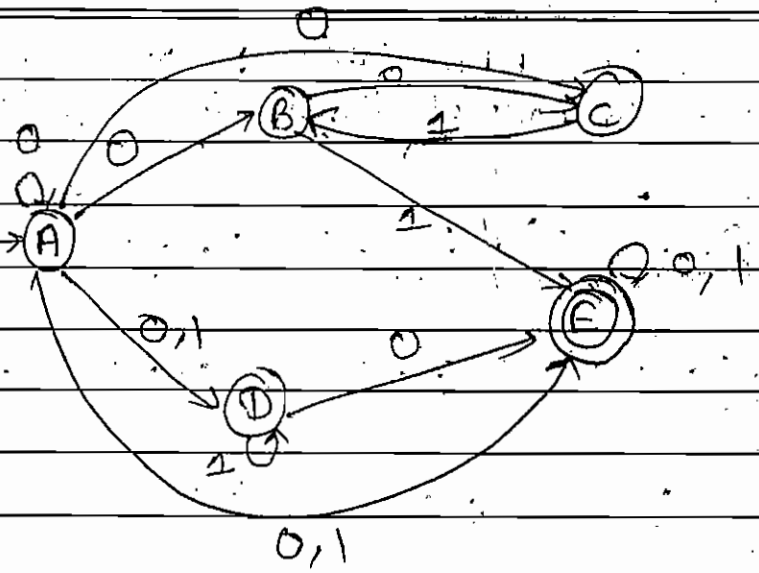


→ ϵ-NFA



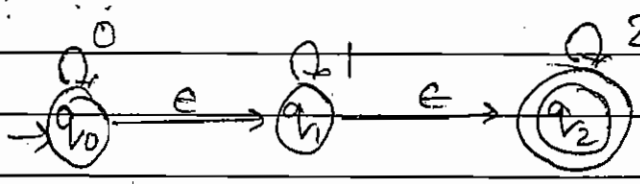






Conversion from ENFA to DFA:

→ Construct an equivalent DFA for following ϵ-NFA:



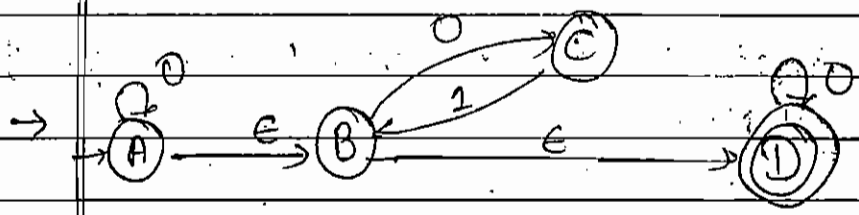
① Create NFA Table

	0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	$\emptyset$	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$	$\{q_2\}$

Date / /
Page

② Create DFA table : Initial state is ϵ-closure of initial state of ϵNFA

	0	1
$[q_0 q_1 q_2]$	$[q_0 q_1 q_2]$	$[q_1 q_2]$	$[q_2]$
$[q_1 q_2]$	DS	$[q_1 q_2]$	q_2
q_2	DS	DS	q_2
DS	DS	DS	DS



NFA:

	0	1
A	{A, B, C, D}	ϕ
B	{C, D}	ϕ
C	ϕ	{B, D}
D	D	ϕ

Date / /

Page

A_2 is

DFA:

	0	1
[A B C D]	[A B C D]	[B D]
[B D]	[C D]	DS
[C D]	D	[B D]
[D]	D	DS
DS	DS	DS

	0	1
[A B D]	[A B C D]	DS
[A B C D]	[A B C D]	[B D]
[B D]	[C D]	DS
[C D]	D	[B D]
[D]	D	DS
DS	DS	DS

Date / /
Page

Decision Properties of FA:-

Decidable Problem: if there exist some generalised algorithm in order to solve the problem.

(1) Membership Problem

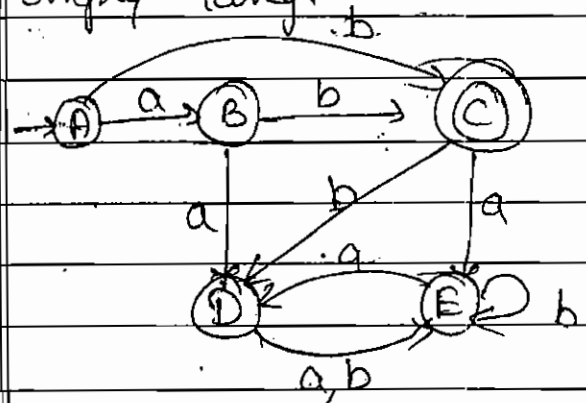
(2) Emptyness Problem Algo:

↳ Eliminate all inaccessible states.

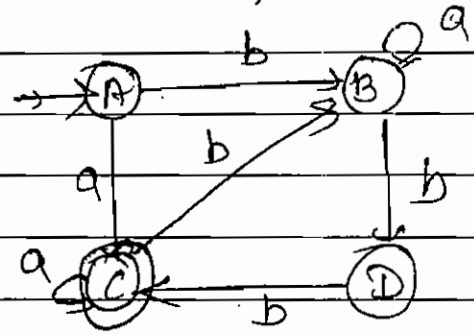
↳ In remaining finite automatea, if we find atleast one final state then the lang. accepted by that automata is non-empty otherwise empty lang.

→ Which of following Automat accepting empty lang.

(1)

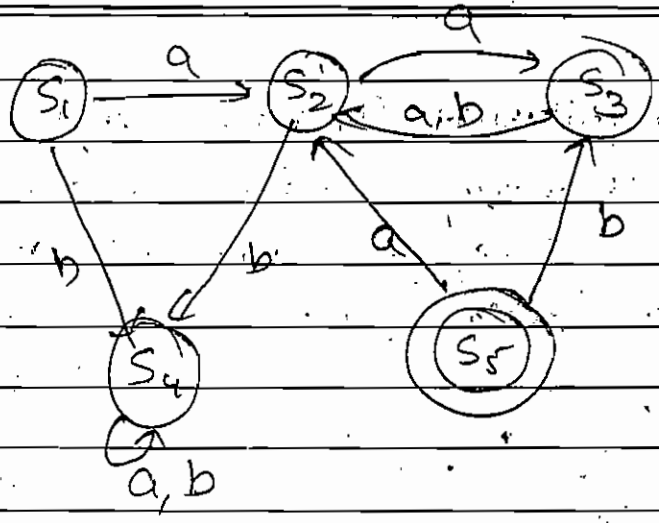


(2)



Date / /
Page

(3)

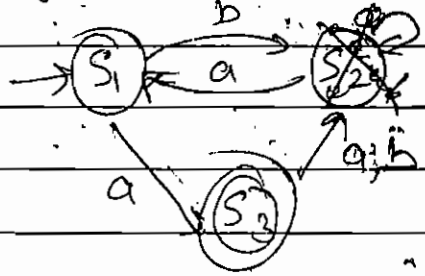


(3) Finiteness Problem Algo:

↳ Eliminate all inaccessible states.

↳ Select some states from which we can't reach final state and delete those states.

↳ In remaining finite automata if there exist any loop or cycle or self loop then the lang. accepted by finite automata is infinite.

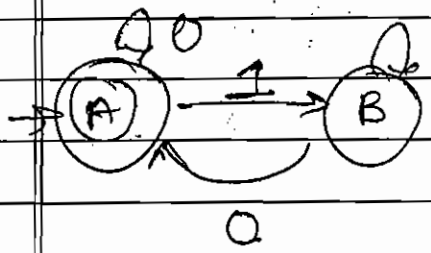


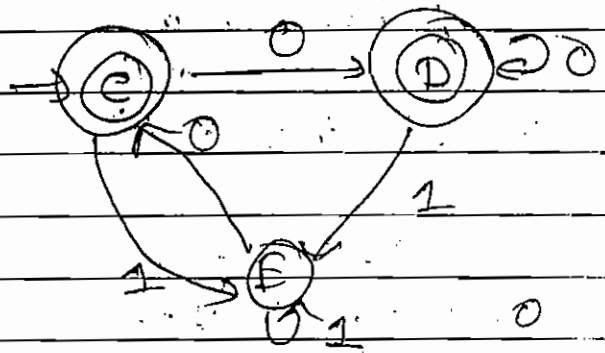
Date / /
Page

(4) Equivalance Problem Algo.

⇒ Checking whether lang. accepted by giving 2 automata are equal or non-equal.

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





Comparison Table

	0	1
→ (A, C)	(A, D)	(B, E)
(A, D)	(A, D)	(B, E)
(B, E)	(A, C)	(B, E)
~~(B, C)~~	ϕ	

Date 04/30/2011
Page

Regular Expressions:

The declarative way of representing a Reg. Lang. (simple & mathematical).

If we can construct DFA from a lang. then it is Reg. Lang.

Operators —

+ ⇒ Union Op.

. ⇒ Concetanation

* ⇒ Kleene closure.

We can construct R.E. for R.L. by taking symbols in I/P alphabet as operands and by using above 3 operators.

	R.L.	R.E.
1)	$L = \{\}$	ϕ
2)	$\{\epsilon\}$	ϵ
3)	$\{a, b\}$	$(a+b)$
4)	$\{a\}$	a
5)	$\{\epsilon, a, aa, \ldots\}$	a^*
6)	$\{a, aa, aaa, \ldots\}$	a^+ → +ve closure (excluded ϵ)
7)	$\{a, b, ab, ba, aa, \ldots\}$	$(a+b)^+$

Date / /
Page

4 Find R.E. that generates all strings of 0s & 1s where each string starts with 0.

$0(0+1)^*$ $\{0, 00, 01, --\}$

→ starting with 0 ending with 1

$0(0+1)^*1$

→ where each string start & ends with diff symbol.

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

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

→ string start & ends with same symbol. $L = \{0, 1, 00, 11, --\}$

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

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

→ when LOS is exactly 4.

$L = \{0000, 0111, 1111 --\}$

$(0+1)(0+1)(0+1)(0+1)$

(a, b, c)

→ How many no. of states in DFA that accepts lang. accepted by following R.E.

strings starts

$(a+b+c)(a+b+c) \cdots = (n-2)$ times.

length $= (n-2) = LOS$

exactly $(n-2)$.

with 1

no. of state $= N+2$

$= (n+2)-2 = n$

ends

→ ten' LOS. is atleast 3.

$L = \{$ ~~a, 00, 000, 000, 0000,~~

no. of sta $= n+1$

same

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

or

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

→ How many no. of states in DFA that accepts Lang. generated by foll. R.E.

$(a+b)(a+b) \cdots (n-2 \text{ times})(a+b)^+$

$LOS \geq n-1$

$\therefore$ no. of state $= N+1 = (n-1)+1$

$= n$

Date / /
Page

→ LOS ≤ 5

L = {ε, 0, 00, 000, ... 00000}

~~(0+1)(0+1).(0+1)..~~

~~ε+~~ (0+1) + (0+1)(0+1).

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

→ LOS is divisible by 4.

L = {0, 4, 8, 12, --}

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

→ LOS is odd.

L = {1, 3, 5, --}

$[(0+1)(0+1)]^* (0+1)$.

→ Each string contains even no. of 0s followed by odd no. of 1s.

L = {1, 001, 111, 00001, --}

$(00)^*(11)^*1$

→ where no. of 0s are divisible by 3.

$L = \{0, 000, 000000, \ldots\}$

$(000)^* 1^*$

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

→ Final R.E for R.L.

$L = \{0^n 1^n \mid n \geq 1\}$

Not possible.

→ $L = \{0^n 1^m \mid n, m \geq 1\}$

$L = \{01, 001, 011, \ldots\}$

$0^+ 1^+$

→ $L = \{0^n 1^m \mid n, m \geq 0 \text{ and } n+m \text{ is even}\}$

$L = \{0, 01, 0001, 0111, \ldots\}$

$0^+ 1^+$

~~$(0^* 1^*)(0^* 1^*)$~~

$\times [(0^+ 1^+)(0^+ 1^+)]^*$

$(00)^* (11)^* + (00)^* 0\ (11)^*$

Date / /
Page

→ $L = \{0^n 1^m \mid n+m \text{ is odd}\}$

even + odd
odd + even } odd.

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

→ Whr No. of 0s are exactly 4 in each string

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

→ where no. of 0s are atleast 2 in each string.

$1^* 0 1^* 0 1^* 0^* 1^*$

or

$1^* 0 1^* 0^+ 1^*$

or

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

→ No. of 0s are atmost 3.

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

→ How many no. of states req. in DFA that accepts lang. gener by foll. R.E.

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

4th element from R.H.S is 1

so no. of state $= 2^4 = 16$

each

Identities of R.E. :-

(1) $\phi + R = R + \phi = R$

(2) $\phi \cdot R = R \cdot \phi = \phi$

(3) $\epsilon \cdot R = R \cdot \epsilon = R$

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

(5) $\phi^* = \epsilon$

(6) $\epsilon + RR^* = R^*R + \epsilon = R^*$

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

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

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

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

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

Date / /

Page

→ Construct the following R.E from P & Q

where

$P = ((01)^* + 1)^*$

$Q = (01)^* (1 + (01)^*)^*$

a) $P = Q$ b) $P \in Q$

c) $P > Q$ d) none.

$P = ((01)^* + 1)^*$

$Q = ((01) + 1)^*$ (1) c)

→ The R.E having minimal * heights is preferrable.

→ Find out which of these 2 are equal

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

(2) $0(00)^*$

(3) $(00)^*$

(4) 0^*

A) I II B) II, IV, C) I, III, D) I, IV

P & Q → Construct following R.E - P & Q -

$P = (11!^*)^*$ 0, 11, 111

$Q = (11 + 111)^*$

$L_1 = \{0, 2, 3, \dots\}$ $L_2 = \{0, 2, 3, 4, \dots\}$

which is true about P & Q

A) B) C)

D)

(1) a) $P = Q$ b) $P \subset Q$ c) $P \supset Q$ d) none.

(2) how many no of strings are there in minimal DFA that accepts P & Q.

A) 4, 3 B) 3, 3 C) 3, 4 d) none

equal

4

2006 → Let $L = (111 + 11111)^*$

How many no of states are there in minimal DFA that accepts L.

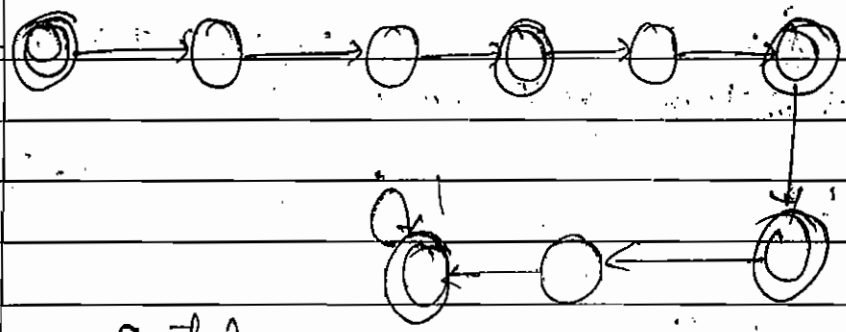
a) 15 b) 8 c) 9 d) 5.

I, IV $L = \{0, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15 \dots$

8 states

Date / /
Page

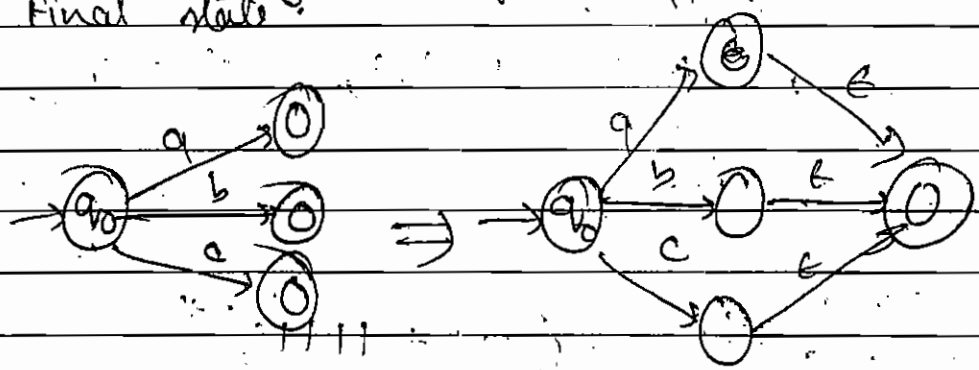
Sol^n ⇒ Untill it create DFA upto the length after which it is accepting all the lengths then for that state makes self loops.



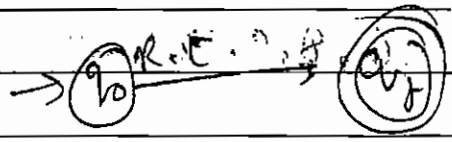
9 states.

Conversion from FA to RE :-

→ Convert any no. of states into single final state.



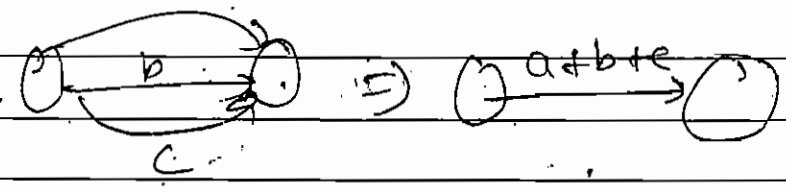
→ Add only one edge from initial to final

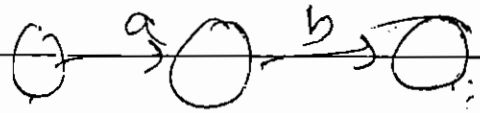


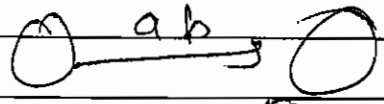
Date / /

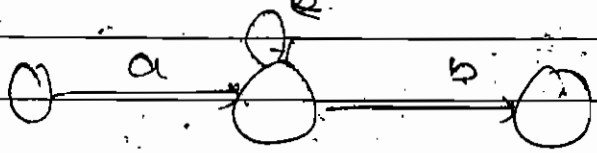
Page

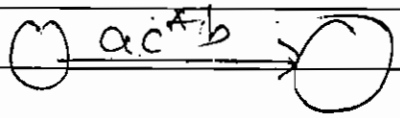
Conditions :-

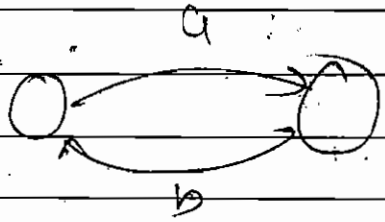
(1) 

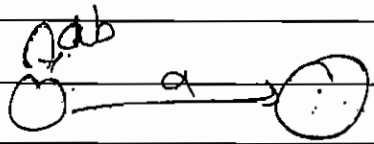
(2) 



(3) 



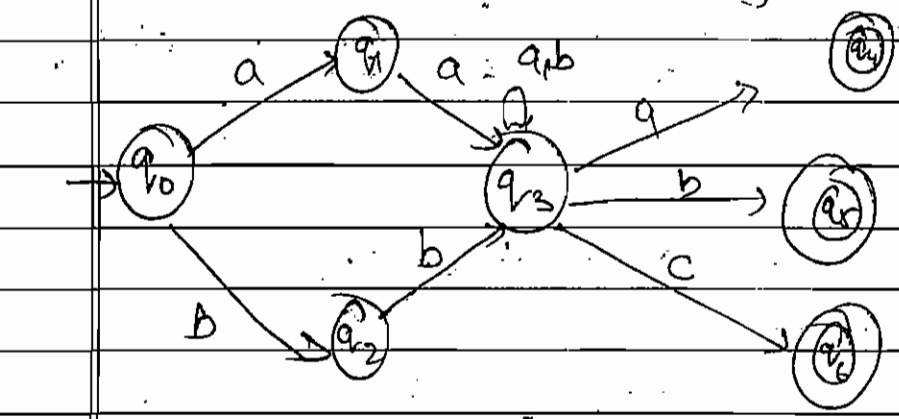
(4) 

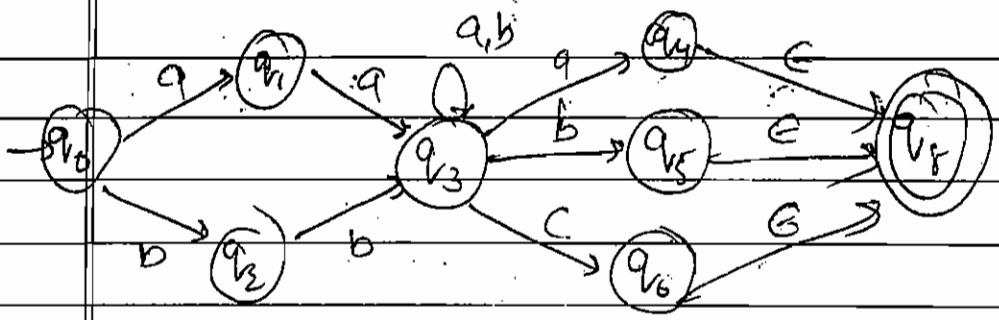


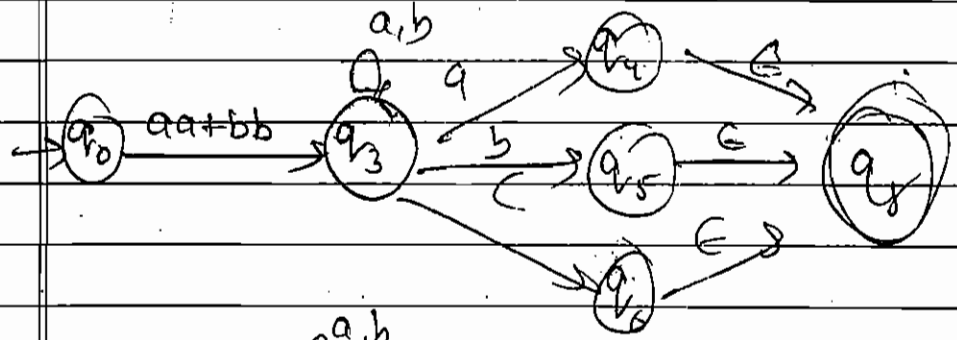
Date / /

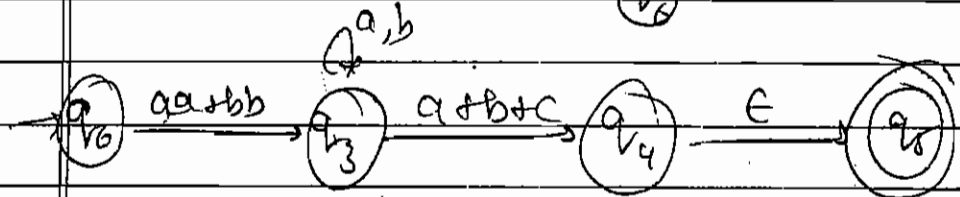
Page

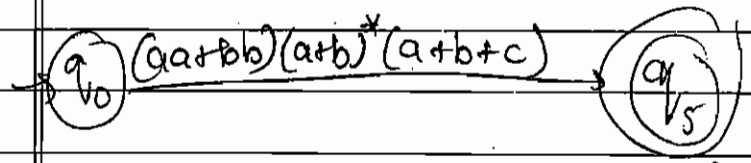
→ Construct RE for following NFA –



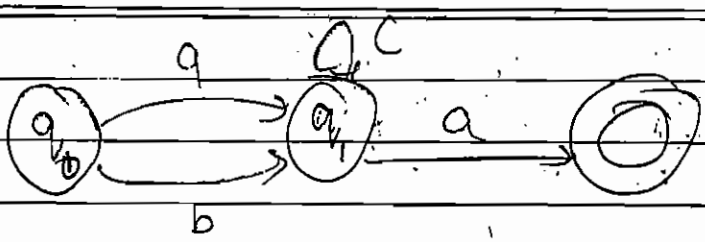


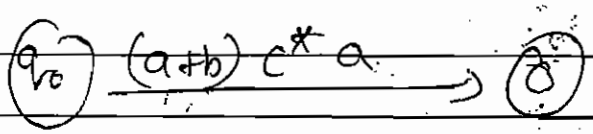


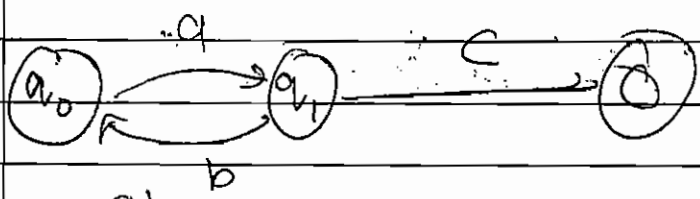


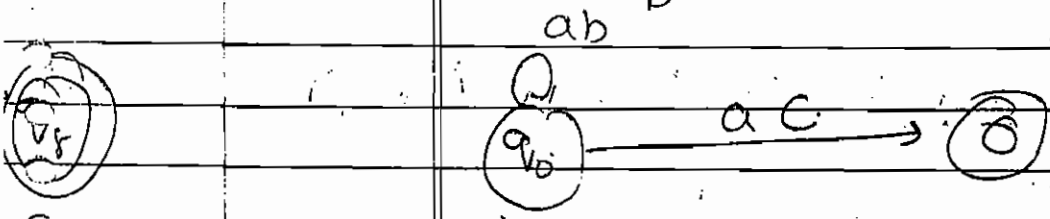


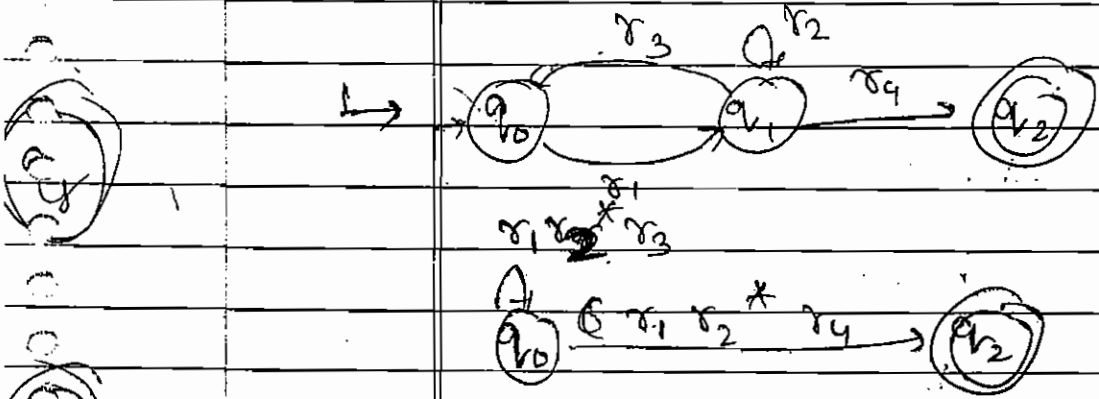
$(aa+bb)(a+b)^*(a+b+c)$

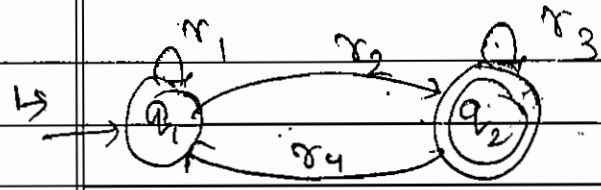


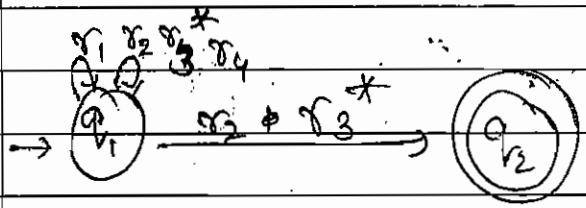












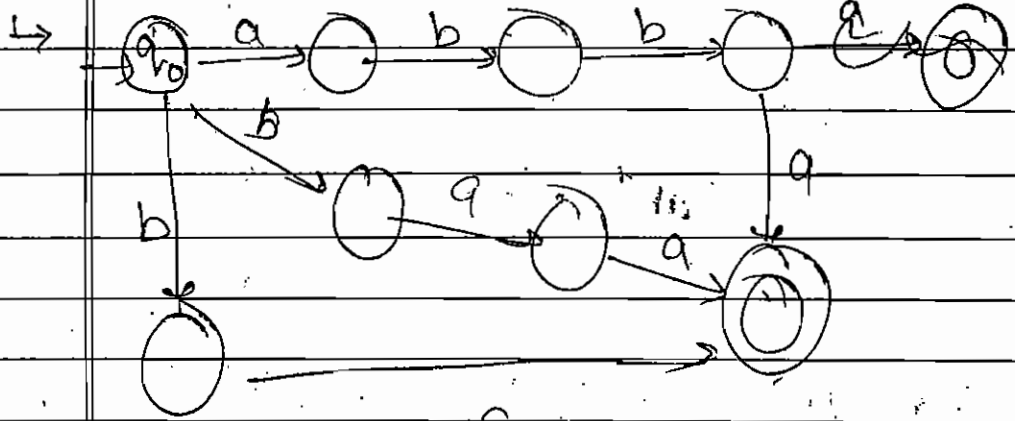
$r_1 \; r_2 \; r_3^* \; r_4$

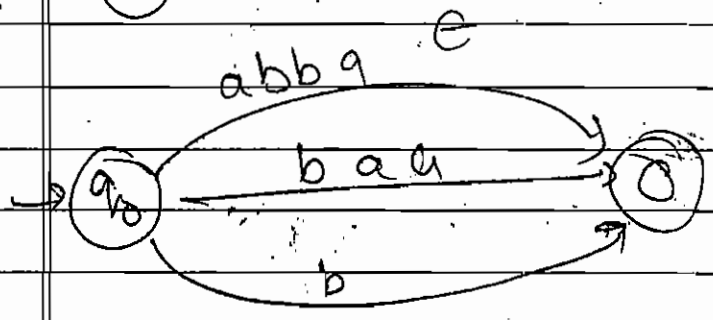
$r_2 + r_3^*$

$r = r_1^* \left(r_1 + r_2 r_3^* r_4 \right)^* r_2 r_3^*$

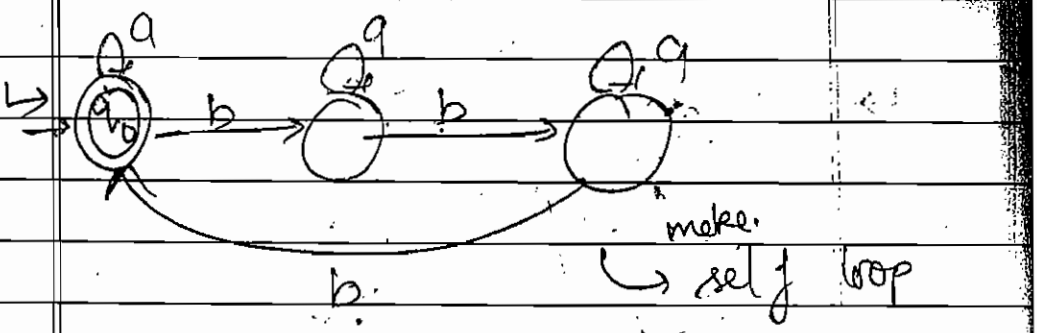
a, b

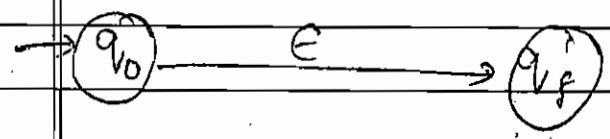
Ans

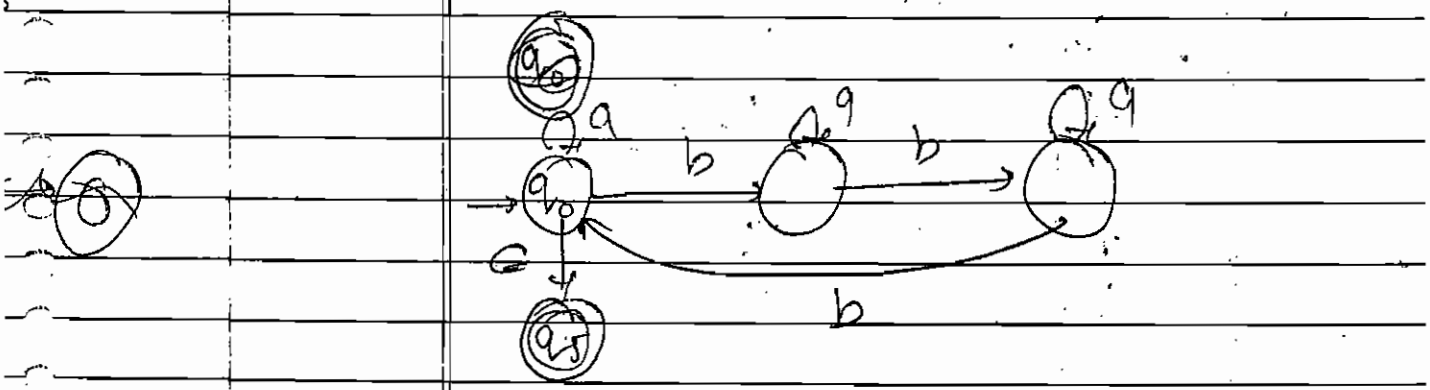


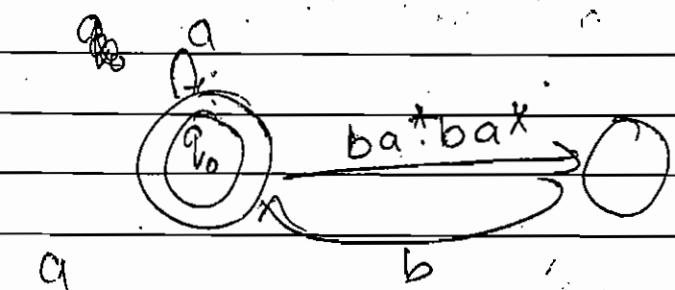


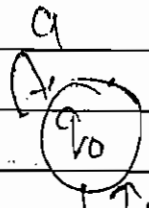
abba + baa + b





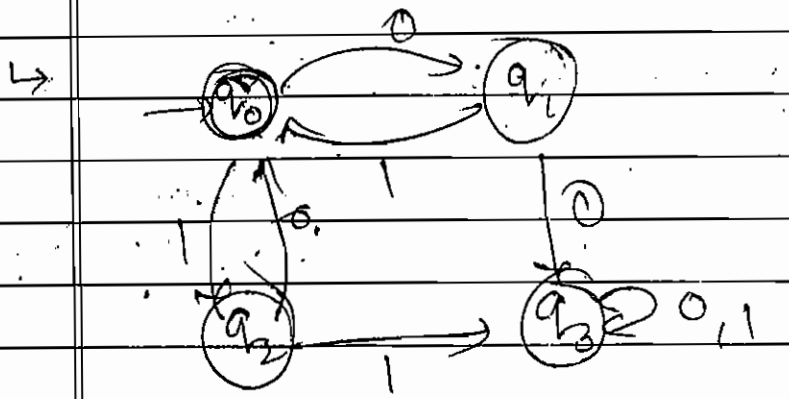


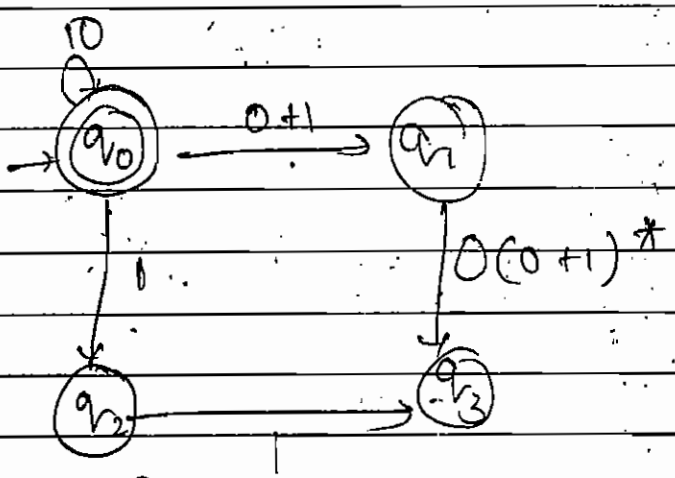


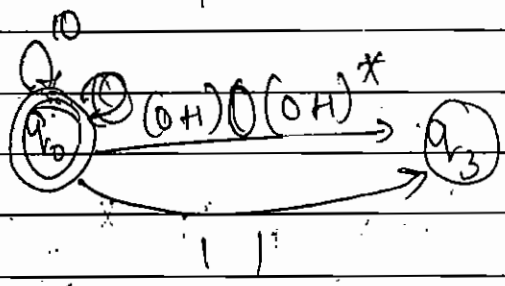


$ba^*ba^*b^*$

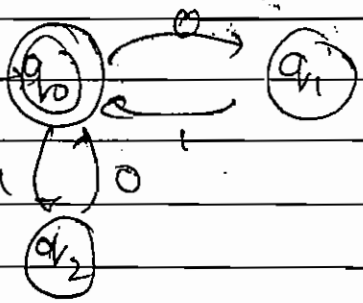
$a + ba^* ba^* b$



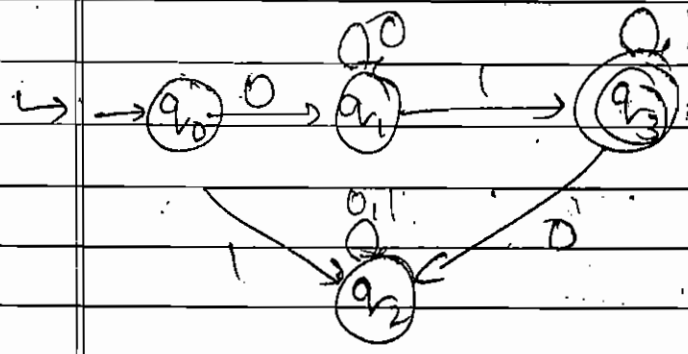


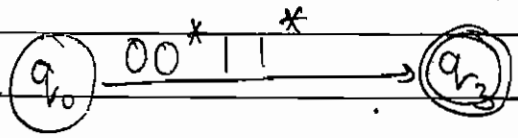


Note! If there is any dead lock state, delete it

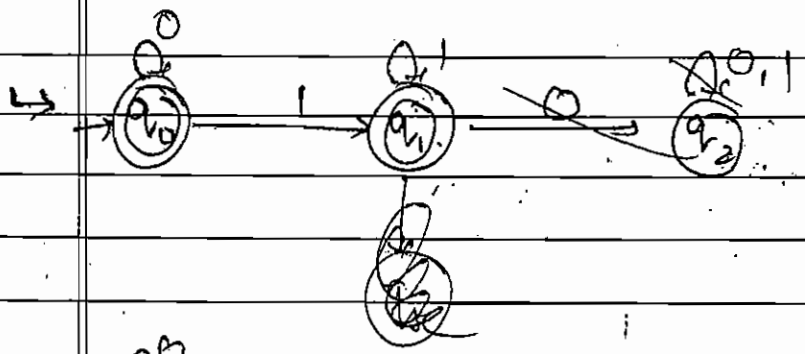


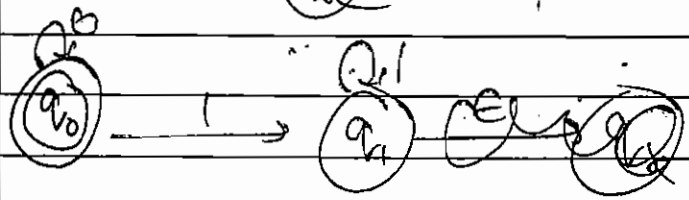
$\Rightarrow (01+10)^*$

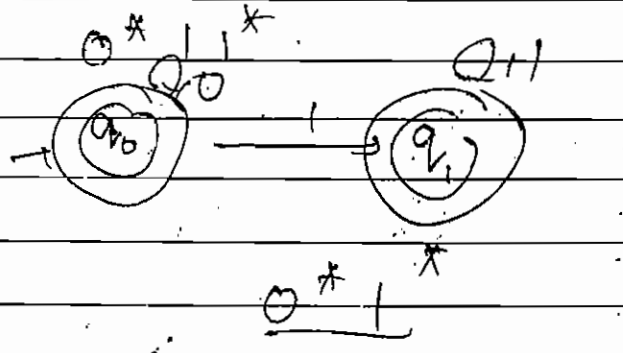




$(0+1+)$







0^*1

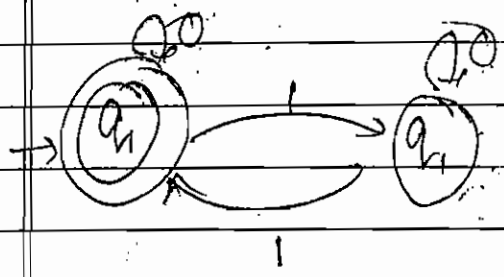
Date / /

Page

→ Find R.E for all (0's & 1's where no. of 1s are even in every string.

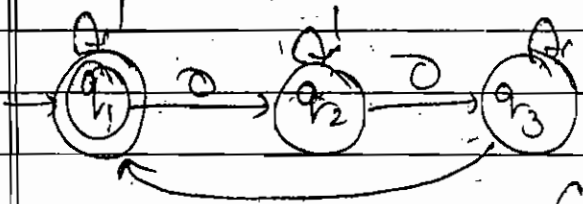
L = {0, 011, 1100, 01110}

$(0^*10^*10^*)^*$

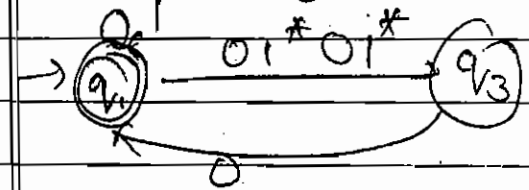


$(0 + 10^*1)^*$

↳ No. of 0s are divisible by 3



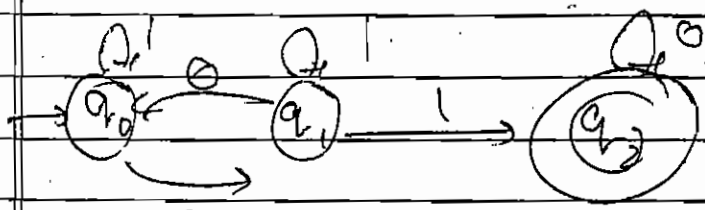
$(1^* + 01^*01^*0)^*$

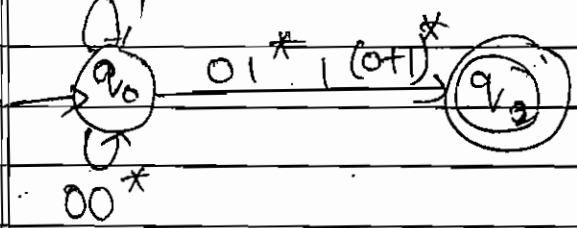


Date / /
Page

Note: First create FA then convert into R.E.

↳ Convert R.E into NFA then into RE.

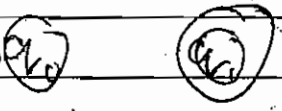


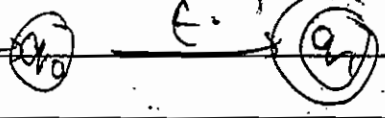


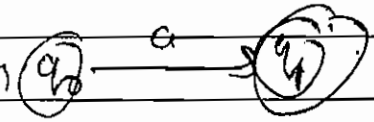
$X\left(1+010^{***}\right)^{*} 01^{*} \mid (0+1)^{*}$

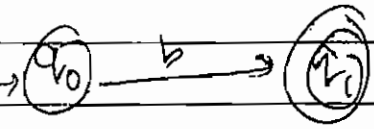
$\left(1+01^{*}0\right)^{*} 01^{+} (0+1)^{*}$

Conversion of R.E. to FA :-

(1) $\emptyset \Rightarrow$ 

(2) $\in \Rightarrow$ 

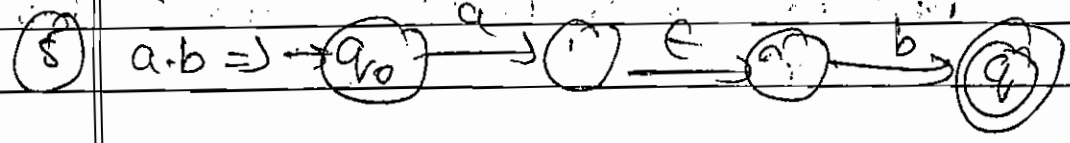
(3) $a \Rightarrow$ 

(4) $b \Rightarrow$ 

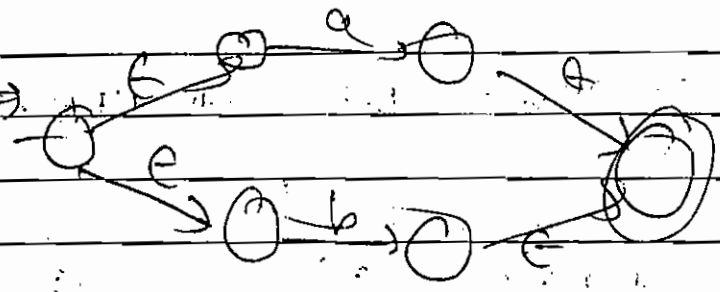
Date / /

Page

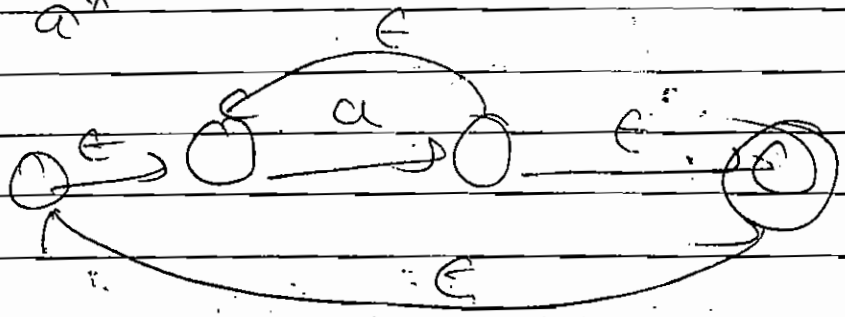
(5) $a.b \Rightarrow$



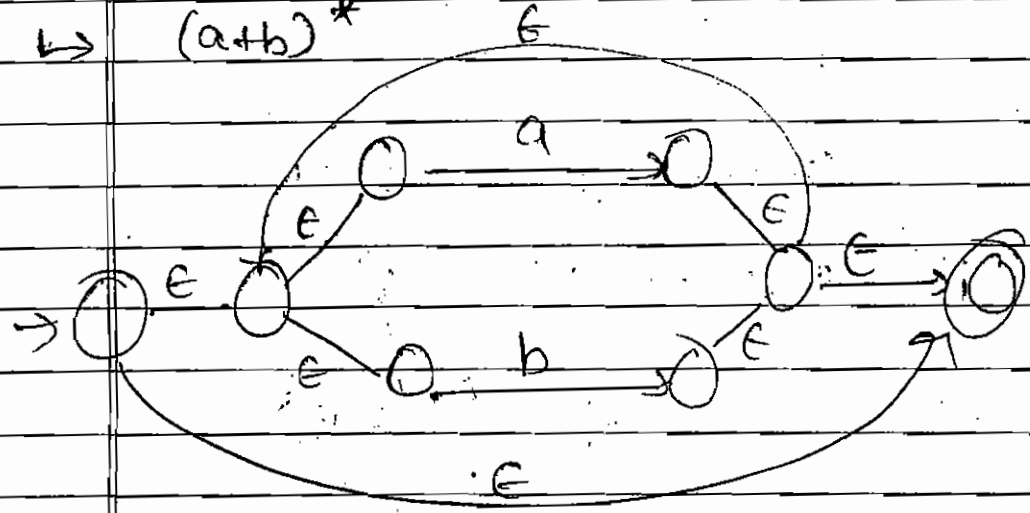
(6) $a+b \Rightarrow$

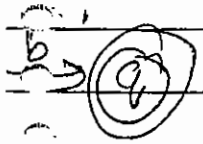


(7) a^*

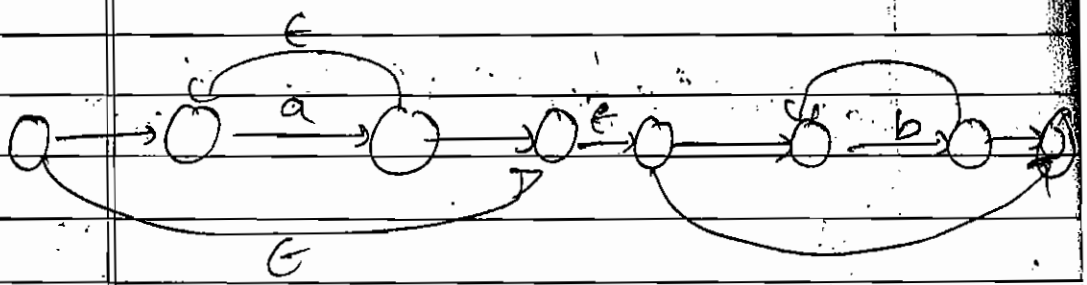


↳ $(a+b)^*$





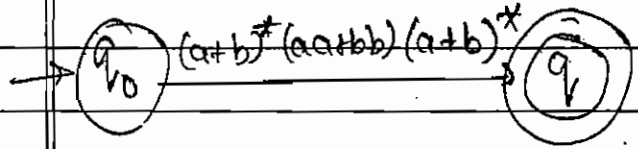
↳ $a^* b^*$

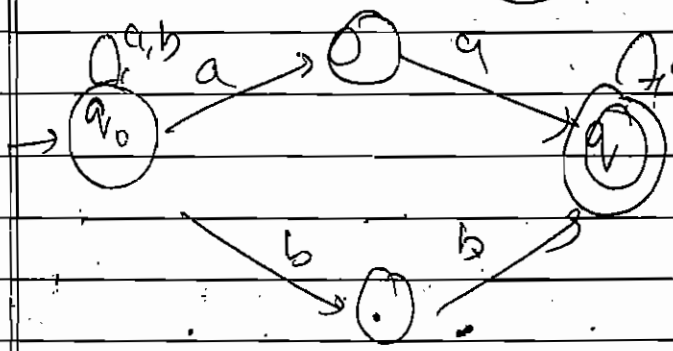


↳ Find no. of ~~string~~ states in NFA for following R.E.

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

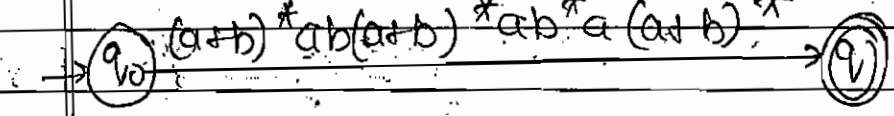
6

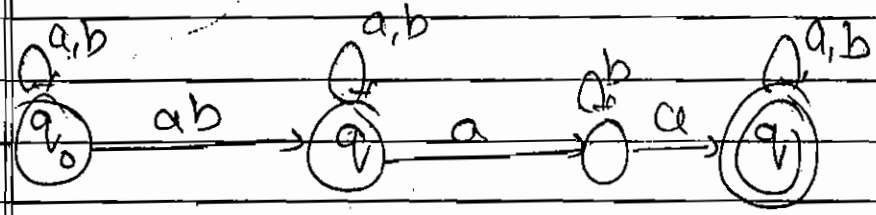


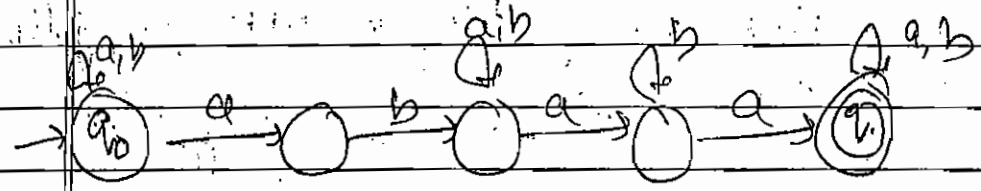


Date / /
Page

↳ $(a+b)^* ab(a+b)^* ab^* a(a+b)^*$

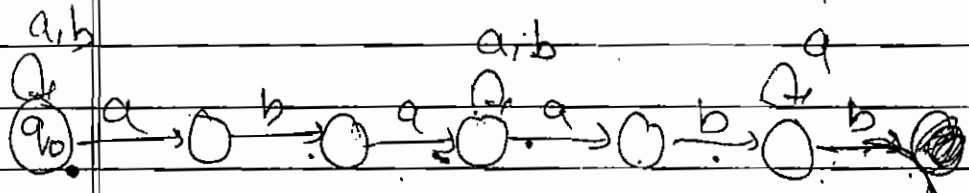


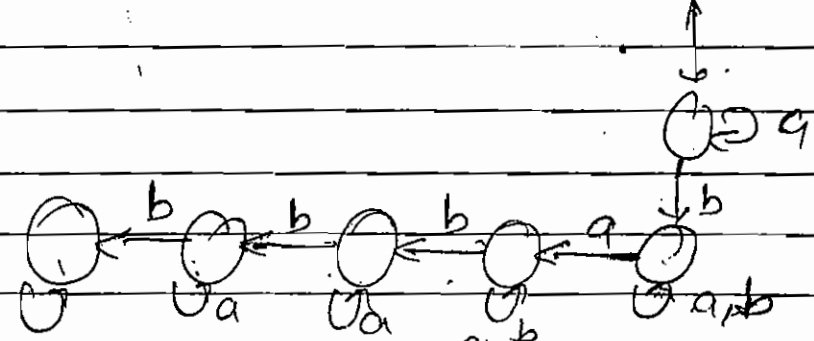




5 states

↳ $(a+b)^* aba(a+b)^* ab\,a^* ba^* b$



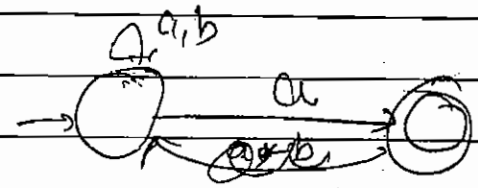


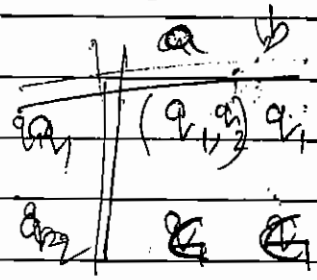
= 12

Date / /
Page

$[a+b]^*$ → No. of state = No. of individual symbol + 1.

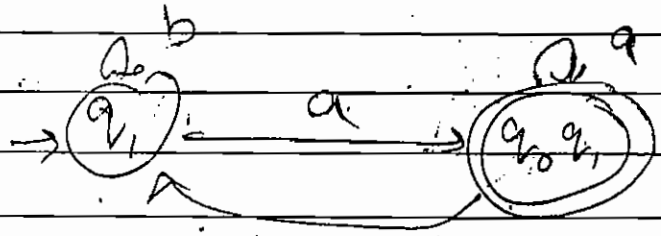
↳ $(a+b)^* a$





DFA

	a	b
→ q_1	$[q_1 q_2]$	q_1
$[q_1 q_2]$ q_2	$[q_1 q_2]$	$[q_1 q_2]$

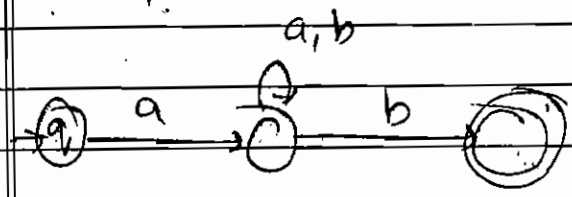


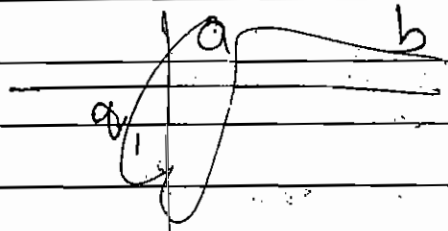
2 states

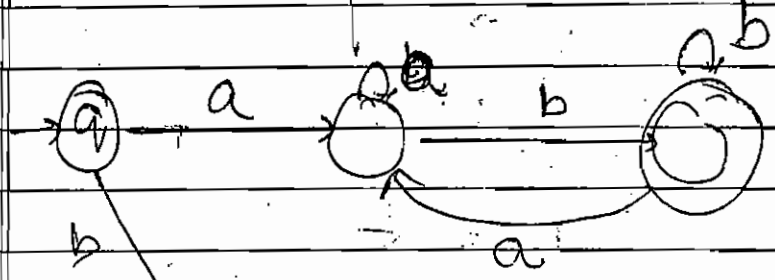
Date / /

Page

↳ a (a+b)* b

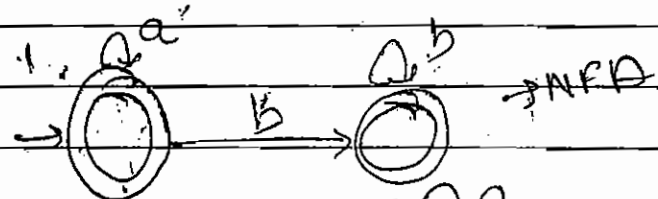


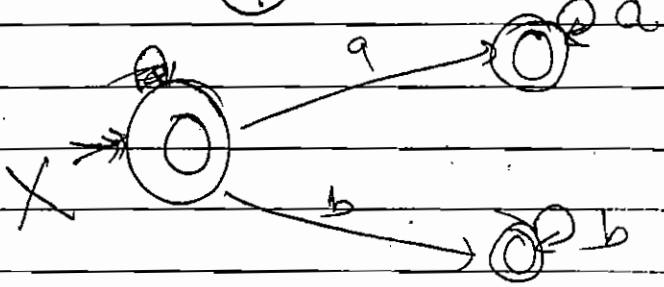


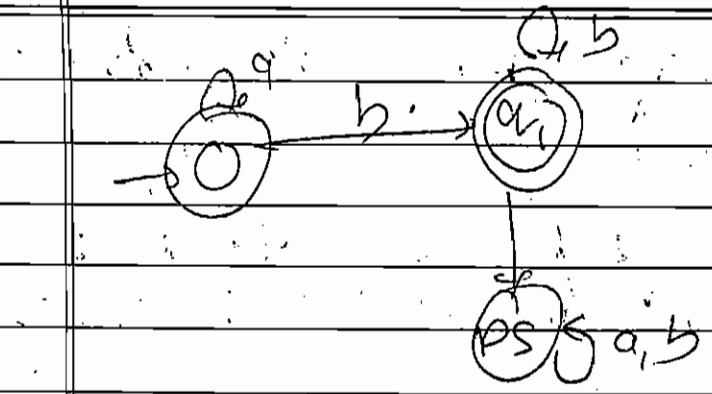


4 states

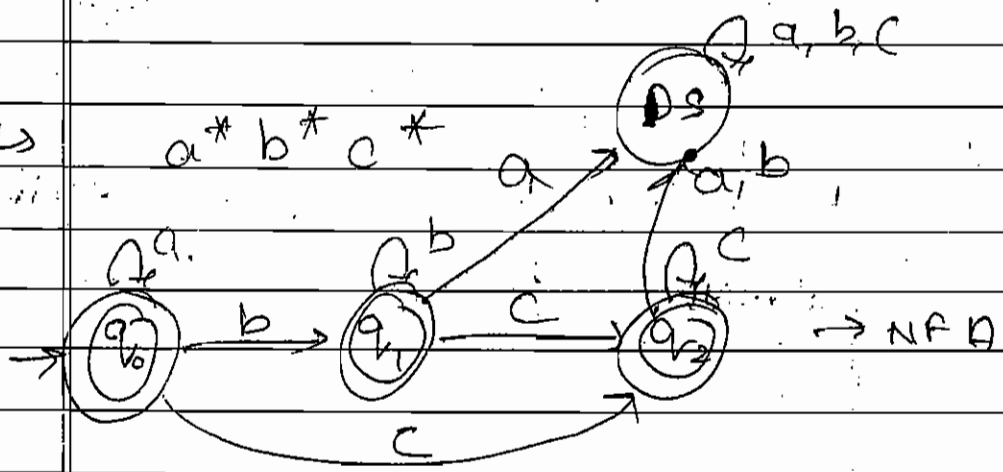
↳ a* b* 0, a, b







↳ $a^* b^* c^*$



↓
DFA

↳ $a^* b^* c^* d^*$

in NFA.

Note: No. of state = No. of symbols

NOS in DFA = No. of symbols + 1

Date / /
Page

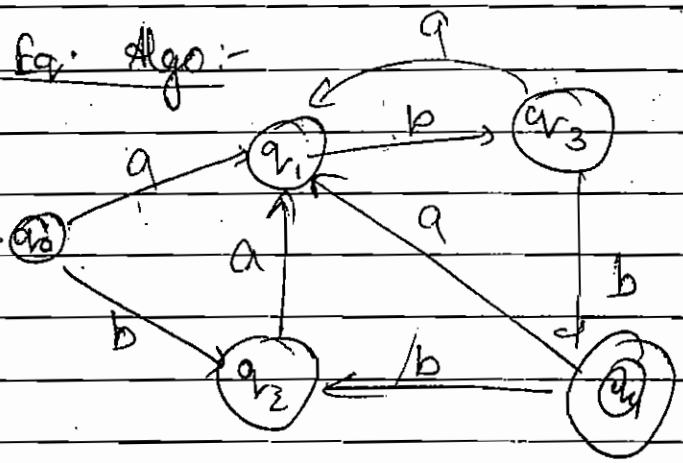
Conversing many states into one:

Equivalent States :- (Q_1 & Q_2)

Two state are said to be equivalent $\forall x \in \Sigma^*$ both Q_1 & Q_2 has to go to either final or non-final. But it is difficult to proof for infinite no. of alphabets hence we have to prove equivalent algorithms for identifying equivalent state.

1. State Equivalence algo.
2. Table filling algo.

State Eq. Algo :-



	a	b
q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	q_4
q_4	q_1	q_2

0. Equivalence = $\{q_0, q_1, q_2, q_3\}\ \{q_4\}$

1-Equivalence = $\{q_0, q_1, q_2\}\ \{q_3\}\ \{q_4\}$

2- Equivalence = $\{q_0, q_2\}\ \{q_1\}\ \{q_3\}\ \{q_4\}$

3- Equ = $\{q_0, q_2\}\ \{q_1\}\ \{q_3\}\ \{q_4\}$

k-Eq = — — —

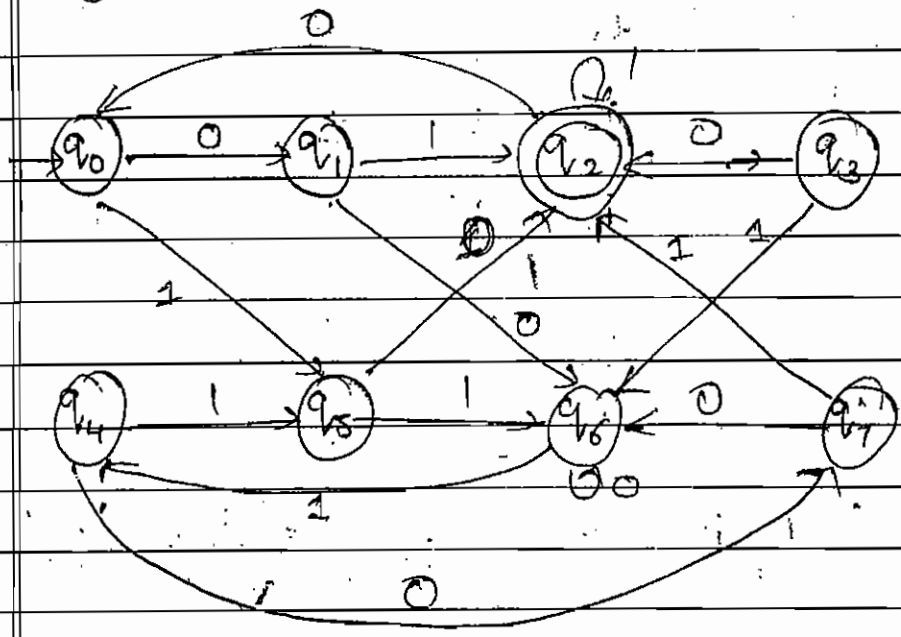
	a	b
$[q_0\ q_2]$	q_1	$q_0 q_2$
q_1	q_1	q_3
q_3	q_1	q_4
q_4	q_1	$q_0 q_2$

4 states

-> direct equivalence.

Date / /
Page

↳ Construct minimal DFA for following DFA.



	0	1
q_0	q_1	q_5
q_1	q_6	q_2
q_2	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

Date / /
Page

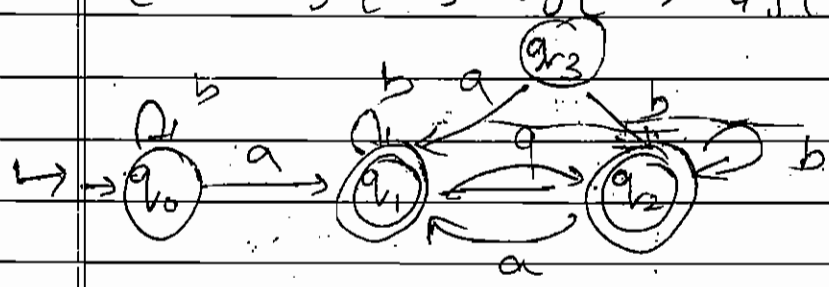
following

0-Equi = $\{q_0, q_1, q_5, q_6, q_4, q_7, q_2, q_3\}$

$= \{q_0, q_1, q_4, q_5, q_3, q_6, q_7\}\{q_2\}$

$\{q_3, q_5\}\{q_1, q_7\}\{q_0\}\{q_2\}\{q_4\}\{q_6\}$

$\{q_1, q_7\}\{q_3, q_5\}\{q_0, q_4\}\{q_6\}\{q_2\}$



(1) How many no. of states in minimal DFA.?

a) 2 b) 3 c) 4 d) none

(2)

0-Eq $\{q_0, q_3\}\{q_1, q_2\}$

$\{q_2, q_3\}\{$

$\{q_0\}\{$ $\}\{q_1, q_2\}$

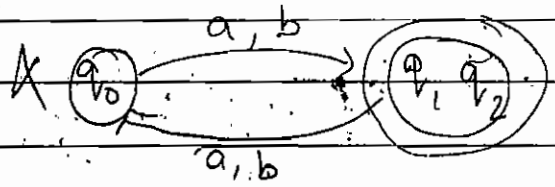
$\{q_0\}\{$ $\}\{q_1, q_2\}$

	a	b
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_1	q_3
q_3	q_1	q_2

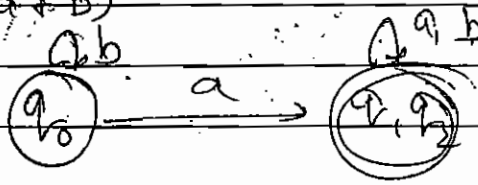
Date / /
Page

(2) Which lang. it accepting

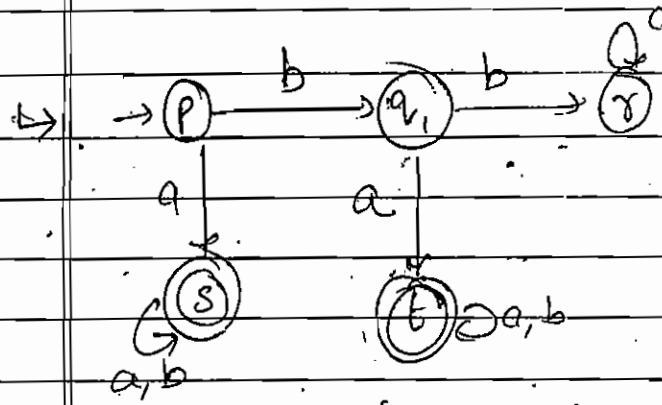
	a	b
q_0	$q_1 q_2$	q_0
$q_1 q_2$	$q_1 q_2$	$q_1 q_2$



a) $(a+b)^*$



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

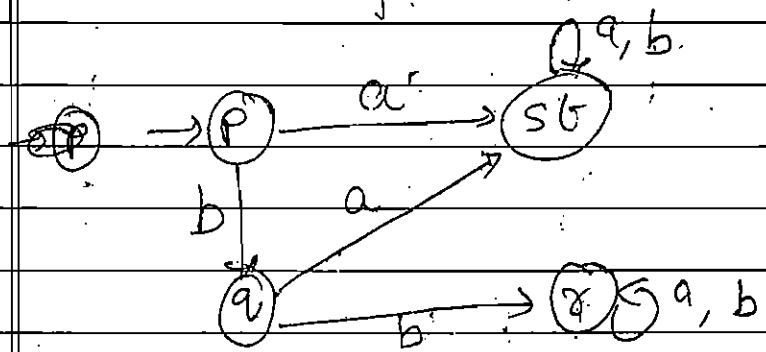


0- Eq. = { p, q, r } { s }

	a	b
p	s	q
q	t	r
r	r	r
s	s	s
t	t	t

$\{p\}$ $\{\{q\}\{r\}\{s,t\}$

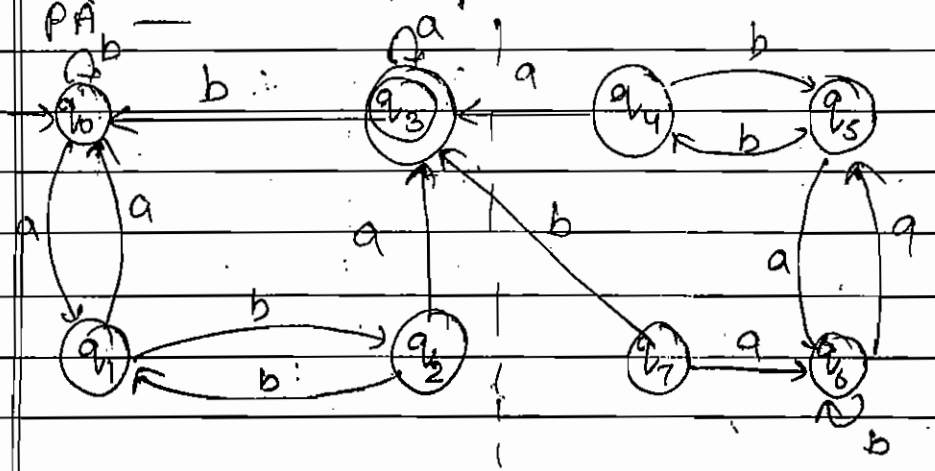
	a	b
p	st	q
(st)	st	st
q	st	r
r	r	r



Date 04/01/2011

Page

↳ Construct an equi. minimal DFA for following DFA —



0-Equi = $\{q_0, q_1, q_2, q_4, q_5, q_6, q_7\}$ $\{q_3\}$

	a	b
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

Date / /

Page

ex following

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

$\{q_2\}$ $\{q_4\}$ $\{q_3\}$

	a	b
$q_0 q_1$	$q_0 q_1$	$q_0 q_2$
$q_0 q_2$	q	

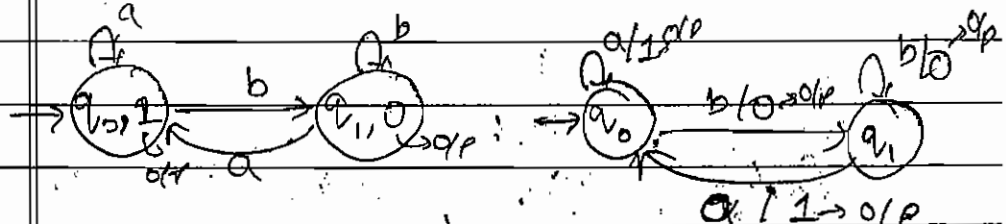
F/A with O/P :-

Moore M/C :

It is a mathematical model in which O/P is associated with state.

Mealy M/C :

It is a mathematical model in which O/P is associated with transition.



$(Q, \Sigma, \delta, q_0, \Delta, \lambda)$

$Q \rightarrow$ finite no. of states

$\Sigma \rightarrow$ i/p alphabet

δ : T.F. $Q \times \Sigma \rightarrow Q$

q_0 : initial state

Δ : O/P alphabet

λ : O/P func.

$\lambda : Q \times \Sigma \rightarrow \Delta$

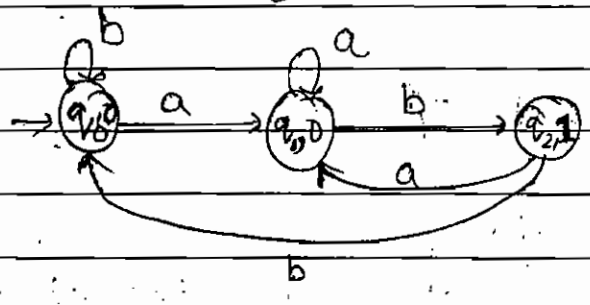
$\lambda : Q \rightarrow \Delta$

Date / /
Page

→ Moore & Mealy M/C's are DFA.

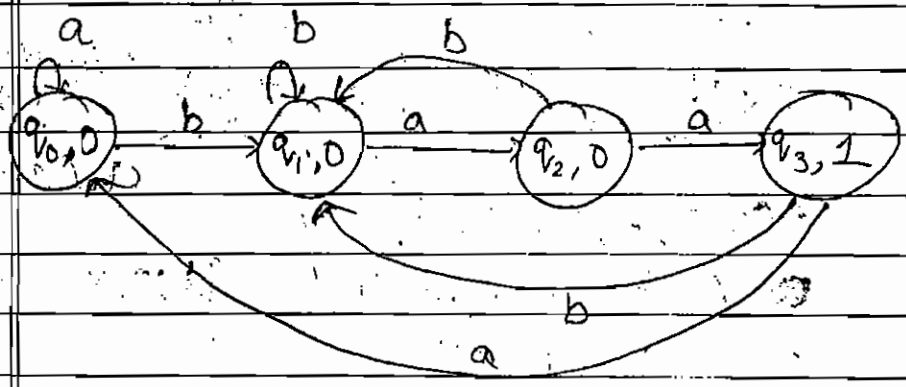
↳ Construct ~~Mealy~~ Moore M/C that takes all strings of as & bs as i/p and counts no. of occurrances of substring 'ab' for given string.

$\Sigma = \{a, b\}$ $\Delta = \{0, 1\}$



↳ Construct Moore M/C that ---- and counts no. of occ. of substring baa over the given string.

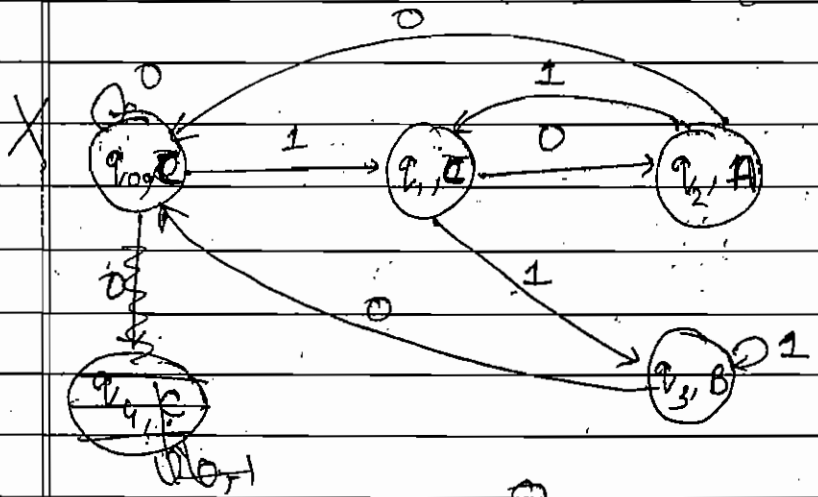
$\Sigma = (a, b)$ $\Delta = \{0, 1\}$

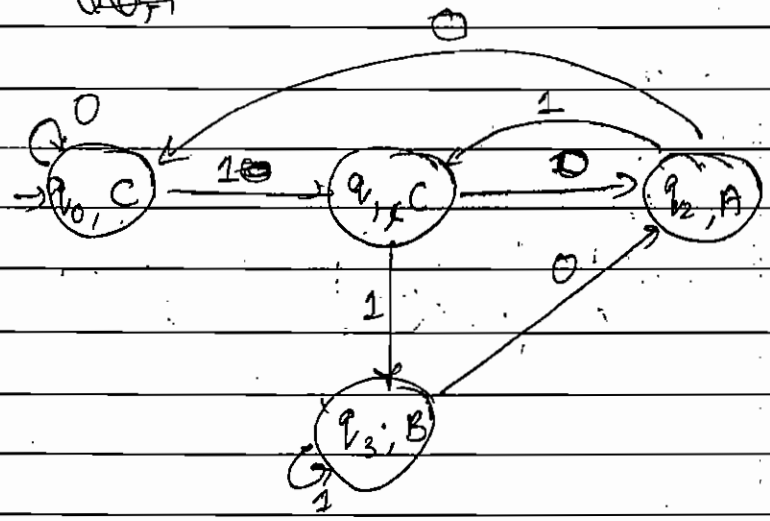


Date / /

Page

that takes all strings of 0s & 1s as i/p and produces A as o/p if i/p ends with 10 or produces B as o/p if i/p ends with 11 otherwise produces C as o/p. $\Sigma = \{0, 1\}$ $\Delta = \{A, B, C\}$





Date / /

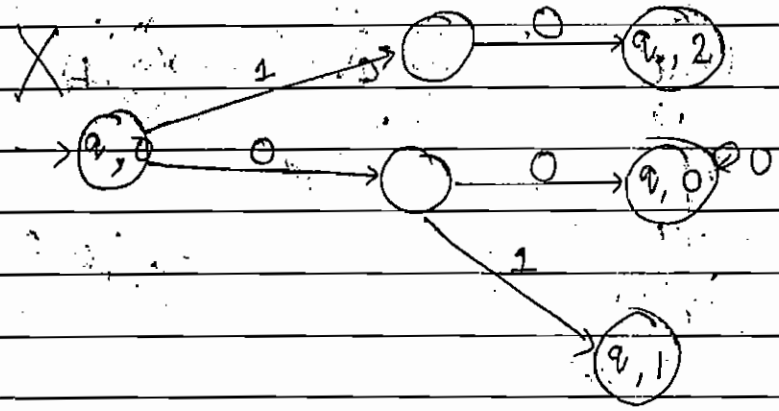
Page

↳ that takes all binary nos. as i/p and produces Residue modulo 3 as o/p.

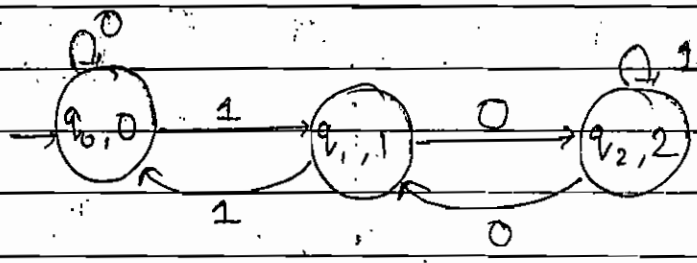
↓ remainder

00, 01, 10

$\Sigma = \{0, 1\}$ $\Delta = \{0, 1, 2\}$



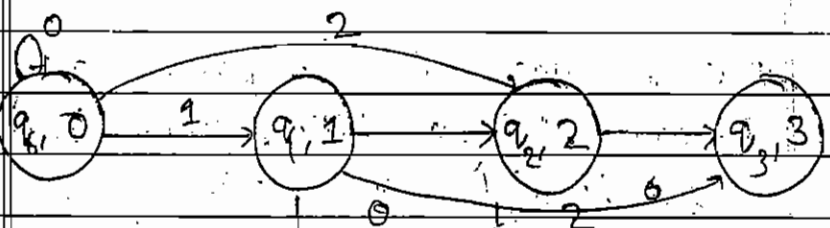
0011



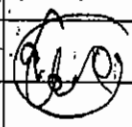
↳ Takes all base 3 nos. as i/p and produces Residue modulo 4 as o/p.

$\Sigma = \{0, 1, 2\}$ $\Delta = \{0, 1, 2, 3\}$

0 → 0
1 → 1
2 → 2
3 → 0
4 → 1
5 → 2



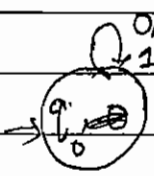
	0	1	2
$[q_0, 0]$	q_0	q_1	q_2
$[q_1, 1]$	q_3	q_0	q_1
$[q_2, 2]$	q_2	q_3	q_0
$[q_3, 3]$	q_1	q_2	q_3



Note: The moore m/c that takes all base m no. as i/p and produces residue modulo n as o/p requires 'n' no. of states.

↳ Construct Mealy m/c that produces 1's complement of given binary string as o/p.

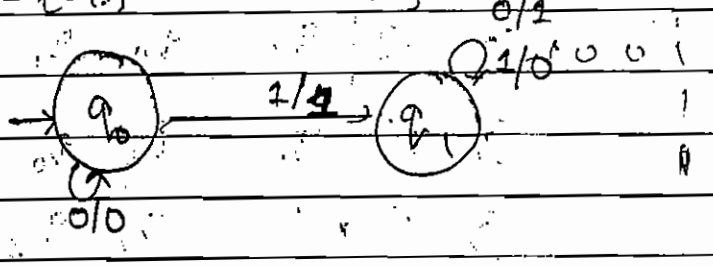
$\Sigma = \{0,1\}$ $\Delta = \{0,1\}$



2005

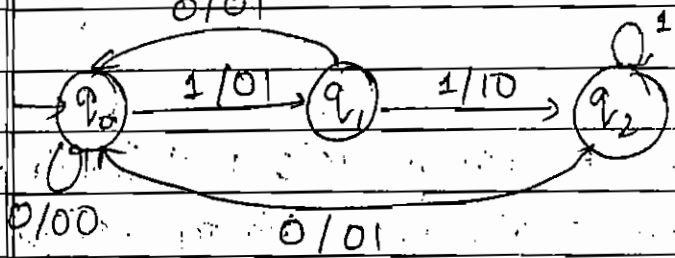
↳ that produces 2's complement of given binary string as O/P. Assume that we are reading string from LSB to MSB and end carry is discard.

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



Logic: from ~~MSB~~ LSB to MSB 0 will be 0 as it is, until first 1 comes. after 1 it will be like 1's complement.

↳ What is O/P produced by follow state M/C.



a) 11 → 01 b) 10 → 00

c) sum of present & previous bit.

d) NOT

0 → 01

01 ⇒ 01

10 ⇒ 01

11 ⇒ 10

00 ⇒ 00

Date / /

Page

1/1

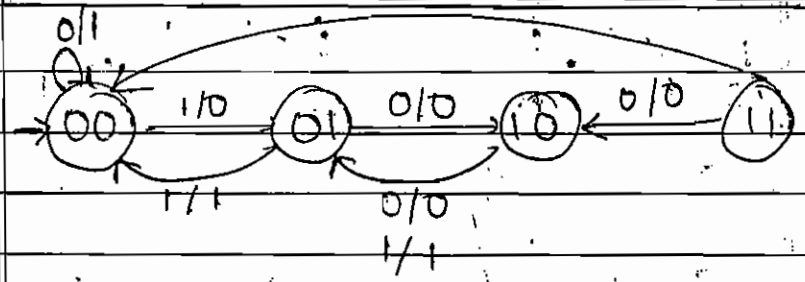
2009

given

we are

LSB

rd.

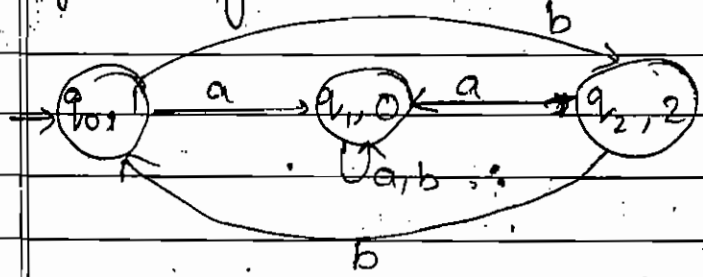


By reading 101, it produces 1 as O/P, so min^m. length = 3.

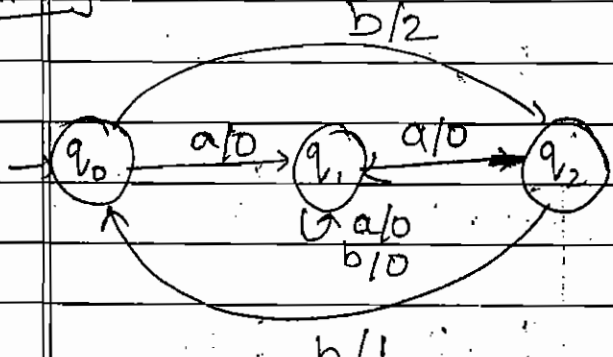
Mealy to Moore

Moore to Mealy M/C:

Construct an equi Mealy M/C for following Moore M/C.

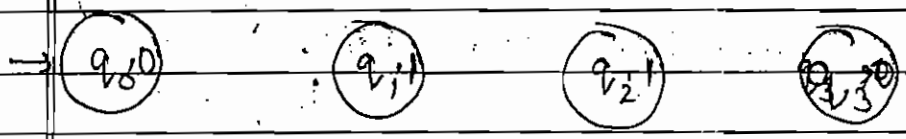


Mealy



Date / /
Page

Present State	i=0 N.S	i=1 N.S	O/P
q_0	q_3	q_2	0
q_1	q_1	q_1	1
q_2	q_2	q_2	1
q_3	q_0	q_3	0



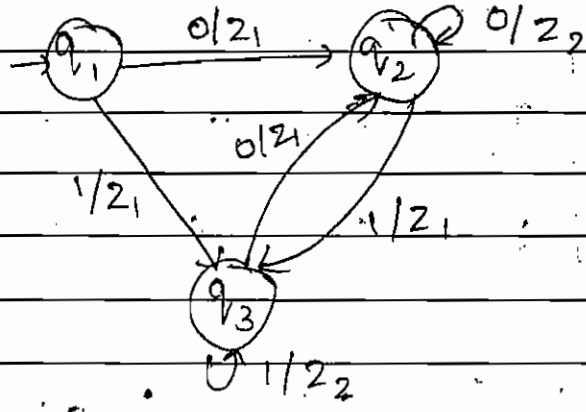
Mealy

P.S.	i=0 N.S O/P	i=1 N.S O/P
q_0	q_3, 0	q_2, 1
q_1	q_1, 1	q_1, 1
q_2	q_2, 1	q_2, 1
q_3	q_0, 0	q_3, 0

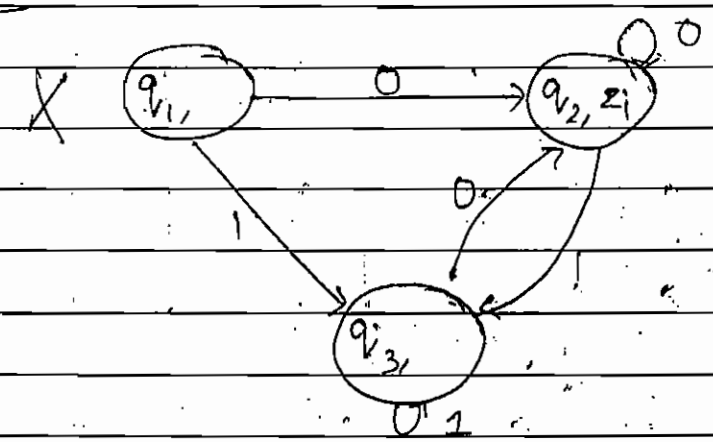
Date / /
Page

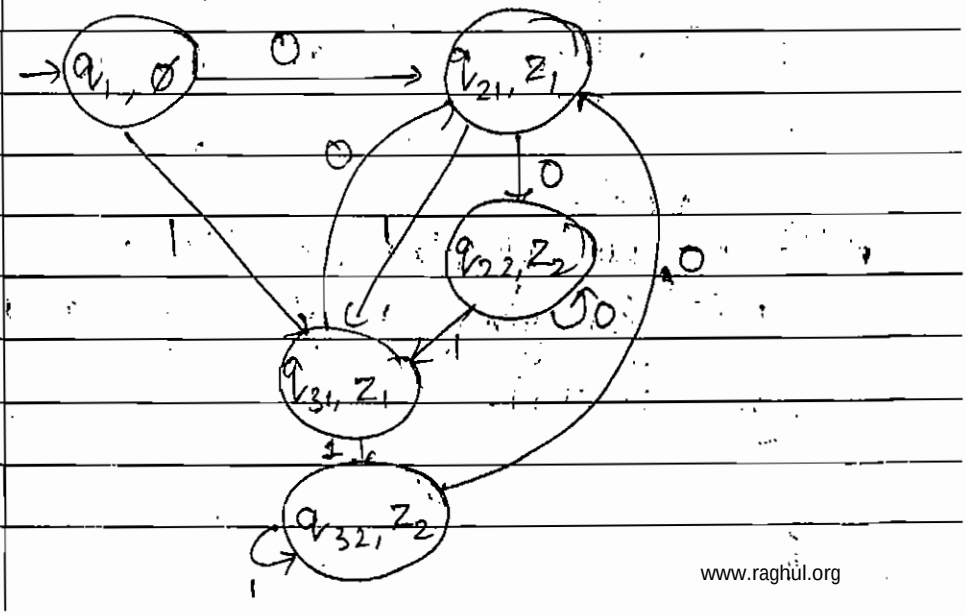
Mealy ⇒ Moore

↳ Construct equ. Moore for Mealy –



Moore:





Date / /
Page

↳

PS	i=0 NS	O/P	i=1 NS	O/P
q_1	q_3	0	q_2	0
q_2	q_1	1	q_4	0
q_3	q_2	1	q_1	1
q_4	q_4	1	q_3	0

PS	i=0 NS	i=1 NS	O/P
q_1	q_3	q_{20}	1
q_{20}	q_1	q_{40}	0
q_{21}	q_1	q_{40}	1
q_3	q_{21}	q_1	0
q_{40}	q_{41}	q_3	0
q_{41}	q_{41}	q_3	1

Note: The no. of states possible in a Moore M/C for given n state, m O/P Mealy M/C is $(n \times m)^m$.

Date / /
Page

↳ How many Mealy M/c's are possible with two states x & y for the i/p alphabet {a, b}, o/p alpa {0, 1}, where x is always initial state.

	i=a, o/p	i=b, o/p
x	2 x 2 = 4	2 x 2 = 4
y	2 x 2 = 4	2 x 2 = 4

$= 4^4 = 256$

↳ How many Moore M/c are possible for above ques?

	i=0	i=1	o/p
x	2	2	3 → may be ∅ also
y	2	2	2

$= 2^5 \times 3 = 32 \times 3 = 96$

Moore

Mealy M/c

Date / /

Page

Regular Languages:-

↳ Which of the following lang. are regular and which are non-regular?

(1) $L = \{a^n b^n \mid n \leq 1\}$

(2) $L = \{a^n b^n c^n \mid n \leq 1\}$

(3) $L = \{a^n b^m \mid n, m \leq 1, n < m\}$

(4) $L = \{a^i b^j \mid i, j \leq 1, i > j\}$

(5) $L = \{a^n b^m c^k \mid n, m, k \geq 1\}$

(6) $L = \{a^i b^{2j} \mid i, j \geq 1\}$

N (7) $L = \{a^n b^{4m} \mid n > m\}$

N (8) $L = \{a^i b^j \mid i \neq j\}$

X (9) $L = \{w \in \{a, b\}\}$

N (9) $L = \{ww^R \mid w \in \{a, b\}^*\}$

$w^R \rightarrow$ reverse.

Date / /

Page

(10) $L = \{ x.2y \mid x, y \in \{0,1\}^* \}$

Regular

(11) $L = \{ xx^R \mid x \in \{0,1\}^*, |x| = 2 \}$
↳ length of string.

N (12) $L = \{ xx \mid x \in \{a,b\}^* \}$

(13) $L = \{ w \mid w \in \{a,b\}^*, |w| \geq 10 \}$

(14) $L = \{ w \mid w \in \{a,b\}^*, |w| \bmod 15 = 0 \}$

N (15) $L = \{ a^{2^n} \mid n \geq 1 \}$

(16) $L = \{ a^{3^n} \mid n \leq 1 \}$

(17) $L = \{ a^{2n} \mid n \geq 1 \}$

N (18) $L = \{ a^{\ln n} \mid n \geq 1 \}$

N (19) $L = \{ a^{n^2} \mid n \geq 1 \}$

N (20) $L = \{ a^p \mid p \text{ is a prime no.} \}$

Date / /
Page

21) $L = \{ a^k \mid k \text{ is a odd no.} \}$

22) $L = \{ a^m \mid m \text{ is an even no} \}$

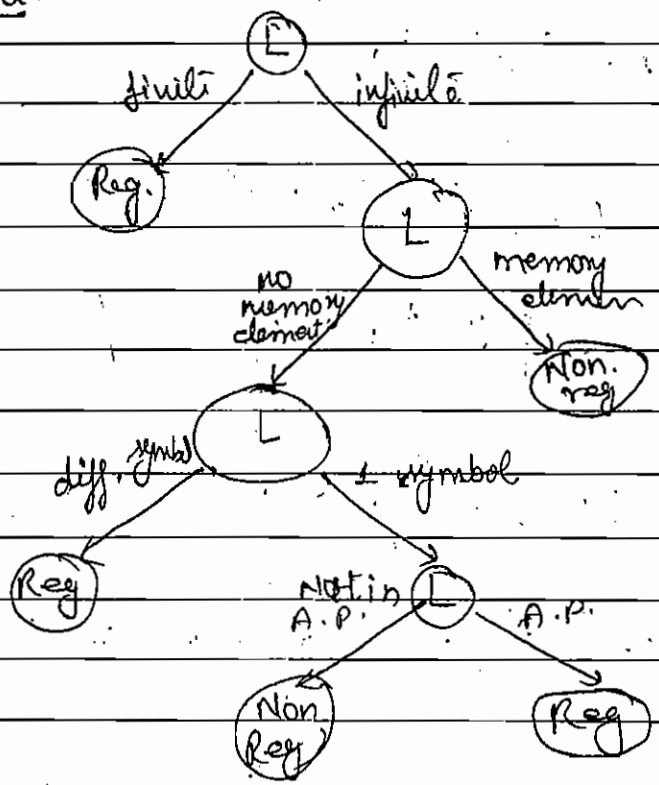
23) $L = \{ \text{total population of INDIA} \}$.

24) $L = \{ \text{Property of Indian Cricketers} \}$.

25) $L = \{ \text{total no. of java programs written in the year 2006} \}$

26) $L = \{ a^i b^j \mid i, j \geq 1 \}$

Method:



→ Powers should be in A.P. for regularity.

27) $L = \{ a^m b^k c^l \mid m = k+l \}$

28) $L = \{ a^i b^{2j} c^{3k} \mid i > j > k \}$

29) $L = \{ x \mid x \in \{a, b\}^* \; n_a(x) > n_b(x) \}$

30) $L = \{ x \mid x \in \{0, 1\}^* \; n_0(x) = 2 n_1(x) \}$

31) $L = \{ a^i b^{2j} \mid i \neq j \}$

32) $L = \{ a^2, a^5, a^8, \ldots \}$

33) $L = \{ 0^{i^i} \mid i \geq 1 \}$

34) $L = \{ a^m b^{n+m} c^n \mid m, n \geq 1 \}$

35) $L = \{ x \mid x \in \{0,1\}^* \text{ and } x \text{ has equal no. of 0s \& 1s} \}$

36) $L = \{ x \mid x \in \{a, b\}^* \; |x| = 3 \bmod 7 \}$

37) $L = \{ ww \mid w \in \{0,1\}^* , |w| = 1 \}$

38) $L = \{ a^n b^n c^n d^n \mid n \geq 1 \}$

39) $L = \{ a^{2n+1} \mid n \geq 1 \}$

Closure Properties of Regular Languages :-

(1) Union operation

Union of two regular lang. is always regular hence R.L. are closed under union operation.

$R_1 \cup R_2 = R_3$

$\{ a^n b^m \mid n, m \geq 0 \} \cup \{ c^n d^m \} \Rightarrow$

$\downarrow r_1 \qquad \downarrow r_2 \qquad \downarrow r$

(2) Concatenation Op :-

Conc. of two R.L. is always regular hence R.L. are closed under Conc. operat

$R_1 \cdot R_2 = R_3$

$\{a^n\} \{b^m\}$

Date / /
Page

③ Kleen Closure

Kleen closure of R.L. is always regular hence R.L. are closed under Kleen closure op.

$L_1 \Rightarrow \gamma_1$

$L_1^* \Rightarrow \gamma_1^*$

④ Positive Closure :-

+ve closure of R.L. is always regular hence R.L. are closed under +ve closure.

$L_1 \Rightarrow \gamma$

$L_1^+ : \gamma_1^+$

⑤ Complement Op :-

Complement of R.L. is always regular hence R.L. are closed under complement op.

$L_1 \Rightarrow$ Reg.

$\overline{L_1} \Rightarrow \Sigma^* - L_1 \Rightarrow$ Reg.

⑥ Intersection Op :-

L_1 : Reg. L_2 : Reg.

$\overline{L_1}$ $\overline{L_2}$

$\overline{L_1} \cup \overline{L_2} \Rightarrow \overline{L_1 \cup L_2}$

$\Rightarrow L_1 \cap L_2$

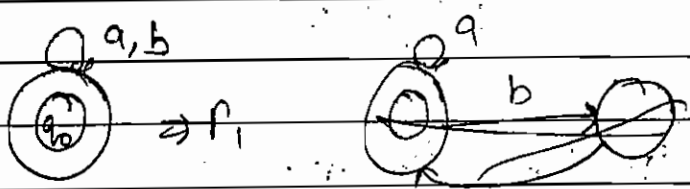
R.L. are closed under Intersection op.

Date / /

Page

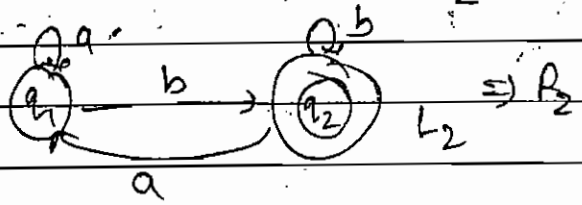
↳ How many no. of states there in min^m. DFA that accept $L_1 \cap L_2$ where L_1 is $(a+b)^*$ & L_2 is $(a+b)^* b$

$(a+b)^*$



$\Rightarrow F_1$

L_1



$L_2 \Rightarrow F_2$

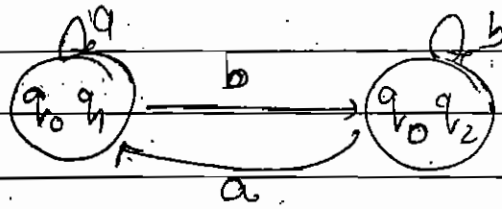
$L_1 \cap L_2 = F_1 \times F_2 = \{Q, \Sigma, \delta, q_0, f\}$

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

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

$q_{v_0} = \{q_0 q_1\}$

$f = \{q_0 q_2\}$

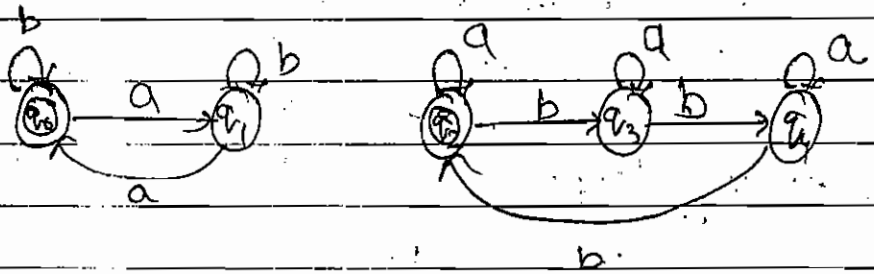


$\delta(q_0, a) = q_0$

$\delta(q_1, a) = q_1$

Date / /
Page

Ques. Construct DFA that accepts all strings of as & bs where no. of as are divisible by 2 & no. of bs are divisible by 3 in every string.



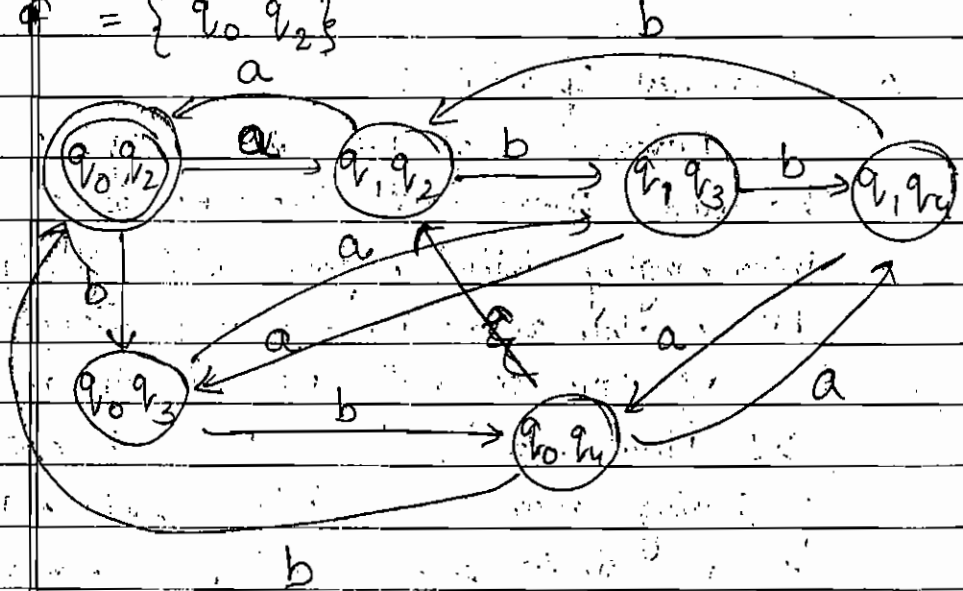
$Q = \{q_0, q_1\} \times \{q_2, q_3, q_4\}$

$= \{q_0 q_2\}\{q_0 q_3\}\{q_0 q_4\}\{q_1 q_2\}\{q_1 q_3\}\{q_1 q_4\}$

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

$q_0 = \{q_0 q_2\}$

$F = \{q_0 q_2\}$



Date / /
Page

↳ How many no. of states in min^m DFA $\Sigma = \{a, b\}$ where no. of a's are divisible by 6 & b's are divisible 8.

$2 \mid 6, 8$
$2 \mid 3, 4$
$3 \mid 3, 4$
1

48 states $= 6 \times 8$

(7) Diff. O/P:-

L_1 RL L_2 RL

$L_1 - L_2 = (L_1 \wedge \overline{L_2})$

Let L_1 & L_2 are R.L. &, diff. of L_1 & L_2 is always reg., Hence reg. R.L. are closed under diff. operation.

(8) Reversal Operation:

Reversal of R.L. is always regular because there exist some FA which is interchanging initial & final states of original FA & also reversing the edge directions.

If the original FA contain more ≥ 1 final state, then there is chance of having more than 1 initial state. Make it as single initial state, by taking new initial state with $\in$ transition.

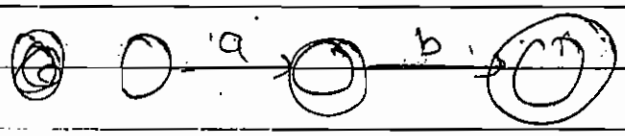
Date / /
Page

DFA
Possible

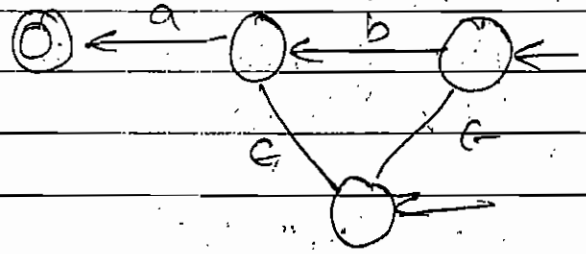
$L = \{a^n b^m\}$

$L^R = \{b^m a^n\}$

ab



ab^R



$L_1 \& L_2$ is closed

(9) Subset Operation :-

Subset of R.L. is may or may not regular hence R.L. are not closed under subset operation.

Regular is original means, more chance. Make easy

$a^* b^* \subseteq (a+b)^*$ $\quad \{a^n b^n \mid n \geq 1\} \subseteq a^* b^*$

$\{a^n b^m \mid n, m \geq 1\} \subseteq (a+b)^*$ $\quad \{a^{2^n} \mid n \geq 1\} \subseteq a^*$

↓
R.L.

$\{a^n b^n \mid n \geq m\} \subseteq (a+b)^*$

↓
Non R.L.

Date / /
Page

(10) Infinite Union & Infinite Intersection.

R.L. are not closed under above two operations. Because

$L_1 \cup L_2 \cup L_3 \cup \dots \cup L_\infty \Rightarrow$ Non Reg.

$L_1 \cap L_2 \cap L_3 \cap \dots \cap L_\infty \Rightarrow$ Non Reg.

Grammer :-

Set of rules use to describe strings of the lang.

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

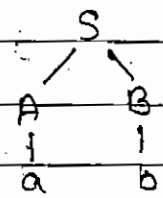
V → Set of variables or Non terminals

T → set of terminal

P ⇒ No. of production

S → Starting Symbol

$S \to AB$
$A \to a$ ⇒
$B \to b$



Parse Tree:

Tree representation of derivation.

Date / /
Page

↳ Identify the lang, represented by following grammer.

$S \rightarrow 0S1 / \epsilon$

$\{0^n 1^n \mid n \geq 0\}$

↳ $S \rightarrow 0S1S / 1S0S / \epsilon$

be strings

0150S1 1S0S

0101 10

$\{0^n 1^m \mid n, m \geq 0\}$

n terminals

↳ $\{\epsilon, 01, 10, 1100, \ldots\}$

↳ $S \rightarrow 0S11 / 011$

000111111

$L = 0^n 1^{2n} \mid n \geq 1$

↳ $S \rightarrow aSa / bSb / \epsilon$

ion.

$L = \{\epsilon, aa, bb, abba, baab, abaaba, \ldots\}$

Palindrom.

↳ Find the grammer that generate $\Sigma = \{a, b\}$ all string of a's & b's including ϵ.

$(a+b)^*$

$S \rightarrow asa/bsb/asb/bsa/\epsilon$ X

$S \rightarrow as/bs/a/b/\epsilon$

↳ Each string starting with a ending with b.

$S \rightarrow aAb$

$A \rightarrow aA/bA/\epsilon$

↳ Each string starting & ending with diff symbol.

$S \rightarrow aAb / bAa$

$A \rightarrow aA/bA/\epsilon$

↳ Each string starts & ends with same symbol

$S \rightarrow aAa / bAb / a/b$

$A \rightarrow aA/bA/\epsilon$

↳ length of string is exactly 3.

X $S \rightarrow aAb/bAa/$

$A \rightarrow a/b/\epsilon$

$S \rightarrow AAA$

$A \rightarrow a/b$

Date / /
Page

↳ At least 3

$S \to AAA|B$

$A \to a/b$

$B \to aB/bB/\epsilon$

↳ At most 3.

dealing with

$S \to A/AA/AAA$ ✗

$A \to a/b$

$S \to AAA$

$A \to a/b/\epsilon$

↳ L(S) is divisible by 3.

$S \to AAAS/\epsilon$

$S \to a/b$

or

$S \to AS/\epsilon$

$A \to BBB$

$B \to a/b$

↳

↳ $L = \{a^n b^n / n \geq 1\}$

no symbol

$S \to aSb/ab$

↳ $L = \{a^n b^n c^m / n, m \geq 1\}$

$S \to aS_1bS_2C$ ✗

$S_1 \to aS$

$S \to AB$

$A \to aAb/ab$

$B \to cB/c$

Date / /
Page

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

$S \rightarrow AB$
$A \rightarrow aAb \mid ab$
$B \rightarrow cBd \mid cd$

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

~~$S \rightarrow AB$~~
~~$A \rightarrow aBd \mid ad$~~
~~$B \rightarrow bBc \mid bc$~~

$abBcd$

a

$S \rightarrow aAd \mid asd$
$A \rightarrow bAc \mid bc$

↳ $L = \{a^m b^{m+n} c^n \mid m, n \geq 1\}$
$a^m b^m b^n c^n$

$S \rightarrow AB$
$A \rightarrow aAb \mid ab$
$B \rightarrow bBc \mid bc$

↳ $L = \{a^{m+n} b^n c^m \mid n, m \geq 1\}$
$a^m a^n b^n c^m$

$S \Rightarrow aAc \mid aSc$

$A \Rightarrow aAb \mid ab$

↳ $S \to aaS \mid saS \mid sbs \mid sab \mid asa \mid abs \mid a \mid b$

identify total no. of strings of length 4, ~~gen~~ is -

a) 3 b) 5 c) 8 d) None.

~~Type of Grammers~~

① Reg. Lan. (3)

⇓

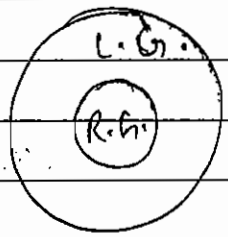
R.G.

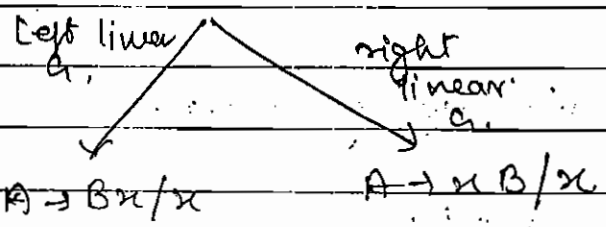
Type3 G.

$A \to xB \mid Bx \mid x$

$x \in \Sigma^*$

$A, B \in V$





Note! L.H.S. exactly 1 variable, R.H.S. atmost 1 Var.

Linear G.! - same, but linear Gr. need not to be R.G.

② CFL (2)

⇓

CFG

2G

$A \to \alpha$

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

Date / /
Page

(3) CSL (1)
⇕
CSG
↓
$\alpha \to \beta \quad \alpha, \beta \in (V+T)^+$
$|\alpha| \leq |\beta|$

NOTE

(4) R.E.L. (0)
⇕
Unrestricted G.
Type 0

$\alpha \not\to \beta \quad \alpha \in (V+T)^+$
$\beta \in (V+T)^*$

↳ Ty. Tell the type of grammer

1) ⓐ $S \to ABC / \epsilon$
$A \to BC / b$
$B \to aB / b$
$C \to a$
ans C.F.G.

Date / /
Page

2) $S \to ABC / \epsilon$

$aA \to BC / bb$

$bB \to aBa / ba$

$aCb \to abab$

Note: S derives ϵ in a valid CSG production, in that case S should not be presented on RHS part of any other grammer productions.

↳ Identify type of grammer, possible type no & higher type no, satisfied by given grammar -

$S \to ABC$

$A \to ab$

$B \to b$

$C \to ac / A$

Ans :- CFG, possible type 2, 1, 0, higher 2.

Linear Grammer:

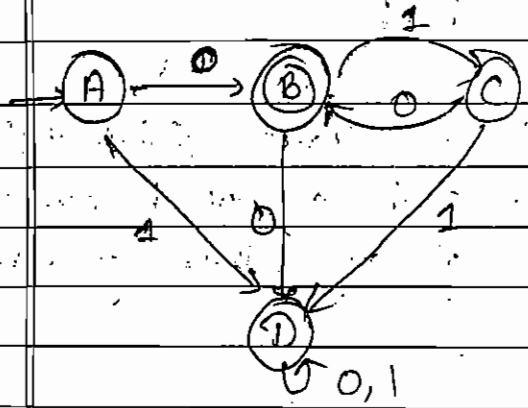
In any grammer if we find exactly one variable on LHS and atmost 1 variable on RHS, or

All RG are linear, all L.G. need not to be RG.

Date / /
Page

FA → RG

↳ Construct equivalent RG for following FA:



G = (V, T, P, S)

V = no. of states

T = ∀{0, 1}

P

S = A

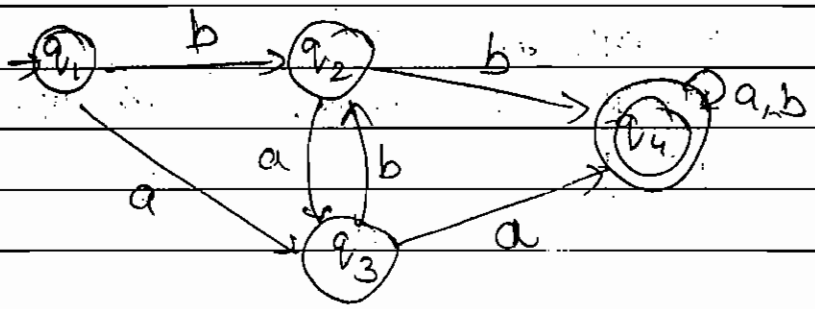
A → 0B / 1D / 0

B → 0D / 1C

C → 0B / 1D / 0

D → 0D / 1D

↳



$q_1 \to a q_3 \mid b q_2$

$q_2 \to a q_3 \mid b q_4 \mid b$

$q_3 \to a q_4 \mid b q_2 \mid a$

$q_4 \to a q_4 \mid b q_4 \mid a \mid b$

↳ How many max^m no. of productions are possible in regular grammar equivalent to given n state DFA, over the i/p alphabet $\{a, b, c\}$, where q_i is always initial state.

$V = n$

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

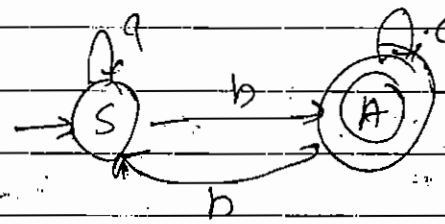
$S = q_i$

Ans:- $6n + 1$ (ϵ-also because initial state is final state also)

↳ No. of states for following RG:—

$S \to aS \mid bA \mid b$

$A \to aA \mid bS \mid a$



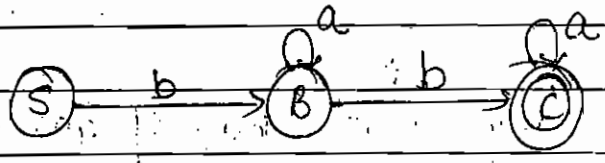
In RHS a or b are present so we can create FA.

→ $S \to bB$

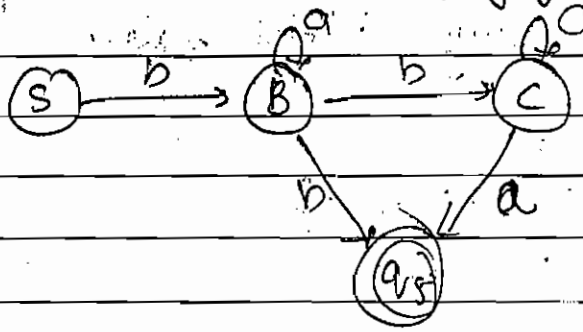
$B \to bC \mid aB \mid b$

$C \to aC \mid a$

DFA not possible

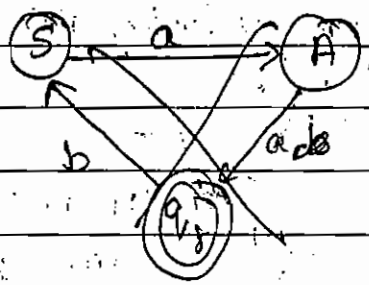


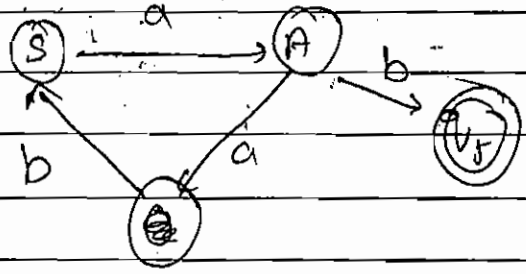
Note: while converting to NFA always take a new state. True only for Right linear G.



↳ $S \to aA$

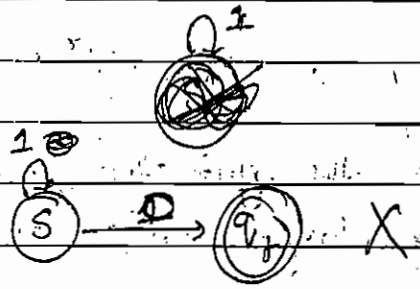
$A \to abS \mid b$





Date / /
Page

↳ $S \to S1 \mid 0$

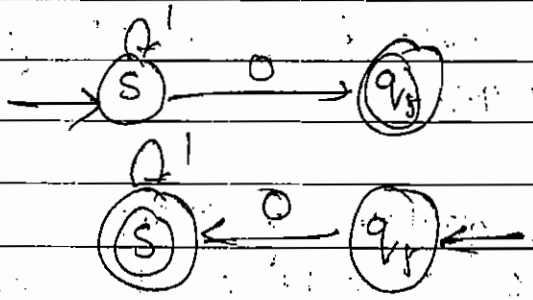


possible

Note: For Right Linear Gramm, 1st reverse the gramm, create FA, then reverse the FA.

ups take linear G.

$S \to 1S \mid 0$

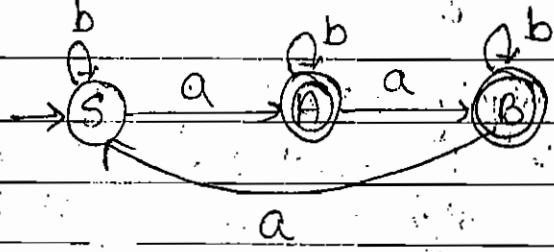


↳ Lang. genin by foll. grammer.

$S \to bS \mid aA \mid a$

$A \to bA \mid aB \mid b \mid a$

$B \to bB \mid aS \mid b$



Ans: no. of a's not divisible by 3.

Date 05/07/11
Page

Context Free Grammer :-

$$A \to \alpha$$
$$\alpha \in (V+T)^*$$

To show the syntactic rules for programming lang. while designing compiler.

⇒ Any grammer of the form
$$A \to \alpha$$
$$\alpha \in (V+T)^*$$
is known as CFG.

⇒ CFG is used to represent syntactic rules of P.L.

⇒ Ambiguity problem of CFG :-
A CFG is said to be ambiguous for atleast one string if there exist
↳ more than 1 Left Most Derivation
↳ ——— 1 RMD
↳ ——— 1 Parse Tree
↳ Only 1 L.M.D & only 1 R.M.D., both are producing diff. Parse Trees.

⇒ The Ambiguity problem of CFG is undecidable because there is no generalised algo to check ambiguity.

⇒ Elimination of ambuity of CFG is

also undecidable coz it is impossible to eliminate ambiguity from every ambiguous CFG.

⇒ Inherently ambiguous CFG :-
The CFG from which elimination of ambiguity is not possible.

2009
→ Find CFG that generates all strings of a's & b's where each string is odd length palindrome.

$S \rightarrow a\ S\ a \mid b\ S\ b \mid a \mid b$

→ Check following is ambiguous or unambiguous

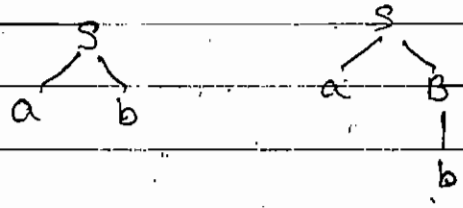
$S \rightarrow aB \mid ab$

$A \rightarrow aAB \mid a$

$B \rightarrow ABb \mid b$

$S \rightarrow aABb \mid aBb \mid ab$

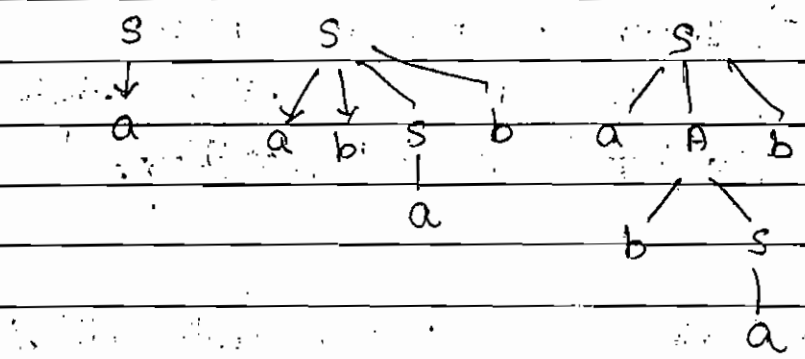
$S \rightarrow aaABBb \mid aaBb \mid aBb \mid ab$



Since there are two parse trees, it is ambiguous.

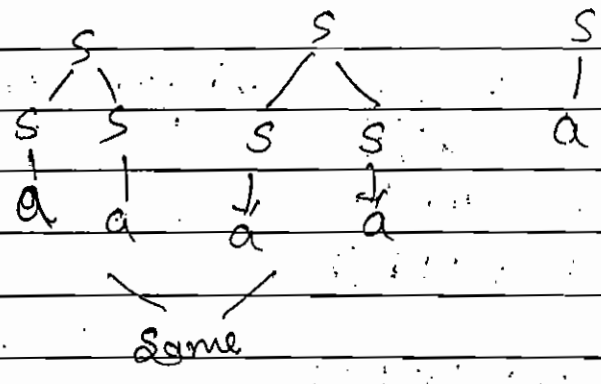
→ S → a / abSb / aAb

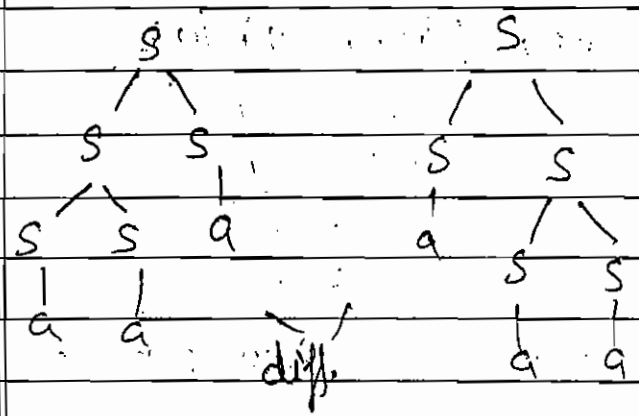
A → bS / aAAb



Ambiguous.

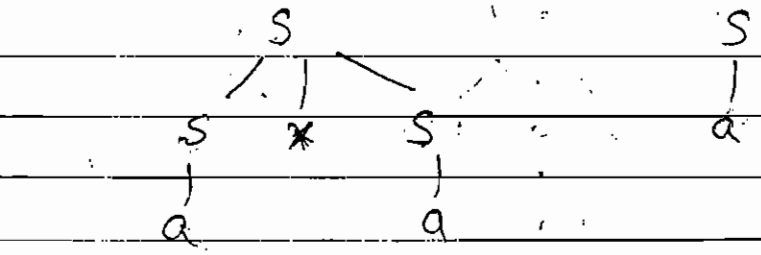
→ S → SS / a

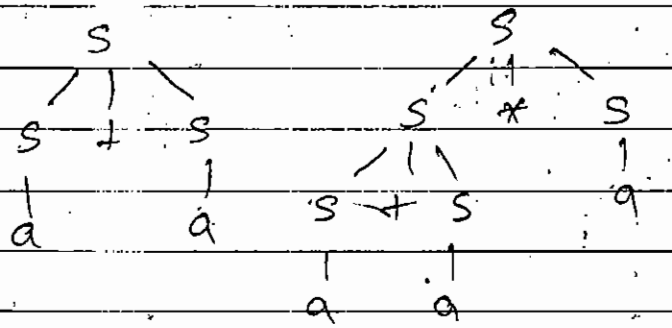


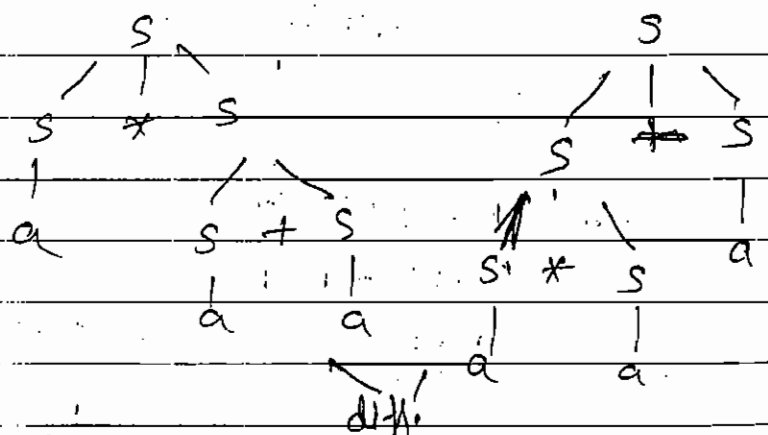


Ambiguous.

→ $S \rightarrow S * S \mid S + S \mid a$





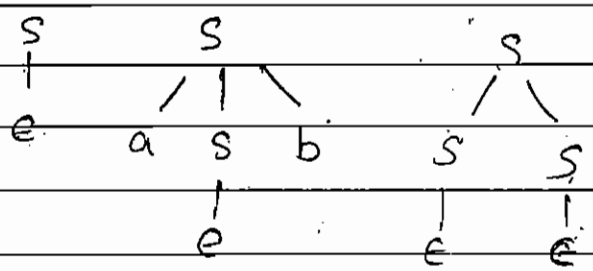


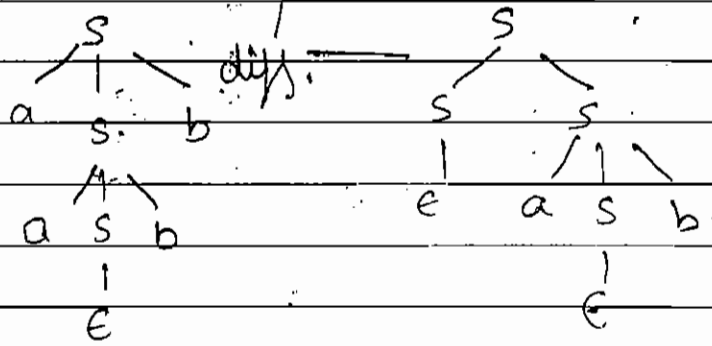
Check for $a * a + a$

↓

ambiguous.

→ $S \to aSb \mid SS \mid \epsilon$





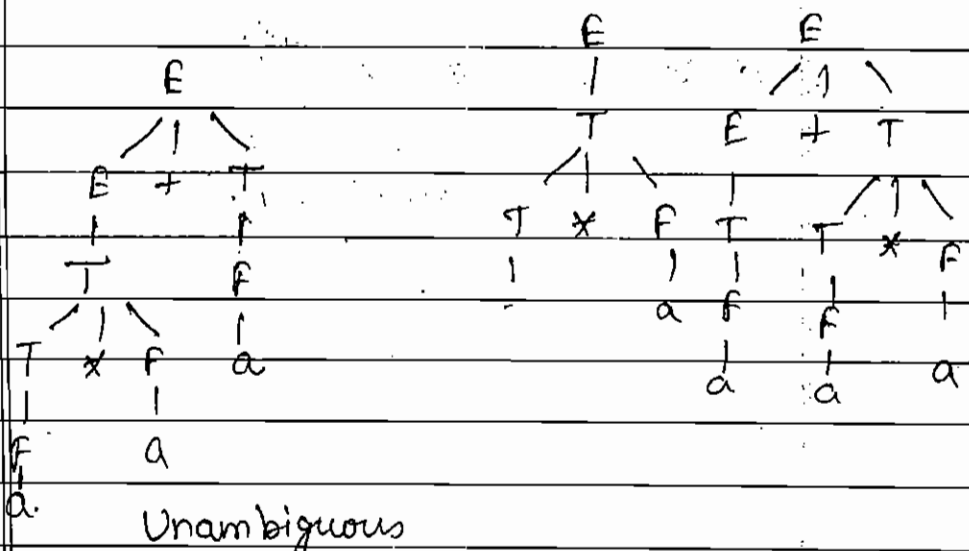
Ambiguous.

→ ~~Check~~ $E \to E+T \mid T$

$T \to T*F \mid F$

$F \to a$

$E+T*F$



Unambiguous

Date / /
Page

Simplification of CFG:

① Useless Symbol:

It means: variables which are not deriving any terminals and variable which are not necessary for derivation.

→ Eliminating useless symbol -

$S \to AB/a$

$A \to BC/b$

$XB \to aB/C$

$XC \to aC/B$

= Eliminate productions which are having B & C.

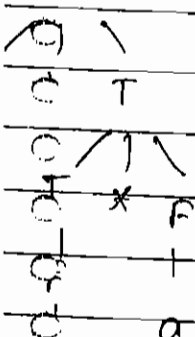
$S \to a$

$A \to b$

= Since A is useless:

$S \to a$.

Only 1 production.



→ $S \to \overset{AB}{\cancel{AB}}/\cancel{AC}$

$A \to aAb/bAa/a$

~~$B \to bbA/aaB/AB$~~ $B \to bbA/aaB/AB$

$XC \to abCa/aDb$

$XD \to bD/aC$

7 Productions.

Date / /
Page

→ S → ~~ABC~~ / BaB
~~A~~ → aA / ~~BAC~~ / aaa ✗ (useless)
~~B~~ → bBb / a
~~X~~ C → CA / AC

S → BaB
B → bBb / a

(2) Unit Productions :-

Any production of the form
A → B, is

⇒ 1st Eliminate Unit then Useless prod.

→ Eliminate Unit Production.

S → Aa / B
B → A / bb
A → a / bc / B

From Down to Top.
S → Aa
S → Aa / a / bc / bb
B → a / bc / bb
A → a / bc / bb

→ S → AB
A → a
B → C / b
C → D
D → E
E → a

(3)

Date / /
Page

Soln: $S \to AB$

$A \to a$

$B \to a/b$

$C \to a$

$D \to a$ } Useless

$E \to a$

$S \to AB$

$A \to a$

$B \to a/b$

③ Null Productions or ϵ-prod :-

Any prod. of the form

$A \to \epsilon$

The original & resultant grammer need not to be equivalent.

→ Eliminate ϵ-prod.

$S \to a S_1 b$

$S_1 \to a S_1 b / \epsilon$

Soln:- $S \to a S_1 b / ab$

$S_1 \to a S_1 b / ab$

→ $S \to AB$

$A \to aAA / \epsilon$

$B \to bBB / \epsilon$

$S \to AB/B/A$

$A \to aAA/aA/a$

$B \to bBB/bB/b$

Soln:-

$S \to AB/B/A/\epsilon$

$A \to aA/a$

$B \to bB/b$

Date / /
Page

⇒ Elimination Sequence —

Null prod. → Unit prod → Useless Symbols.

→ Simplify following CFG:

S → AbaC
A → BC
B → b / ϵ
C → D / ϵ
D → d

S → AbaC / Aba
A → BC / C / B / ϵ
B → b
C → D
D → d

S → AbaC / Aba / baC / ba
A → BC / C / B
B → b
C → D
D → d

(1)

(2)

S → Aba / Aba / baϵ / ba
A → Bϵ / d / B
B → b
C → d

Note:

Date / /
Page

$S \rightarrow AbaC / Aba / baC / ba$ 4

Symbols- $A \rightarrow d / b$ 3

$B \rightarrow b$ 1

$C \rightarrow d$ 1

$P \rightarrow q$

Normal Form of CFG:

Applying condition or restriction on R.H.S. of CFG, u

N.F.

Chomsky N.F

Greibach N.F.

(1) Chomsky N.F.:

Any CFG of the form

$A \rightarrow BC$

$A \rightarrow a$

(2) Greibach N.F.:

Any CFG of form

$A \rightarrow a\alpha$

$\alpha \in V^*$

Note: For the given ϵ-CFG only we can construct an equivalent CNF or GNF CFG.

Date / /
Page

→ Construct an equi. CNF - CFG for following -

$S \rightarrow bA \mid aB$

$A \rightarrow bAA \mid aS \mid a$

$B \rightarrow aBB \mid bS \mid b$

Sol^n:

$S \rightarrow bA$ ⇓

$S \rightarrow C_bA$ 1

$C_b \rightarrow b$ 2

$S \rightarrow aB$ ⇓

$S \rightarrow C_aB$ 3

$C_a \rightarrow a$ 4

$A \rightarrow bAA$ ⇓

$A \rightarrow C_bE$ 5

$E \rightarrow AA$ 6

$A \rightarrow aS$ ⇓

$A \rightarrow C_aS$ 7

$A \rightarrow a$ 8

$B \rightarrow aBB$ ⇓

$B \rightarrow C_aF$ 9

$F \rightarrow BB$ 10

$B \rightarrow bS$ ⇓

$B \rightarrow C_bS$ 11

$B \rightarrow b$ 12

Date / /
Page

→ Construct an equi. GNF - CFG for following —

$S \to AB$

$A \to aA \mid bB \mid b$

$B \to b$

Sol^n.

replace A by its Production

$S \to aAB \mid bBB \mid bB$

$A \to aA \mid bB \mid b$

$B \to b$

→ Construct equi GNF.

$S \to aSb \mid \epsilon$

$S \to aSb \mid ab$

$\Downarrow$

$S \to aSB \mid aB$

$B \to b$

} GNF.

Date / /
Page

Decision Properties of CFG's :

① **Emptiness Problem** —

Means checking whether the lang. generated by given CFG is empty or non-empty.

Algo :

① Check whether starting symbol derives any terminal or not.

② If S.S. derives atleast one terminal then the lang. generated by grammer is non-empty, otherwise empty.

→ $S \to XY$

$X \to AX / AA$

$A \to a$

$Y \to BY$

$B \to b$

It will produce empty lang.

→ $S \to XY$

$X \to AX / AA$

$A \to a$

$Y \to BY / BB$

$B \to b$ ↳ Non-empty.

→ $S \to XY$

$X \to AA / XY / b$

$A \to BC$

$B \to AC$

$C \to AB$

$Y \to aY / b$

↳ Non-empty

(2) Finiteness Problem :-

Algo:

1> Simplify CFG.

2> Construct transition graph by taking variables as vertices and production as directed edges.

3> In resultant T.G. if there exist any self loop or cycle then the lang. generated by that grammer is infinite lang. otherwise finite lang.

→ $S \to AB$

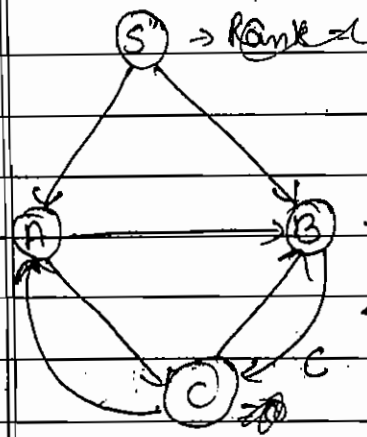
$A \to BC / a$

$B \to CC / b$

$C \to AB / a$

also find rank of variable S, A, B, C.

1> It is simplified.

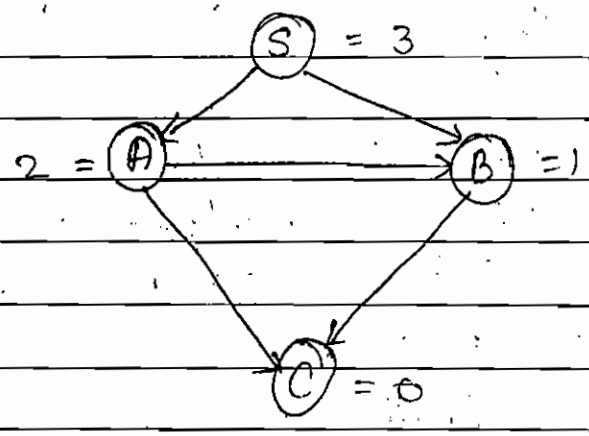
27. 

← cycle is present, so it is infinite.

Rank of Variables:

Length of the longest path starting from that variable in T.G.

→ S → AB
A → BC / a
B → CC / b
C → a



Date / /
Page

→ S → XY

X → AA / XY / b

XA → BC

X B → AC

X C → BA

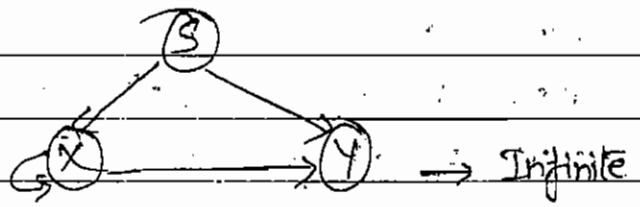
Y → a

↓

S → XY

X → XY / b

Y → a



③ Membership Problem :-

Means checking whether the given string x is valid member of given CFG or not.

CYK Algo :

C → Coocke

Y → Younger

K → Kasami

→ Dynamic Programming

→ Bottom Up Approach

→ $O(n^3)$

→ P-class

→ Chart parsing

Date / /
Page

→ (1) Check whether the string baaba is a valid member of following CFG —

$S \rightarrow AB \mid BC$

$A \rightarrow BA \mid a$

$B \rightarrow CC \mid b$

$C \rightarrow AB \mid a$

(2) Check whether the following strings are valid members of given grammer or not —

(1) b
(2) ba
(3) baa
(4) baab
(5) aab
(6) aba
(7) aaba
(8) aa

(3) How many substrings of given string baaba are valid members of given grammer? 5

(4) How many diff. substrings of given string are valid members of given grammer? 4

(5) How many substrings of the given string are not valid members. ? 10

(6) How many substrings of the given string are only generated from varible B.? 5

Date / /
Page

(7) How many diff. substrings of given string are only generated from variable: A & C?

Soln:

	b	a	a	b	a
	B	A, C	A, C	B	A, C
	A, S	B	S, C	A, S	
	∅	B	B		
	∅	S, A, C			
	A, S, C				

* 1st row filled with corresponding R.H.S. Non Terminals.
* 2nd row calculated by diagonal combinations.

Note: If in last block, starting symbol appears then it is member of given CFG otherwise not.

* For substring, check digits diagonally according to the length of substring.

(3) No. of boxes containing starting symbol is the no. of substring of given string, which are valid member.

Date / /
Page

(5) No. of boxes containing only B. So 10

(ii) diff. = 2

(6) 1

→ Check whether the string aa bbb is a valid member of foll. grammer or not —

$S \rightarrow AB$

$A \rightarrow BB/a$

$B \rightarrow AB/b$

2 → Check whether foll. string are member or not.

✗ 1) a ✗ 4) bb

✓ 2) ab ✗ 5) abb

✗ 3) aa ✗ 6) aabb

✓ 7) bbb

3 → How many diff. substring of given string are valid members ? 5

4 → How many diff substrings of given string are not valid mems ? 10

5 → How many diff. substring of given string are generated from only variable A ? 4

Sol^n:

a	a	b	b	b
A	A	B	B	B
∅	S, B	A	A	
S, B	A	S, B		
A	S, B			
S, B				

bb
member

① Yes, Valid member

2→ aab
→ ab
bbb
abbb
aabbb

All are diff. so all are different & valid.

number

4→ 10

5→ →a
a
bb
bb
abb
aabb

4

er. string

ven

ID

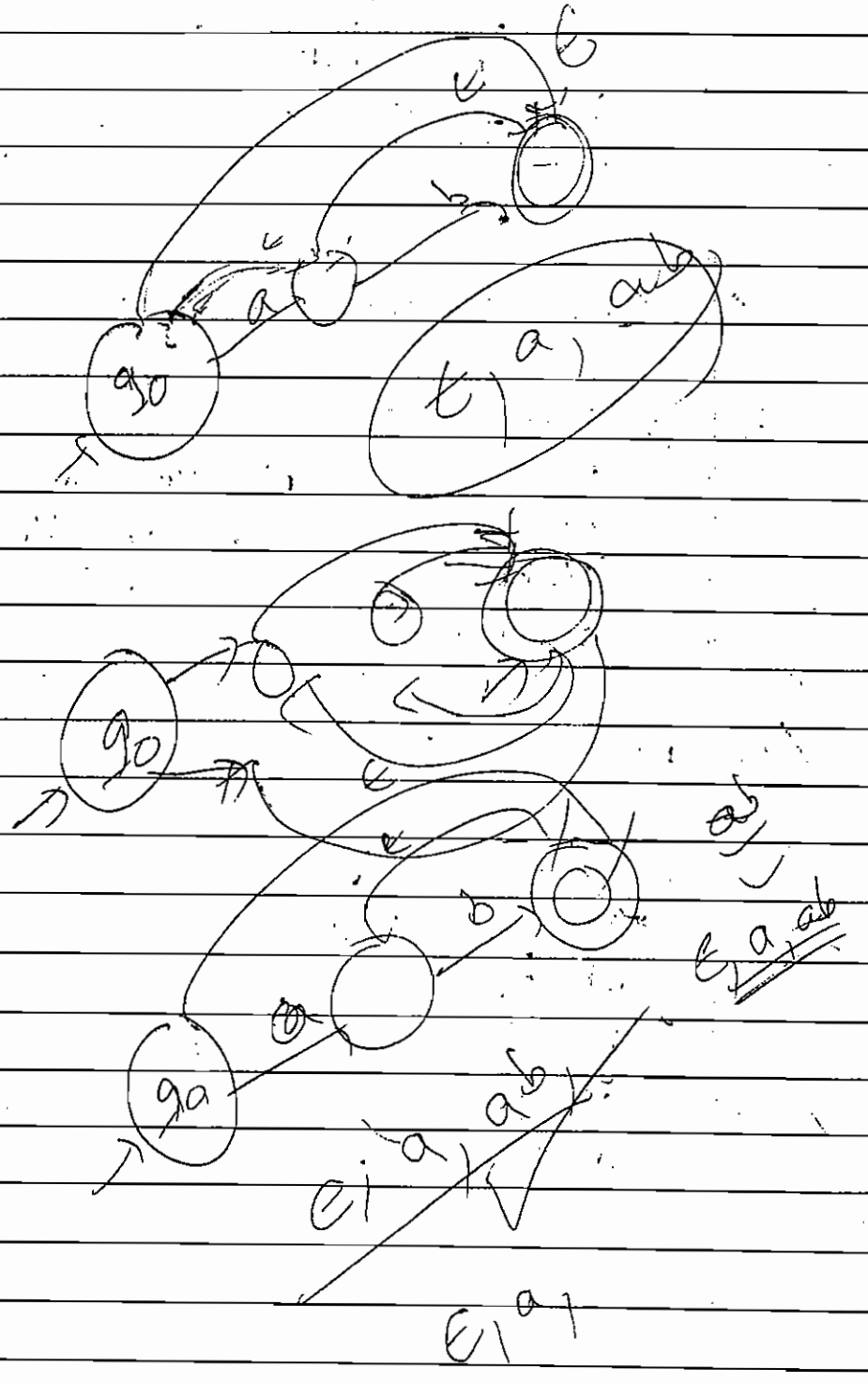
string

le A 9

4

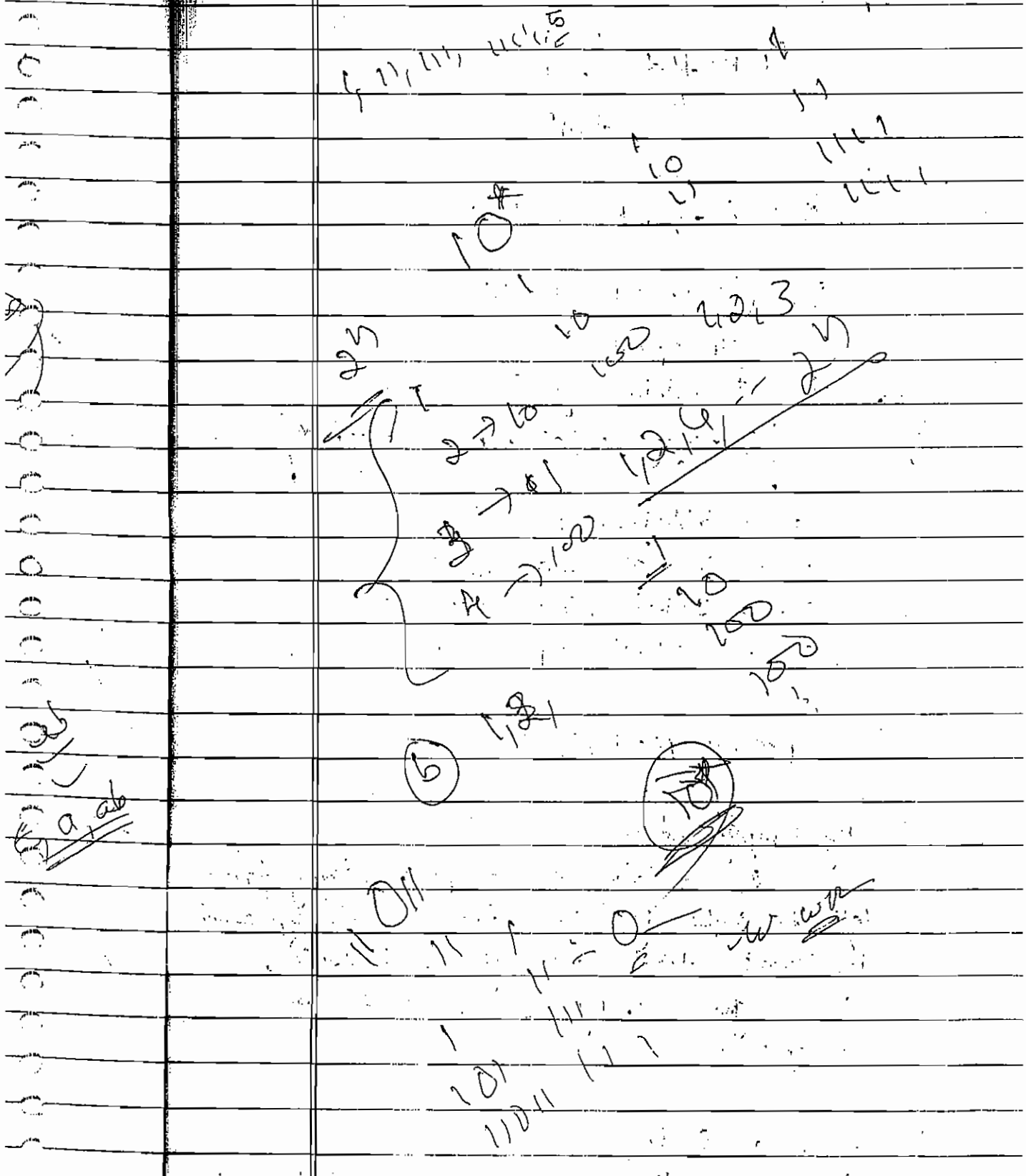
Date / /

Page



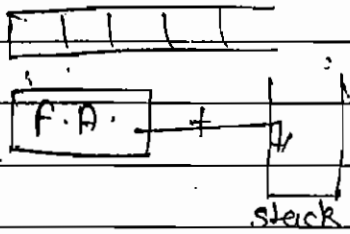
Date / /

Page



Date 05/08/2011
Page

Pushdown Automata :-



$(Q, \Sigma, \delta, q_0, F, Z_0, \Gamma)$

$Q \rightarrow$ finite set of states
$\Sigma \rightarrow$ i/p alphabet
$\delta \rightarrow$ transition fun^c.

$$\delta : Q \times \Sigma \cup \{\in\} \times \Gamma \rightarrow Q \times \Gamma^*$$

$q_0 \rightarrow$ initial state
$F \rightarrow$ Set of final states
$Z_0 \rightarrow$ Initial stack element
$\Gamma \rightarrow$ stack alphabet.

Acceptance Mechanism:

(1) By Empty stack

By reading the complete string from left to right if the stack is empty and irrelevant about no. of final states, then given string is said to be accepted.

(2) By final State

By reading string from left to right

if PDA enters into final state & irrelevent about no. of stack symbols, given string will be accepted.

Note: No. of lang. accepted by empty stack mechanism is equal to no. of lang. accepted by final states mechanism.

Notations

- Transition Diagram
- Transition Function.

Note: If we restrict the size of stack in PDA then the lang. accepted by that PDA is all R.L. (not only finite.)

⇒ PDA fails to recognize lang. where more than 1 memory element required.

→ Construct PDA for the lang.

$$L = \{a^n b^n / n \geq 1\}$$

$$\delta : Q \times \Sigma \cup \{\epsilon\} \times \Gamma \Rightarrow Q \times \Gamma^*$$

aabb

$\delta(q_0, a, z_0) = (q_0, az_0)$

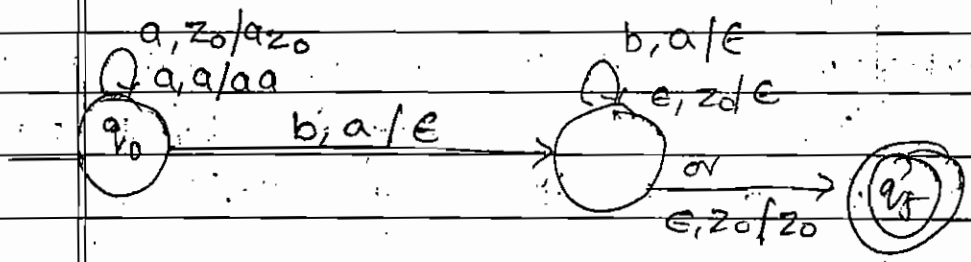
$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, \epsilon)$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, z_0) = (q_1, \epsilon)$ or (q_f, z_0)

by empty stack. by final state



→ Construct PDA that accept all string of a's & b's where each string contains equal no: of a's & equal no: b's.

L = {ab, ba, abab, aabb}

Logic!

Same symbols - push

diff. - pop.

$\delta(q_0, a, z_0) = (q_0, a z_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_0, \epsilon)$

$\delta(q_0, b, z_0) = (q_0, b z_0)$

$\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, b) = (q_0, \epsilon)$

$\delta(q_0, \epsilon, z_0) = (q_0, \epsilon)$

$\rightarrow L = \{a^n b^n c^m / n, m \geq 1\}$

$\delta(q_0, a, z_0) = (q_0, a z_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, \epsilon)$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_1, c, z_0) = \delta(q_2, z_0)$

$\delta(q_2, c, z_0) = (q_2, z_0)$

$\delta(q_2, \epsilon, z_0) = (q_2, \epsilon)$

Date / /
Page

→ $L = \{a^{m+n} b^n c^m / n, m \geq 1\}$

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, \epsilon)$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, a) = (q_2, \epsilon)$

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

$\delta(q_2, \epsilon, z_0) = (q_2, \epsilon)$

→ $L = \{a^m b^{m+n} c^n / m, n \geq 1\}$

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, \epsilon)$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_0, b, z_0) = (q_1, bz_0)$

$\delta(q_1, b, b) = (q_1, bb)$

$\delta(q_1, c, b) = (q_2, \epsilon)$

$\delta(q_2, c, b) = (q_2, \epsilon)$

$\delta(q_2, \epsilon, z_0) = (q_2, \epsilon)$

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

Not possible

→ Identify the lang. generated by following PDA.

$\delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, ba)$

$\delta(q_1, b, b) = (q_1, bb)$

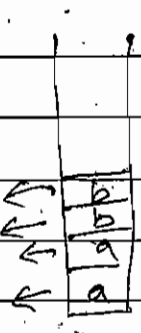
$\delta(q_1, \epsilon, b) = (q_2, \epsilon)$

$\delta(q_2, c, b) = (q_2, \epsilon)$

$\delta(q_2, d, a) = (q_3, \epsilon)$

$\delta(q_3, d, a) = (q_3, \epsilon)$

$\delta(q_2, \epsilon, z_0) = (q_2, \epsilon)$



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

→ Construct PDA -

$L = \{a^m b^n \mid m > n\}$

$\delta(q_0, a, z_0) = (q_0, a z_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, \epsilon)$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, a) = (q_2, \epsilon)$

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

$\delta(q_2, \epsilon, z_0) = (q_2, \epsilon)$

→ $L = \{a^n b^{2n} \mid n \geq 1\}$

$\delta(q_0, a, z_0) = (q_0, a z_0)$

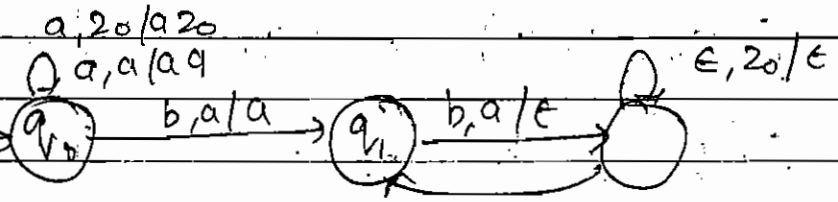
$\delta(q_0, a, a) = (q_0, aa)$

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

$\delta(q_1, b, a) = (q_2, \epsilon)$

$\delta(q_2, b, a) = (q_1, a)$

$\delta(q_2, \epsilon, z_0) = (q_2, \epsilon)$



b, a / a

→ $L = \{a^n b^{2^n} \mid n \geq 1\}$

Note ⇒ If symbols are not in A.P., then FA & PDA both are not possible.

→ Construct $L = \{wcw^R \mid w \in \{a, b\}^*\}$

abcba → palindrom

$\delta(q_0, a, z_0) = (q_0, a z_0)$

$\delta(q_0, b, z_0) = (q_0, b z_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, a, b) = (q_0, ab)$

$\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, c, b) = (q_\phi, b)$

$\delta(q_\phi, b, b) = (q_2, \epsilon)$

$\delta(q_2, b, b) = (q_2, \epsilon)$

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

$\delta(q_2, \epsilon, z_0) = (q_2, \epsilon)$

Date / /
Page

$\delta(q_0, a, z_0) = (q_0, a z_0)$

$\delta(q_0, b, z_0) = (q_0, b z_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, a, b) = (q_0, ab)$

$\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, c, z_0) = (q_1, z_0)$

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

$\delta(q_0, c, b) = (q_1, b)$

$\delta(q_1, a, a) = (q_1, \epsilon)$

$\delta(q_1, b, b) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, z_0) = (q_1, \epsilon)$

Date / /

Page

→ $L = \{ WW^R \mid W \in \{a, b\}^* \}$

$\delta(q_0, \epsilon, z_0) = (q_0, \epsilon)$

$\delta(q_0, a, z_0) = (q_0, a z_0)$

$\delta(q_0, b, z_0) = (q_0, b z_0)$

$\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$

$\delta(q_0, a, a) = (q_0, aa)$ or (q_1, ϵ)

$\delta(q_0, b, b) = (q_0, bb)$ or (q_1, ϵ)

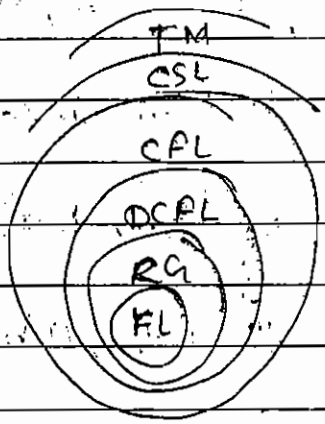
$\delta(q_1, a, a) = (q_1, \epsilon)$

$\delta(q_1, b, b) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, z_0) = (q_1, \epsilon)$

Date / /
Page

⇒ All DCFLs are CFL but all CFL need not be DCFL. Hence the no. of lang. accepted by NPDA is more than DPDA.



→ Which of the following is CFL but not DCFL.

a) $L = \{ w c w^R \mid w \in \{0,1\}^* \}$

b) $L = \{ 0^n 1^n 2^n \mid n \geq 1 \}$

c) $L = \{ 0^n 1^m 2^{n+m} \mid n, m \leq 10 \}$ → finite lang.

d) None.

→ $L = \{ wcw \mid w \in (a,b)^* \}$ PDA not possible
abcab aba caba bba cbba

$\delta(q_0, c, z_0) = (q_0, \epsilon)$

$\delta(q_0, a, z_0) = (q_0, a)$

$\delta(q_0, b, z_0) = (q_0, b)$

Date / /
Page

need
cepted

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, a, b) = (q_0, ab)$

$\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, c, z_0) = (q_f, \$ z_0)$

$\delta(q_0, c, a) = (q_f, \$ a)$

$\delta(q_0, c, b) = (q_f, \$ b)$

t. not

→ $L = \{w w \mid w \in (a,b)^*\}$

PDA not possible

inite
ang.

→ Identify the lang. accepted by following PDA.

$\delta(q_0, a, z_0) = (q_1, z_0)$

$\delta(q_0, b, z_0) = (q_0, z_0)$

t possible

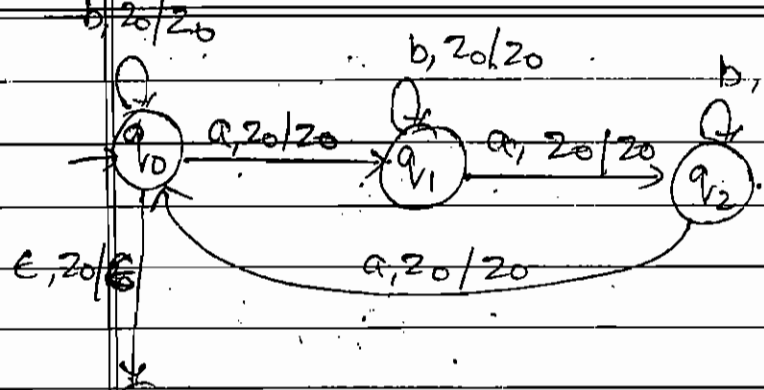
$\delta(q_1, a, z_0) = (q_2, z_0)$

$\delta(q_1, b, z_0) = (q_1, z_0)$

$\delta(q_2, a, z_0) = (q_0, z_0)$

$\delta(q_2, b, z_0) = (q_2, z_0)$

$\delta(q_0, \epsilon, z_0) = (q_f, \epsilon)$



Ans:- Accepting R.L.

$a^* b^*$

Context Free Languages:

→ Which of the following lang. are
(a) CFL (b) non-CFL, (c) CFL but not regular
& (d) non CFL, non regular.

d)

① $L = \{a^n b^n c^m \mid n > m\}$

a) ② $L = \{a^n b^n c^n d^n \mid n \leq 1\}$

c ③ $L = \{a^n b^m \mid n < m\}$

d ④ $L = \{a^n b^n c^n d^n \mid n \geq 1\}$

c ⑤ $L = \{a^n b^m c^m d^n \mid m, n \geq 1\}$

a ⑥ $L = \{a^n b^m c^n d^m \mid n, m \leq 1\}$

c ⑦ $L = \{0^n 1^m 2^{n+m} \mid m, n \geq 1\}$

$a^n b^n c^n \cdots b^n a^m \quad n \geq m$

Date / /

Page

8) $L = \{0^n 1^{2n} 2^{3n} \mid n \geq 1\}$

9) $L = \{x c y \mid x, y \in \{0,1\}^*\}$

10) $L = \{x x^R \mid x \in \{a,b\}^* \text{ and } |x| = 1\}$

11) $L = \{x c x \mid x \in \{0,1\}^*\}$

12) $L = \{www w^R \mid w \in \{a,b\}^*\}$

13) $L = \{a^n b^{3^n} \mid n \geq 1\}$

14) $L = \{a^m b^n \mid m \neq n\}$

15) $L = \{a^m b^n \mid m = 2n+1\}$

16) $L = \{a^i b^j \mid i \neq 2j+1\}$

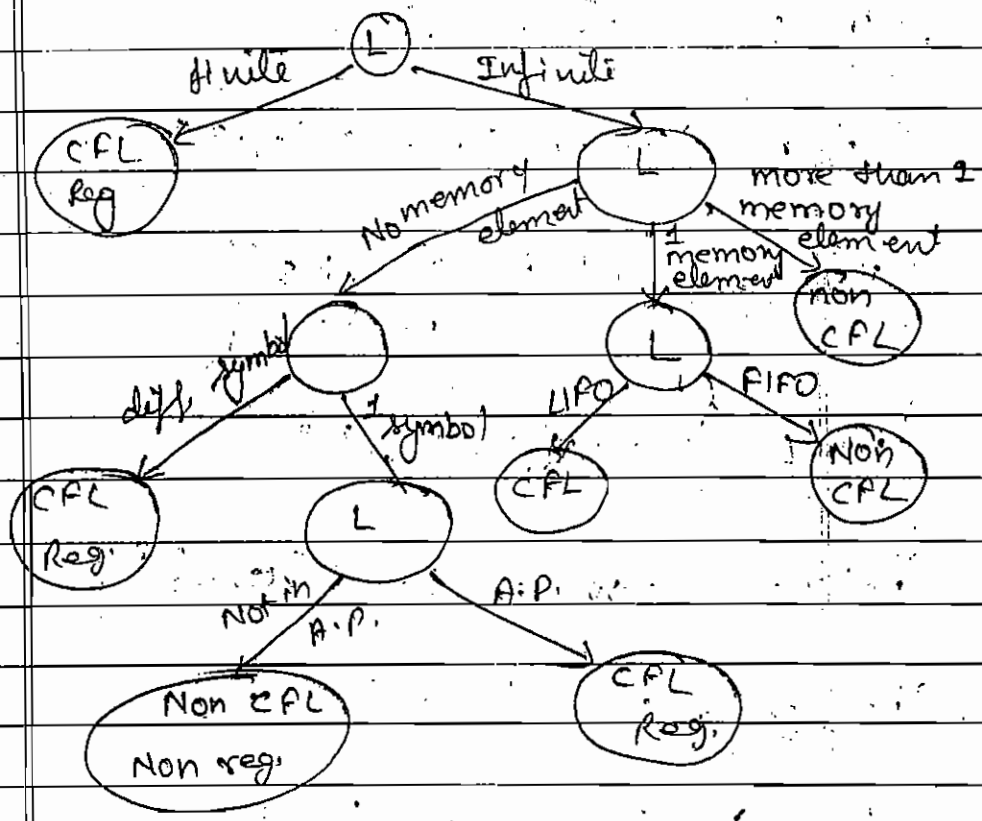
17) $L = \{a^{n^2} \mid n \geq 1\}$

18) $L = \{a^{2n} \mid n \geq 1\}$

19) $L = \{b^{n!} \mid n \geq 1\}$

20) $L = \{a^m \mid m \text{ is a prime no.}\}$

21) $L = \{a^k \mid k \text{ is even no.}\}$



d (22) $L = \{a^i b^j k^k \mid i > j > k\}$

c (23) $L = \{a^i b^j c^k \mid j = i + k\}$

d c (24) $L = \{a^i b^j c^k d^l \mid i = k \text{ or } j = l\}$

c (25) $L = \{a^i b^j c^k d^l \mid i = j \text{ and } k = l\}$

d (26) $L = \{a^i b^j c^k d^l \mid i = k \ \& \ j = l\}$

c (27) $L = \{a^i b^j c^k d^l \mid i = l \ \& \ j = k\}$

a (28) $L = \{a^m b^l c^k d^n \mid m, l, k, n \geq 1\}$

a (29)

a (30)

a (31)

d (32)

a (33)

d (34)

c (35)

a (36)

Date / /
Page

(a) (29) $L = \{a^n b^{4m} \mid n, m \geq 1\}$

(a) (30) $L = \{a^3, a^8, a^{13}, - - -\}$

mory element

(a) (31) $L = \{ , ^{2n+1} \mid n \geq 1\}$

CFL

(d) (32) $L = \{0^n \mid n \geq 1\}$

Non CFL

(a) (33) $L = \{w \mid w \in \{a, b, c\}^* \; |w| \geq 100\}$

(d) (34) $L = \{w \mid w \in \{a, b, c\}^* \; n_a(w) = n_b(w) = n_c(w)\}$

(c) (35) $L = \{w \mid w \in \{a, b, c\}^* \; \& \; n_a(w) \geq n_b(w) + 1\}$

(a) (36) $L = a^* b^* c^* d^*$

Date / /
Page

Context Free → PDA

Let GNF

$A \to a\alpha \mid b\gamma$

where $\alpha, \gamma \in V^*$

then we can construct PDA as follows steps:

① Let push the starting symbol Let A.

$\delta(q_0, \epsilon, Z_0) = (q_1, A Z_0)$

② Push R.H.S. of A.

$\delta(q_1, a, A) = (q_1, \alpha)$

$\delta(q_1, b, A) = (q_1, \gamma)$

③ Final State

$\delta(q_1, \epsilon, Z_0) = (q_f, \epsilon)$

→ Construct equivalent PDA for following CFG:

$S \to aSb \mid ab$

$\delta(q_0, \epsilon, Z_0) = (q_1, S Z_0)$

$\delta(q_1, a$

~~$S \to aSB \mid ab$~~ $S \to aSB \mid aB$

~~$B \to$~~ $B \to b$

① $\delta(q_0, \epsilon, Z_0) \to (q_1, SZ_0)$

Date / /
Page

② $\delta(q_1, a, S) = (q_1, SB)$ or (q_1, B)

$\delta(q_1, b, B) = (q_1, \epsilon)$

follows

③ $\delta(q_1, \epsilon, Z_0) = (q_f, \epsilon)$

A.

Note:- Grammer to PDA is ~~by~~ by default NPDA.

→ $S \to aAA$

$A \to aS/bS/a$

① $\delta(q_0, \epsilon, Z_0) \to (q_1, SZ_0)$

② $\delta(q_1, a, S) \to (q_1, AA)$

$\delta(q_1, a, A) \to (q_1, S)$ or (q_1, ϵ)

$\delta(q_1, b, A) \to (q_1, S)$

eg CFG:

~~$\delta(q_1$~~

③ $\delta(q_1, \epsilon, Z_0) = (q_f, \epsilon)$

→ $S \to 0A$

$A \to 0AB/1$

$B \to 1$

① $\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$

Date / /
Page

② $\delta(q_1, 0, S) = (q_1, A)$

$\delta(q_1, 0, A) = (q_1, AB)$

$\delta(q_1, 1, A) = (q_1, \epsilon)$

$\delta(q_1, 1, B) = (q_1, \epsilon)$

③ $\delta(q_1, \epsilon, Z_0) = (q_f, \epsilon)$

Closure Properties of DCFL & CFL:

① Union Op:-

$L_1 \cup L_2$

Let L_1 & L_2 are any two CFL and S_1 & S_2 are CFGs for L_1, L_2 respectively. Then '$L_1 \cup L_2$' is

$S \rightarrow S_1 \mid S_2$

We can construct the PDA for this hence $L_1 \cup L_2$ is is always CFL. Hence CFLs are closed under union op.

Note

* DCFL are not closed under Union. coz we can't construct DPDA from union grammer.

② Concatenation:

$S \rightarrow S_1 S_2$

We can construct PDA for this hence $L_1 L_2$ are always CFL. So, CFLs are closed under Union Concatenation

Date / /
Page

* DCFLs are not closed under Concatenation op.

(3) Kleene Closure:

$L_1 \rightarrow CFL \rightarrow S_1$

$L_1^* \rightarrow S_1 S / \epsilon$

~~Kleene~~ CFLs are closed under Kleene closure, but DCFLs are not closed.

(4) Positive Closure:

$L_1 \rightarrow CFL \rightarrow S_1$

$L_1^+ \rightarrow S_1 S / S_1$

CFLs are closed under Positive closure but DCFLs are not closed.

(5) Intersection Closure:

Let L_1 & L_2 are any two CFLs and Intersection of L_1 & L_2 may or may not context free hence CFL are not closed under intersection op.

Note: Intersection of two DCFLs may or may not DCFL hence DCFLs are not closed under intersection opr.

(6) Intersection with R.L.:

Let L_1 be CFL & L_2 be R.L., then $L_1 \cap L_2$ is always CFL, need not be regular. Hence, CFL are closed under intersection with

Date / /
Page

R.L. operation. Hence D

Note: Intersection of DCFL & R.L. is always DCFL & need not be R.L. hence DCFL are closed under Intersection with R.L. op.

(7) Complement Op:- (10)

Let L_1 & L_2 are two DFLs. Assume that L_1 & L_2 are closed under Complement, So

$\overline{L_1} \cup \overline{L_2} = CFL$ $\quad \overline{L_1} \to \Sigma^* - L_1$

$\overline{L_1 \cup \overline{L_2}} \to CFL$

$L_1 \cap L_2 \to NCFL$

So, ~~CFL~~ CFLs are not closed under Complement Op.

Note: Complement of DCFL is always DCFL hence DCFLs are closed under complement op. (Complemented DPDA is possible.)

(8) Diff. Op:-

L_1 (CFL) $\quad L_2$ (CFL)

$L_1 - L_2 = L_1 \cap \overline{L_2} \to NCFL$

NCFL

$L_1 - L_2 = NCFL$

So, CFLs are not closed under Diff. op.

Note: DCFLs are not closed under Diff. op.

(9) Diff. with Reg. Lang.:-

L_1 (CFL) $\quad L_2$ (Reg.)

(1) Uni
(2) Inte
(3) Kree
(4) Posi
(5) ~~Int~~
(6) Inte
(7) Comple
(8) Diff.
(9) Diff.
(10) Reve

$L_1 - L_2 = L_1 \cap \overline{L_2} \rightarrow$ CFL

Reg.

Hence, CFL & DCFL are closed under Diff. with R.L.

(10) Reverse Op:-

$L_1 \rightarrow$ CFL $\{a^n b^n \mid n \geq 1\}$

$S \rightarrow aSb/ab \rightarrow$ PDA

$L_1^R \rightarrow \{b^n a^n \mid n \geq 1\}$

$S \rightarrow bSa/ba \rightarrow$ PDA

CFLs are closed under Reverse Op.,
DCFLs are not closed under Reverse Op.

	CFL	DCFL
(1) Union	✓	X
(2) Intersection	X	X
(3) Kleen	✓	X
(4) Positive	✓	X
(5) Concatenation	✓	X
(6) Intersection with R.L.	✓	✓
(7) Complement	X	✓
(8) Diff.	X	X
(9) Diff. with R.L.	✓	✓
(10) Reversal	✓	X

Date / /
Page

→ Let L_1 & L_2 are two CFLs, L_3 & L_4 are two RLs then which of following is CFL or ~~not~~ may or may not CFL —

A) CFL
B) may or may not CFL

(1) $L_1 \cup L_2$ → CFL
(2) $L_3 \cup L_4$ → CFL
(3) $L_1 \cup L_2 \cup L_3 \cup L_4$ → CFL
(4) $(L_1 \cup L_2) \cap (L_3 \cap L_4)$ → CFL
(5) $(L_1 \cap L_3) \cup (L_2 \cap L_4)$ → ~~NCFL~~ A
(6) $(L_1 \cap L_4) \cap (L_2 \cap L_3)$ → B
(7) $(L_1 \cap L_2) \cap (L_3 \cup L_4)$ → B
(8) $(L_1 \cap L_3) \cup (\overline{L_2} \cup L_4)$ → B
(9) $(L_2 - \overline{L_4})$ → A
(10) $(\overline{L_3} - \overline{L_4}) \cap (L_2 - L_3)$ → A
(11) $(\overline{L_2} - L_4) - (\overline{L_3} - \overline{L_4})$ → ~~NCFL~~ A
(12) $(L_1 - L_2) - (L_3 - \overline{L_4})$ → B

2009

→ Let $L = L_1 \cap L_2$
where $L_1 = \{a^n b^n c\, a^m b^m \mid n, m \geq 0\}$ → CFL
$L_2 = \{a^i b^j c^k \mid i, j, k \geq 1\}$ → ~~NCFL~~ Reg.
then L is

a) CFL but not R.L. b) CFL & R.L.
c) CFL but not CSL d) Not Recursive Enum

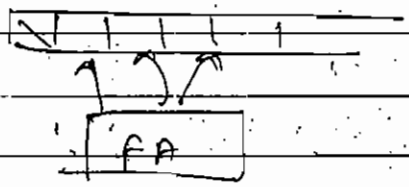
$L_1 \cap L_2 = a^n b^n c$.

Date / /
Page

→ Consider two lang. P & Q. where P is R.L. & Q is CFL. then which of the following is always R.L. –

a) $P \cap Q$ b) $Q - P$ c) $\Sigma^* - P$ d) $\Sigma^* - Q$

TURING M/C :-



① Infinite
② Read & write

$T.M = (Q, \Sigma, \delta, q_0, F, B, \Gamma)$

δ : transition func.

$Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

B: Blank Symbol

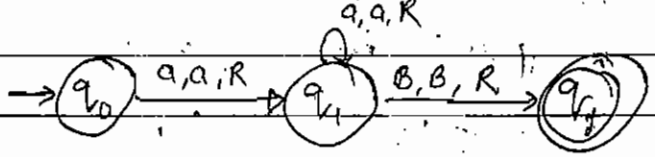
Γ : & Tape alphabet

Notation → T.D? T.Table

T M types.
→ L.R. O/P generator Lang. generator

→ CFL → Construct TM for lang.

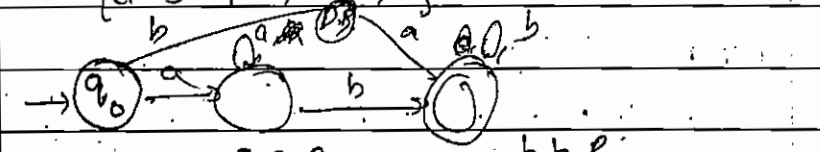
$L = \{a^n / n \geq 1\}$

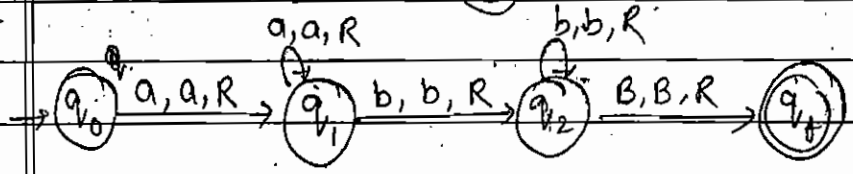


	a	B
q_0	(q_1, a, R)	
q_1	(q_1, a, R)	(q_f, B, R)
q_f	HALT	

Date / /
Page

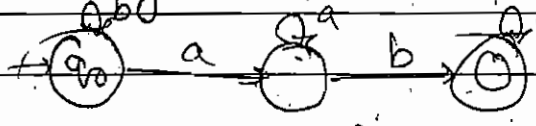
→ $L = \{a^n b^m \mid n, m \geq 1\}$

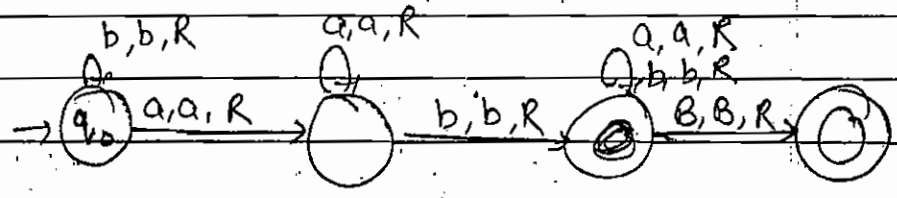




Note: If any string goes to halt state, not acceptable by TM otherwise acceptable.

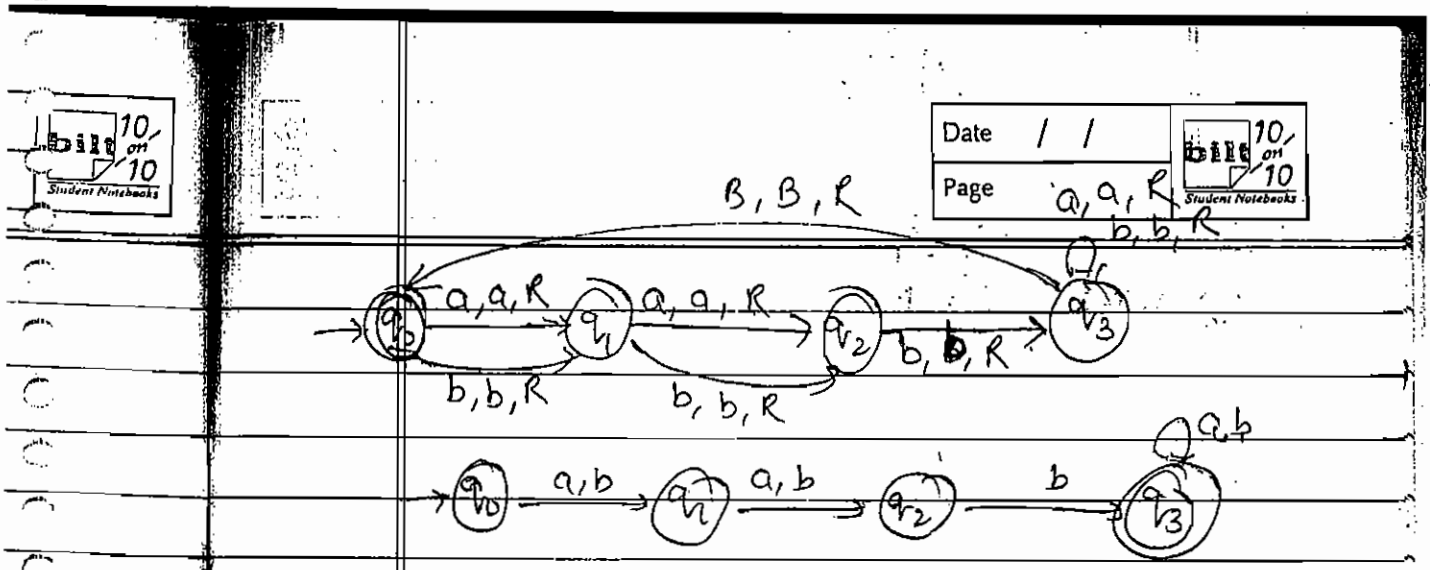
→ TM that accepts all strings of a's & b's where each string contain ab as substring.



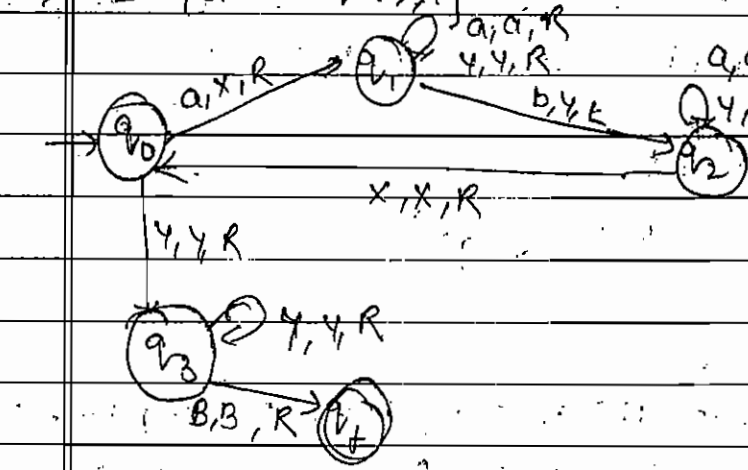


→ Identify lang. determined by following TM –

	a	b	B	B
q_0	(q_1, a, R)	(q_1, b, R)	HALT	
q_1	(q_2, a, R)	(q_2, b, R)	HALT	
q_2	HALT	(q_3, B, R)	HALT	
q_3	(q_3, a, R)	(q_3, b, R)	(q_1, B, R)	
q_T	HALT			



→ $L = \{a^n b^n \mid n \geq 1\}$ aaabbbB

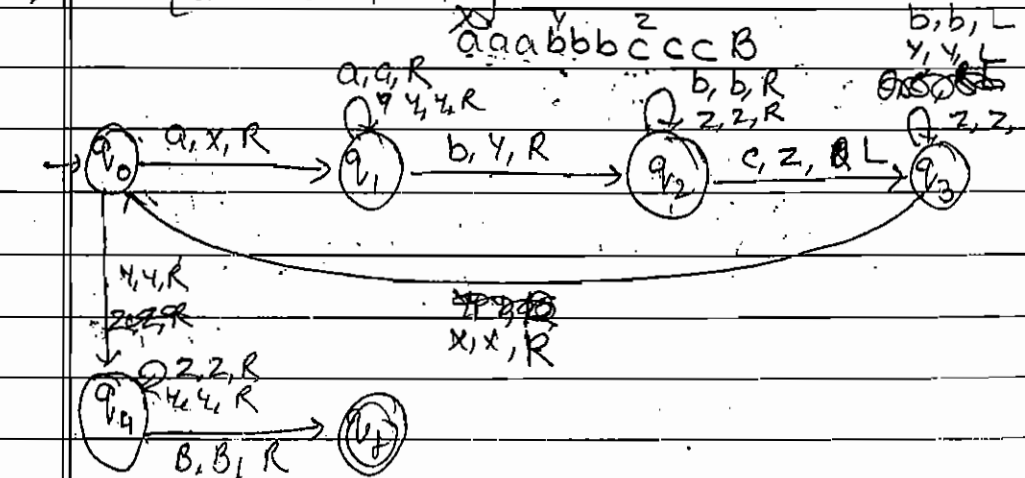


acceptable

where

→ $L = \{a^n b^n c^n \mid n \geq 1\}$

aaabbbccc B



TM -

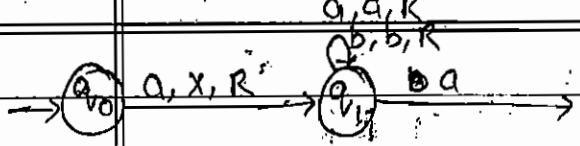
→ TM that accepts 0's & 1's where each string is even length palindrom.

$L = \{ w w^R \mid \ldots \}$

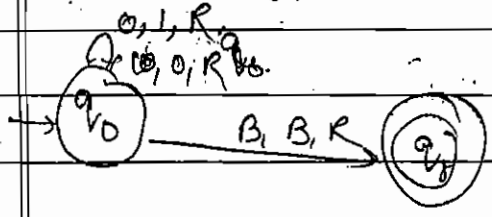
a b a a b a

X X B B

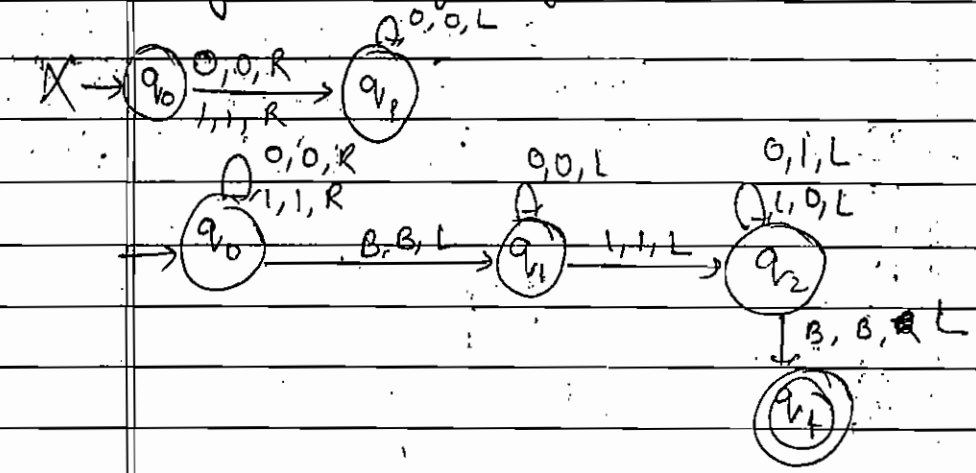
abaaba
R K B B



→ Construct TM that produces 1's complement of given binary string as O/P.



→ Construct TM that produces 2's complement of binary string as O/P (Assume that we are using two ways infinite tape TM.)



Lang. Generated Category of TM:

String of lang. generated by TM are effectively enumerated in either alphabetical order or canonical order, then that lang. is known as Recursive lang.

Date / /
Page

& Lang. Generated

Rec. Lang. ⊂ R.E.L.

R.E.L:

If string of lang. is enumerated in either alphabetical or canonical or in any order, then that lang. is R.E.L.

All Recursive lang. are R.E.L but all R.E.L need not be Recursive.

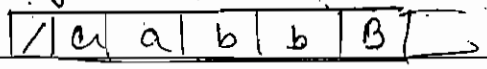
2003 →

If string of lang. produced by T.M. are effectively enumerated in lexico graphic order then that lang. is known as. - ~~Recursive lang.~~

① R.E.L but not Recursive
② Not Recursive & not R.E.L.
③ Recursive & R.E.L.
④ NOT

Lexico Graphic → Alphabetical order

Modifications of T.M.:-



① 2-way infinite tape TM
② Multitape
③ Two Type TM
④ NTM
⑤ Offline TM

After applying modification to T.M, expressive power of it remains same.

Note: For every NTM of polynomial time complexity, we can construct an equivalent DTM of exponential time complexity.

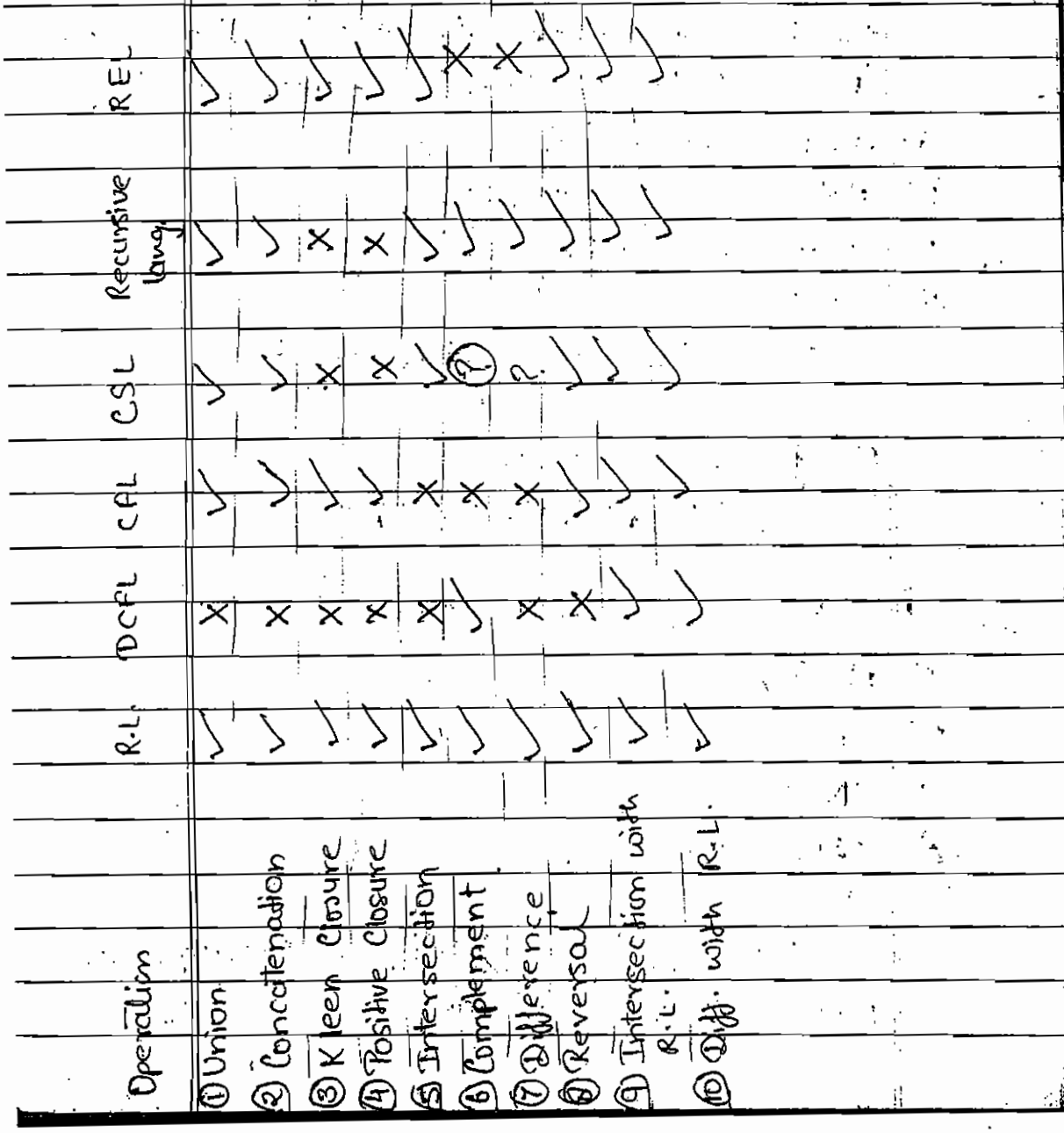
Date / Page

Recursive & R.E.L:

↳ By reading every string S in lang., TM always Halts, the lang. accepted by such type of TM are known as recursive R Lang.

↳ By reading string S in lang., TM may Halt or may enter into ∞ loop, lang. recognized by such TM are known as REL.

↳ All Recursive lang. are REL but all REL need not be recursive.



4-1, we

Similarly, if A, B & C are disjoint.

$P(A \cup B \cup C) = P(A) + P(B) + P(C)$

Undecidability

Problem	REL	Recursive	CSL	CFL	DCFL	RL
a) Is $L = \emptyset$?	U.D	U.D	U.D	D	D	D
b) Is $L = \Sigma^*$? (complement closed, emptiness)	U.D	U.D	U.D	U.D	U.D	D
c) Is $L_1 = L_2$?	U.D	U.D	U.D	U.D	U.D	D
d) Is $L_1 \subseteq L_2$? ($L_1 \cap L_2$ & equi)	U.D	U.D	U.D	U.D	U.D	D
e) $L_1 \cap L_2 = \emptyset$?	U.D	U.D	U.D	U.D	U.D	D
f) If L finite or not?	U.D	U.D	U.D	D	D	D
g) Complement of L is L? Same type?	U.D	U.D	U.D	U.D	D	D

always are recursively enumerated

REL

Collectively Exhaustive Events:

A set of Events of S is said to be exhaustive if it includes all possible outcomes of R.E.

→ Consider

$A = \{1, 3, 5\}$, $B = \{2, 4, 6\}$ of $S = \{1, 2, 3, 4, 5, 6\}$ in throwing single dice, then we can observe that two events A & B mutually Exclusive Collective Exha & equally likely events

Addition theorem of Probability :- Note!

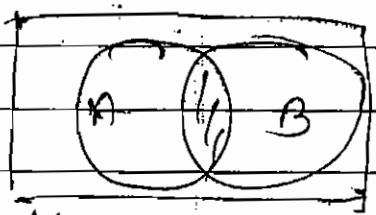
Let A & B any two events of S of R.E. then probability of occ. of atleast one of two event A, B is given by

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Proof :- From fig.

$$n(A \cup B) = n(A) + n(B) - n(A \cap B)$$

$$n(S) \qquad n(S)$$



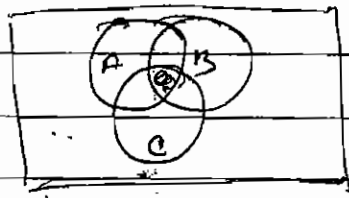
Note!

Therefore $P(A \cup B)$ gives probability of occurence of event A or B or Both A & B.

Note: Similarly $P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$

Therefore $P(A \cup B \cup C)$ gives probability of occ. of A or B or C or any two of them or all three of them.

i.e. occ. of atleast 1 of them.



Note: Suppose A & B are disjoint then

$$P(A \cup B) = P(A) + P(B)$$

Complement of Event:-

Let $E \subseteq S$, be any event of sample space S of random Exp, then E' or $\bar{E}$ or $\bar{E}$

$$P(\bar{E}) = \frac{n(\bar{E})}{n(S)} = \frac{n(S) - n(E)}{n(S)}$$

$$\boxed{P(\bar{E}) = 1 - P(E)}$$

gives probability of non occurrence of event E.

Therefore $P(E) + P(\bar{E}) = 1$

Note: $\frac{n(E)}{n(E')}$ → odds in Favorable of 'E'.

$\frac{n(E')}{n(E)}$ → odds against to 'E'.

Note: If it is given either of them, we can calculate $P(E)$ & $P(E')$.

$$P(E) = \frac{n(E)}{n(S)} = \frac{n(E)}{n(E) + n(E')}$$

$$P(E') = \frac{n(E')}{n(S)} = \frac{n(E')}{n(E) + n(E')}$$

Equally likely Events:

Two or more events of a sample space are said to be Equally likely Events, if there is no reason to believe that one particulare event occurs more than others.

Mutually Exclusive Events:

Two or more events of S are said to be Mutually Exclusive or Disjoint events if one of them occurs then the others will not occur or no two of them can occure together.

Probability of Occurance of an Event :-

Let E subset of S, be any event of sample space S of R.E, the probability of occurence of E

$$P(E) = \frac{n(E)}{n(S)} = \frac{\text{no. of favorable cases of } E}{\text{total no. of cases in } S}$$

→ Consider $S = \{1, 2, 3, 4, 5, 6\}$, then $A = \{1, 3, 5\}$, $B = \{2, 3, 5\}$, $C = \{5, 6\}$ etc. are events then

$$P(A) = \frac{n(A)}{n(S)} = \frac{3}{6} = \frac{1}{2} = 50\%$$

$$P(B) = \frac{n(B)}{n(S)} = \frac{3}{6} = \frac{1}{2} = 50\%$$

$$P(C) = \frac{n(C)}{n(S)} = \frac{2}{6} = \frac{1}{3} = 33\%$$

Note: We know that empty set is subset to any set, i.e. $\emptyset \subseteq S$, then

$$P(\emptyset) = \frac{n(\emptyset)}{n(S)} = \frac{0}{n(S)} = 0 = 0\%$$

hence $\emptyset$ is called impossible event.

Note: We know that every set is subset to itself i.e. $S \subseteq S$

$$P(S) = \frac{n(S)}{n(S)} = \frac{1}{1} = 1 = 100\%$$

hence, the sample space S itself is called certain or sure event.

Note: We know that $\emptyset \subseteq E \subseteq S$

$$n(\emptyset) < n(E) < n(S)$$

$$0 < P(E) < 1$$

Hence, probability of occurance of any event lies b/w 0 & 1 (0% – 100%).

r no. of objects are of 3rd category & remain objects are independently diff. Then total no. of arrangements

$$\frac{n!}{p!\,q!\,r!}$$

G) $\left({}^{7}C_4 \times {}^{5}C_3\right) \times 7!$

↓ selection ↓ arrangement.

Probability:

Experiment: It is an operation, in which the result or outcome can predicted with certainity.

Random Exp: It is an experiment, in which outcome cannot be predicted with certainity.

* If a stone is thrown upwards, definedly it will come down (Experimental).

* If a dice thrown once, we may get 1, 2 -- 6. (Random Experiment).

* Drawing a card from pack of cards (R. Experiment)

Sample Space:-

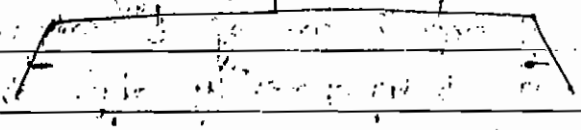
Set of all possible outcomes of random experiment. It is denoted by S.

Event (E):-

Any subset of sample space of random experiment.

Date / /
Page

$^3C_1 \times {}^2C_1 \times {}^1C_1$



4 people. 3!

(i) $(7-1)! = 6!$

(ii) $2! \times (6+1)!$ A, B, C, D, E, F, G

$= 2! \times 5!$

(iii) If 3 persons sit together

A, B, C, D, E, F, G

3!

$3! \times (5+1)! = 3! \times 4!$

3 persons not sitting together $= 6! - 3! \times 4!$

$= 720 - 144 = 576$

F) 5R, 2G, 3B

$\frac{10!}{5!\,2!\,3!}$

Note1 → We know that n diff. objects can be arranged in a row in n! ways

$n! \longrightarrow {}^nC_1 \times {}^{n-1}C_1 \times \cdots \times {}^2C_1 \times {}^1C_1$

1 2 n-1 n

Note2 But among n objects suppose p no. of objects are of one kind, & q no. of objects are of 2nd kind,

Date / /
Page

B) 4 M, 5 P, 6 C

① $(4! \times 5! \times 6!) \times 3!$

②

③ $5C_1$ (5 P), 4 M, 6 C

$(5!) \times 11!$

C) $^4C_1 \times {}^3C_1 \times {}^2C_1 \times {}^1C_1$

M1 W1 M2 W2 M3 W3 M4 W4 M5

$^5C_1 \times {}^4C_1 \times {}^3C_1 \times {}^2C_1 \times {}^1C_1$

$5! \times 4! = 120 \times 24$

D) 5 M, 7 P ⇒ 3 M & 2 P

① $^5C_3 \times {}^7C_2 = \cancel{350}$ RTO

5M → $^5C_3$ → 3M
7P → $^7C_2$ → 2P

② $^4C_2 \times {}^7C_2$

4M → $^4C_2$ → 1M, 2M
7P → $^7C_2$ → 2P

③ 5M → $^5C_3$ → 3M
6P → $^6C_2$ → 2P
1P

E) We know that n diff. objects can be placed around a table of any shape in ~~that case~~ (circular, or triangle or rectangular any shape) in $(n-1)!$ ways.

Date / /
Page

D→ Out of 5 mathematician & 7 physicists a committee consisting of 3 mathematician & 2 P is to be formed in how many ways this can be done if:

(1) Any M or any P can be included in comitee.

(2) 1 particular M must be in commitee always.

(3) 1 particular P can not be in commitee.

E→ In how many ways can 7 people be sitted at a round table if

(1) They sit anywhere.

(2) 2 particular people are sitting together always.

(3) 3 particular people must not sit together.

F→ 5 red marbles, 2 green & 3 blue marbles are arranged in a row, if all the marbles of same color are not distinguishable from each other then how many diff. arrang. are possible.

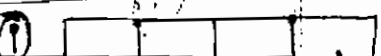
G→ From 7 consonants & 5 vowels how many words can be formed consisting of 4 diff. consonants and 3 diff. vowels

A) (1) ${}^{10}C_4 = \frac{10!}{4!\,6!} = \frac{10\times9\times8\times7}{4\times3\times2} = 210$

(2) ${}^{10}P_4 = \frac{10!}{6!} =$

(3) $\frac{\quad}{\quad} = \frac{7}{\quad} = {}^{9}C_3$

B)

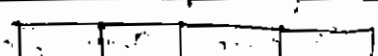


${}^{9}C_1 \times {}^{9}C_1 \times {}^{8}C_1 \times {}^{7}C_1$

$= 9\times9\times8\times7 = 4536$

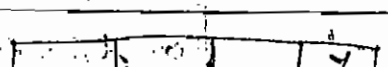
C→

(2)



${}^{9}C_1 \times {}^{10}C_1 \times {}^{10}C_1 \times {}^{10}C_1 = 9000$

(3)



${}^{8}C_1 \times {}^{8}C_1 \times {}^{7}C_1 = 448$

Date / /
Page

committee ⇒ Let x, y, z 3 diff. items, then we can select
formed any two items in $3C_2 = \frac{3!}{1!\,2!} = 3$
! (x, y) (x, z) (y, z).
committee.

always. ⇒ Similarly, after selecting 2 items from 3, if we
ittee. arrange them in an order, then total no. of
arrangements $3P_2 = \frac{3!}{1!} = 6$
filled (x,y) (y,x) (x,z) (z,x) (y,z) (z,y)

always. A → How many 4 digit nos. can be formed with 10 digits from 0, 1, 2, ---9, if

1) Repetitions are not allowed.

2) Repetitions are allowed.

ers are 3) Last digits must be 7 & repetitions are not allowed.

of same 4) 4
other than

B → 4 diff. mathematics books, 5 diff. Physic & 6 diff. Chemis books are to be arranged on a shelf, then how many diff. arrangements are possible, if

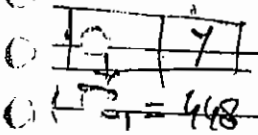
words ① Books in each particular subject must all
nts and standing together.

② Only physics books must stand together.

~~③ It is required~~

C → It is required to sit 5 men & 4 women in a row so that the men occupy odd places, then how many such arrangements are possible?

$= 4! \times 5! = 4!5!6$

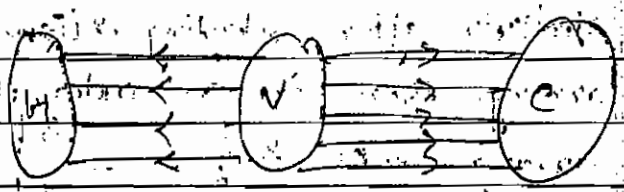


Date / /
Page

→ Suppose, A person at H railway station reaching C, via V, i.e. he is doing both activities A_1 & A_2 then the total no. of ways of reaching C is

$4 \times 5 = 20$

Case 2:



→ If the person is at V then he can do either A_1 or A_2 but not both i.e. the total no. of ways of leaving the station V by person

$4 + 5 = 9$

Combination → Selection

Permutation → Arrangement

Combination:

Let there be 'n' diff. objects available. Among n objects, a 'r' no. of objects being selected the total no. of such selection given by

$$^nC_r = \frac{\lfloor n}{\lfloor n-r \cdot \lfloor r}$$

Permutation:

Suppose n diff. objects, r no. of objects being selected & also arranged in proper order, then total no. of such arrangement

$$^nP_r = \frac{\lfloor n}{\lfloor n-r}$$

Case 1:

Date / /
Page

PERMUTATION:

PROBABILITY:

Fundamental Concepts in Counting :-

Product Rule —

Let there be n activities $A_1, A_2 - - A_n$. Suppose A_1 can be done in m_1 ways, A_2 can be done in M_2 ways & same as An can be done in Mn ways, if all the activities are completed to get the job completed.

Total no. of ways of getting the job completed is given by $M_1 \times M_2 \times - - \times M_n$, This is called product rule.

Sum Rule —

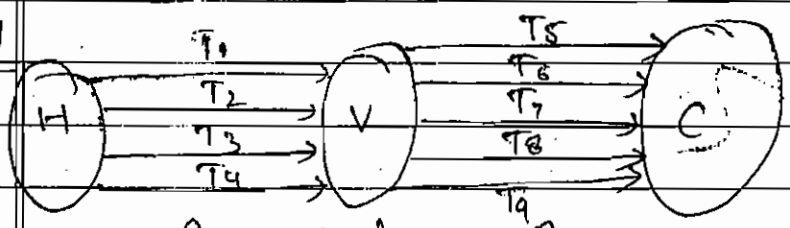
Suppose among n activities as mentioned above, any 1 activity is over to get the job done i.e. A_1 is completed or A_2 is completed or - An is completed then total no. of ways of getting the job done is

$M_1 + M_2 + M_3 + - - - + M_n$.

and : **, ∩, ∩, Cond.

or : +*, ∨, ∪, Total

Case : 1



A_1 and A_2

A_1 : Reaching from H to V

A_2 : Reaching from V to C.

2

Date / /

Page

4:00

→ 05:30 - 09:00 : wakeup / Ready / food / 1hr study

09:00 - 07:30 : office / Rest

07:30 - 09:30 : Cooking / Dinner

→ 09:30 - 10:00 : Study.

11:00 - 05:30 : Sleeping

Facts about Watches

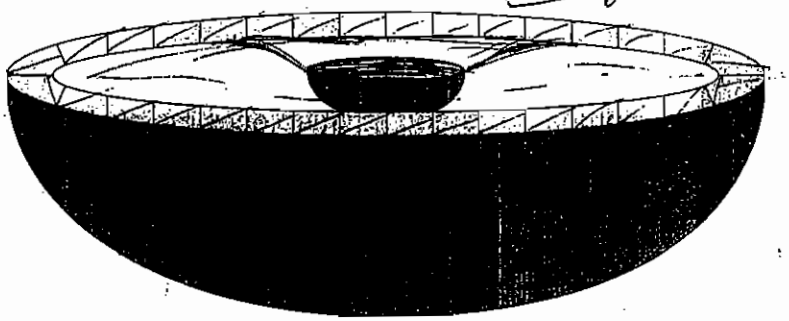
- The first pocket watches were invented in Tudor, England, in the 16th century. These pocket watches were incredibly big, which is why people wore them around their neck.
- Breguet was the most preferred watch of the Russian Tsars. Breguet was also worn by Napoleon at Waterloo.
- In 1795, Breguet was the one to invent the tourbillon, which till date represents a great achievement in making time pieces. It compensates watch errors caused by gravity. Even today tourbillon is one of the most sophisticated mechanisms to manufacture.
- The first wristwatch was invented by Patek Philippe. Until the First World War the wristwatch was considered to be a woman's accessory. Till then men wore pocket watches.
- The first wristwatch for men was invented by Cartier. The famous jeweler created a watch for his friend, Brazilian pilot Alberto Santos-Dumont.
- The first mass production of wristwatches was started by Girard Perregaux. The company created its timepieces mainly for military use.
- The Tag Heuer Carrera Chronograph was made in memory of famous Carrera Panamericana automobile races that were held in 1950s.
- When Sir Edmund Hilary climbed Mt. Everest (the first human to conquer the mountain) in 1953 he wore a Rolex Oyster.
- The Russian and American astronauts wore Omega Speedmaster during the world's first craft meeting of Apollo-Soyuz that took place in 1975.
- The world's first anti magnetic watch was created by the famous Swiss watchmaker Tissot.
- The Jantar Mantar at Delhi consists of 13 architectural astronomy instruments, built by Maharaja Jai Singh II of Jaipur, from 1724.

Source: Internet

1.c 2)c 3)c 4)a 5)a 6 7)d 8)b
9) b 10) c 11) d 12) a 13) b 14) c 15) b 16) b
17) a 18) a 19) a 20) b

Facts about Watches

The Egyptians built a device to divide their day into parts something like our hours called 'Obelisks' (slender, tapering, four-sided monuments) as early as 3500 BCE. Their moving shadows formed a kind of sundial, enabling people to partition the day into morning and afternoon.



Water clocks were among the earliest timekeepers that didn't depend on the observation of celestial bodies. They were stone containers designed to slowly fill with water coming in at a constant rate. Markings on the inside surfaces measured the passage of "hours" as the water level reached them. Another version consisted of a metal bowl with a hole in the bottom; when placed in a container of water the bowl would fill and sink in a certain time. These were still in use in North Africa in the 20th century.

Source: Internet