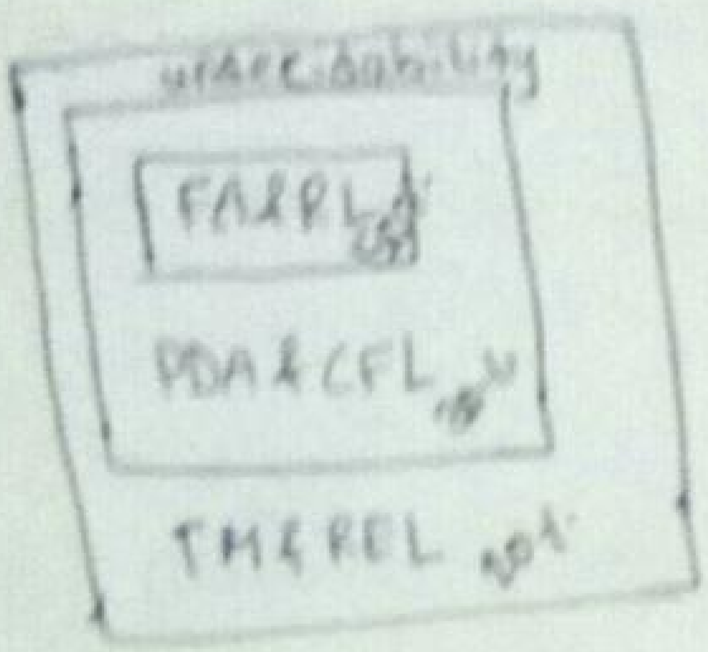
15/12/19 # Theory of Computation (6-8M) (9491919183)

Dasaradh. R.k

Syllabus:



Text books:

1. Introduction to formal languages & Automata by peter linz
2. Theory ~~of~~ of Computation by I.A. Daniel
3. * Introduction to language, automata & computation by Ullman, Aho
4. Introduction to automata & compiler design by Dasaradh. R.k

TOC

Automata: Mathematical sys used to do certain computation

(i) FA
(ii) PDA
(iii) Linear Bounded Automata (LBA)
(iv) TM

Formal Languages

(i) Regular
(ii) Context free
(iii) Context sensitive
(iv) Recursively Enumerable

Fundamentals & Terminology:

Alphabet: An alphabet is finite set of symbols denoted by Σ.

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

$\Sigma = \{0,1,2 \dots 9\}$ valid

$\Sigma = \{0,1,2 \dots\}$ invalid

words / strings / sentences: A finite sequence of symbols over some fixed alphabet called as string, which is denoted by 'w' or 'x'.

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

$w = 101110$ – valid string

$w = 101121$ – invalid string

length of the string: The no of symbols composing the string is called length of the string denoted by $|w|$

Empty string is denoted by ε (or) λ (or) Λ

Eg: $w = 10110 1 = 1\varepsilon 011\varepsilon 01\varepsilon$

$|w| = 6$ ↳ length is still 6.

power alphabet: The power alphabet Σ^k denotes set of all strings of length k over Σ.

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

$\Sigma^0 = \{\epsilon\}$

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

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

⋮

→ Σ^* – All strings of length 0 or more (i.e., Universal set over the alphabet)

$$\Sigma^* = \Sigma^0 + \Sigma^1 + \Sigma^2 + \cdots = \sum_{i=0}^{\infty} \Sigma^i$$

→ Σ^+ = All strings of length 1 or more.

$$\Sigma^+ = \sum_{i=1}^{\infty} \Sigma^i = \Sigma^* - \varepsilon$$

Operations on string:

prefixes: All possible leading symbol substrings

Suffixes: All possible tail end symbol substrings

Eg: w = toc

prefixes: ϵ, t, to, toc Proper prefixes: t, to, toc } exclude 'ϵ'

suffixes: ϵ, c, oc, toc proper suffixes: c, oc, toc

Concatenation: To obtain concatenation of two strings w_1 & w_2. Write w_1 followed by w_2 without any space b/w them

w_1 = cat
w_2 = walk } $w_1 w_2$ = catwalk

palindrome: A string w is said to be a palindrome, it should be identical whether we read forward or backward

Eg: ~~Sol(or1)~~ $\Sigma = \{0,1\}$

palindromes: 0, 00100, 1001

Consider the following alphabet and the string w_1 & w_2

$\Sigma = \{0,1\}$ $w_1 = 0011$ $w_2 = 1100$

(a) find no of strings of length 3 begins and end with 1. - 2

(b) $|w_1 w_1| = 8$

(c) $\{w_1 w_2 w_1\} = 001111000011$

(d) whether $w_1 w_2$ is it palindrome? - Yes

(e) no of prefixes of w_1 - 5 (length + 1)

(f) No of common suffixes of w_1 & w_2 $[\epsilon]$

g) How many string of length 5 are palindromes with '1' in 3rd place: 4

Formal Languages: Set of (finite or infinite) strings over some fixed alphabet is called is language denoted l or L.

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

$l = \{01, 0011, 000111 \dots\}$

$l = \{01, 0^2 1^2, 0^3 1^3 \dots\}$

$l = \{0^n 1^n \mid n > 0\}$

$l = \{w \in \{0,1\}^+ \mid$ every w has equal no of 0's followed by equal no of 1's$\}$

→ Let L_1 ~~can~~ be described as set of all strings over set $\{0,1\}$ with equal no of 0's and ~~eq~~ ~~fo~~ followed by equal no of 1's.

→ List the members of the language over the alphabet $\{0,1\}$, all strings of length 4.

$\{0000, 0001 \cdots 1111\}$

$L = \{w \in \{0,1\}^+ \mid |w| = 4\}$

Operations on language:

→ $L_1 = \{\ \} = \phi$ (empty language)

→ $L_2 = \{\epsilon\} = \epsilon$ (empty string language) (Its size is one)

$(L_1 \neq L_2)$

Countably infinite : Eg: $\{1, 2, 3 \ldots\}$

Uncountably infinite : Eg: $\{x \mid x$ is a real number$\}$

$\{12.11, 12.12, 12.111 \ldots\}$

** → All formal languages are either finite or countably infinite but no uncountably infinite. **

Previous

Q: Consider the following strings over alphabet $\Sigma = \{a, b, c\}$

$w = \underline{ba}acabb$ $\quad 7 + 6 \ldots$ 29

find the no of substring of the w?

Sol:

No of substring of length 0 = 1

" " " " 1 = 7

" " " " 2 = 6

3 = 5

⋮

7 = 1

$\frac{7 \times 8}{2} + 1 = \underline{29}$

0 length	1 length	2 length	3	4	5	6	7	
ϵ	a	ba	5	4	3	2	1	=> 25
	b	aa						
	c	ac						
		ca						
		ab						
		bb						

Q: Let $\Sigma = \{0,1\}$. How many strings are possible of length n?

a) n b) $n+1$ c) 2^n d) $2n$

Q: Let $\Sigma = \{a,b,c\}$. no of strings of length ≤ 3.

0	1	2	3	
1	3	9	27	= 40

Note:

Let Σ be the alphabet with m symbols, the possible no of strings of length n are: m^n

Language Operations:

Union: $l_1 \cup l_2 = \{w \mid w \text{ is in either } l_1 \text{ or } l_2\}$

Intersection: $l_1 \cap l_2 = \{w \mid w \text{ is in both } l_1 \,\&\, l_2\}$

Concatenation: $l_1 l_2 = \{xy \mid x \in l_1 \,\&\, y \in l_2\}$

Complementation: l_1' or $\bar{l_1}$ or $\sim l_1 = \{w \mid w \in \Sigma^* \text{ but not in } l_1\} = \Sigma^* - l_1$

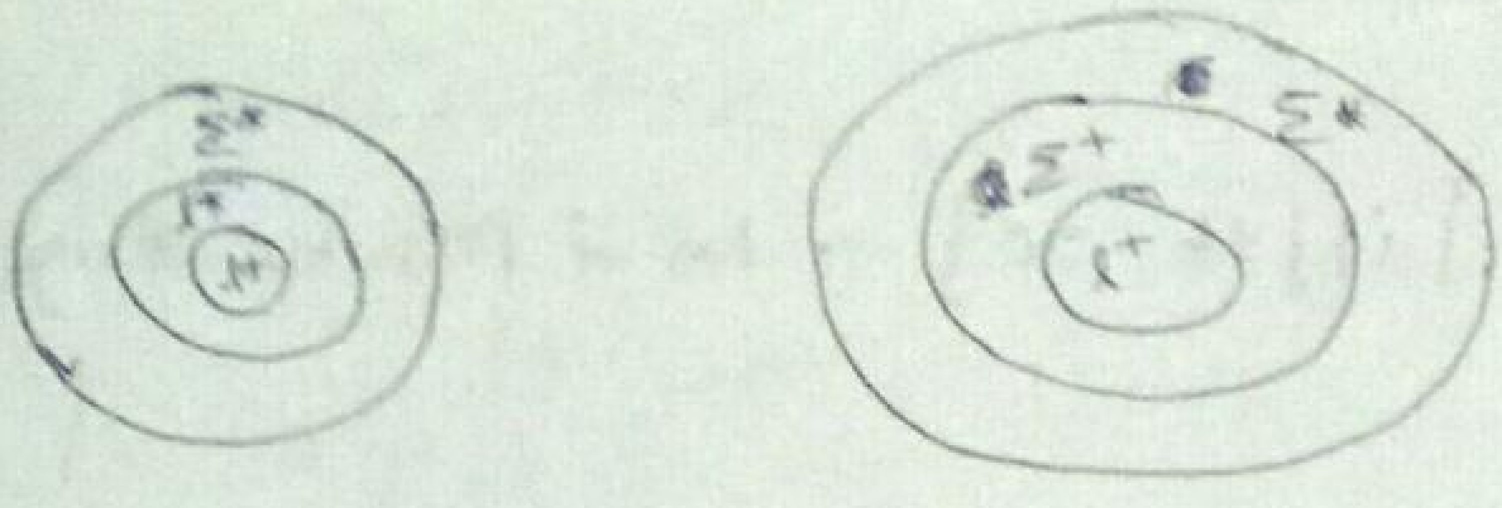
Kleen closure: $l^* = l^0 \cup l^1 \cup l^2 \ldots$ i.e., zero or more concatenations of l.

$\Rightarrow l^0 = \{\epsilon\}$

$l^1 = l$

$l^2 = l \cdot l$ (concatenation)

Positive closure: $l^+ = l^* - l^0$ i.e. one or more concatenations of l.



Q: Consider the following languagage $L=\{a^ib^j \mid i=j>0\}$

which of the following represents complementation of L

a) $L=\{a^ib^j \mid i\neq j>0\}$ abab

b) $L=\{a^ib^j \mid i\neq j\geq 0\}$

c) $L=\{w \mid w\in\{a,b\}^+$ and w contains unequal no of a's, b's$\}$

✓d) None of the above.

Correct answer is d

~~$L=\{w \mid w\in\{a,b\}^*$ and w contains unequal~~

Q: Consider the following lang over the alphabet = $\Sigma=\{0,1\}$. which of the following is true

(i) $L_1=\{w \mid w\ n_0(w)=n_1(w)\}$,

where n_0 = no of zeroes in w

n_1 = no of ones in w

(ii) $L_2=\{w \mid w$ contains all strings $\{0,1\}\}$

(iii) $L_3=\{w \mid w$ contains all palindroms over $\{0,1\}\}$

a) Find the no of strings of length 4 common in all three lang. - 2 (1001, 0110)

b) which one of the following relationship true of these lang

a) 1) $L_1\subseteq L_2\subseteq L_3$

2) $L_2\subseteq L_1\subseteq L_3$

3) $L_2\subseteq L_3\subseteq L_1$

✓4) None of the above

Q: Consider $L = \{0^i 1^j 2^k \mid i \neq j \neq k > 0\}$. No of possible strings of length 2 or less.

0	1	2
ϵ	~~0~~	
	~~2~~	

0	1	2
$0^0 1^0 2^0$ not valid	$0^1 1^0 2^0$ (not valid)	$0^2 1^0 2^0$ (N.V)
	$0^0 1^1 2^0$ (N.V)	$0^0 1^2 2^0$ (N.V)
	$0^0 1^0 2^1$ (N.V)	$0^0 1^0 2^2$ (N.V)

Ans: 0

Q: Let w be the string of length n and w contains diff symbols. How many non empty substring string are possible with w?

a) 2^n

b) $2n$

c) $n+1$

✓d) $\frac{n(n+1)}{2}$

Check computing substring by removing prefix & suffix

Sol:

No of substring of length 'n' = 1
" " " " " n-1 = 2
⋮
2 = n-1
1 = n

$$\left.\begin{matrix} n = 1 \\ n-1 = 2 \\ \vdots \\ 2 = n-1 \\ 1 = n \end{matrix}\right\} \text{sum} = \frac{n(n+1)}{2}$$

Q: Set of all strings of length n can how many substrings

Ans: $(n+1)$ to $\frac{n(n+1)}{2} + 1$

when all symbol are equal

when all symbols are distinct.

Q: If $L^* = L^0 + L^1 + L^2 + \cdots$ then $L^+ =$ ______

a) $L^* + \varepsilon$

b) $L^* + L$

c) LL^*

d) None

Sol:

$$LL^* = L(\varepsilon + L + L^2 + L^3 + \cdots)$$

$$= L + L^2 + L^3 + \cdots = L^+$$

Q: Let $L = \{w \mid w \in \{0,1\}^* \text{ and } n_0(w) \neq n_1(w)\}$

Find the strings of length 4 or less in L'.

~~0~~	~~1~~	~~2~~	~~3~~	~~4~~
~~1~~	~~2~~	~~2~~	~~8~~	$16 - {}^4C_2$
				10

= 2

Method 1:

For L

0	1	2	3	4
0	2	2	8	$16 - {}^4C_2$
				$= 10$

$= 22$

Total no of string of length 4 or less $= 2^5 - 1 = 31 \quad (2^0 + 2^1 + \cdots + 2^4 = 2^{4+1} - 1)$

$\therefore$ Ans $= 31 - 22 = 9$

Method 2:

Directly compute L'

$L' = \{\varepsilon, 00, 11, 0011, 1010, 1100, 0101, 1001, 0110\} \rightarrow 9$

Q: An lang defined over a alphabet $\Sigma = \{0,1\}$ and every w length is 5 such that first place, 4th place and last place are 1's. Find no of string in lang.

Sol:

1 _ _ 1 _ _ 1

$2^4 = 16$

Aut

Finite Automata:

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

- Q → set of states
- Σ → input symbols
- δ → Transitions
- q_0 → initial state
- F → set of final states

→ can be represented by (i) state diagram / transition diagram / transition system
(ii) state table / transition table

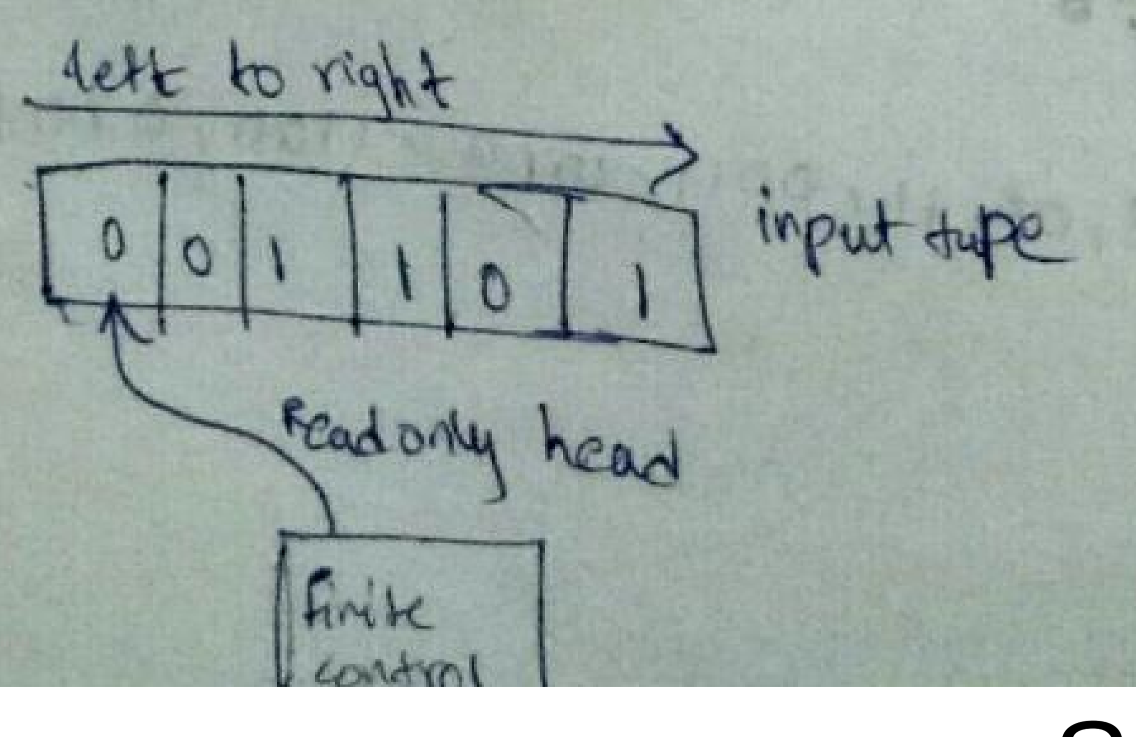
→ FA is a mathematical system consists of 5 tuples

→ →O ---- initial state

◎ ---- final state

→ Although there are many applications with the F.A, the main obj of the F.A is to perform certain computational operations in the computer science. The following is finite automata to ~~iden~~ identify whether binary number has odd zeroes or even zeroes

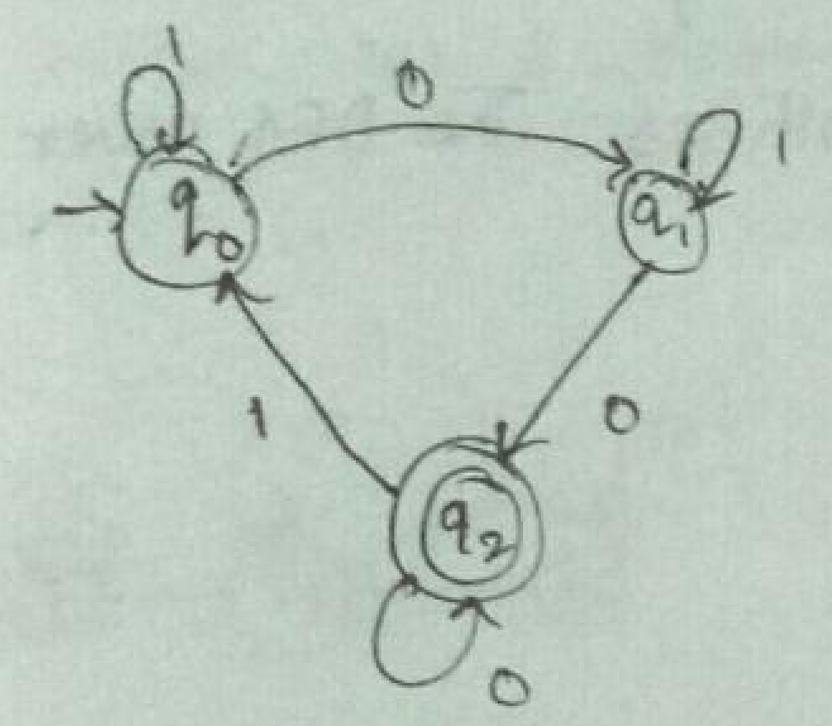
Block diagram:



A F.A's basic obj is recogonisation of string. A string w is said to be accepted by F.A if and only if a machine terminated at final state by reading the last symbol in w. In other words a string w said to be accepted by F.A $\Leftrightarrow$ $\delta(q_0, w) = p$ for some 'p' in 'F'.

Q: Conside below F.A

101010



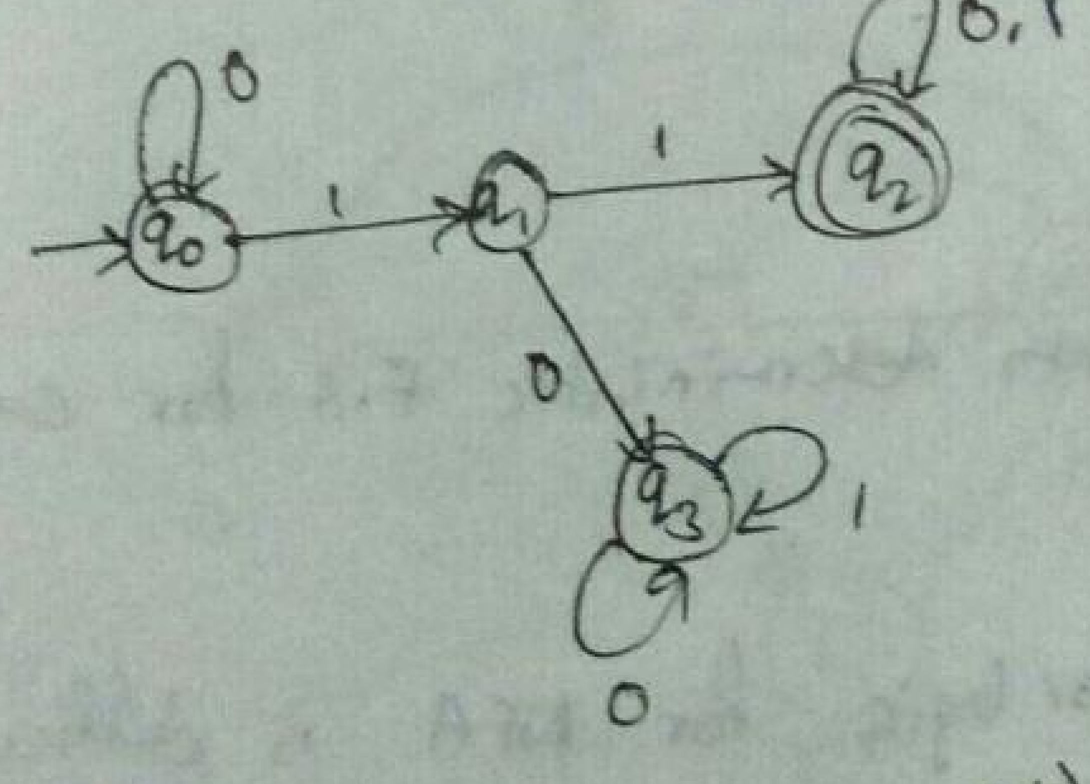
Check whether the following strings are recogonised by the F.A or not

✓a) 101110

×b) 010101

×c) 1001

Q: Consider following F.A



×a) 010010

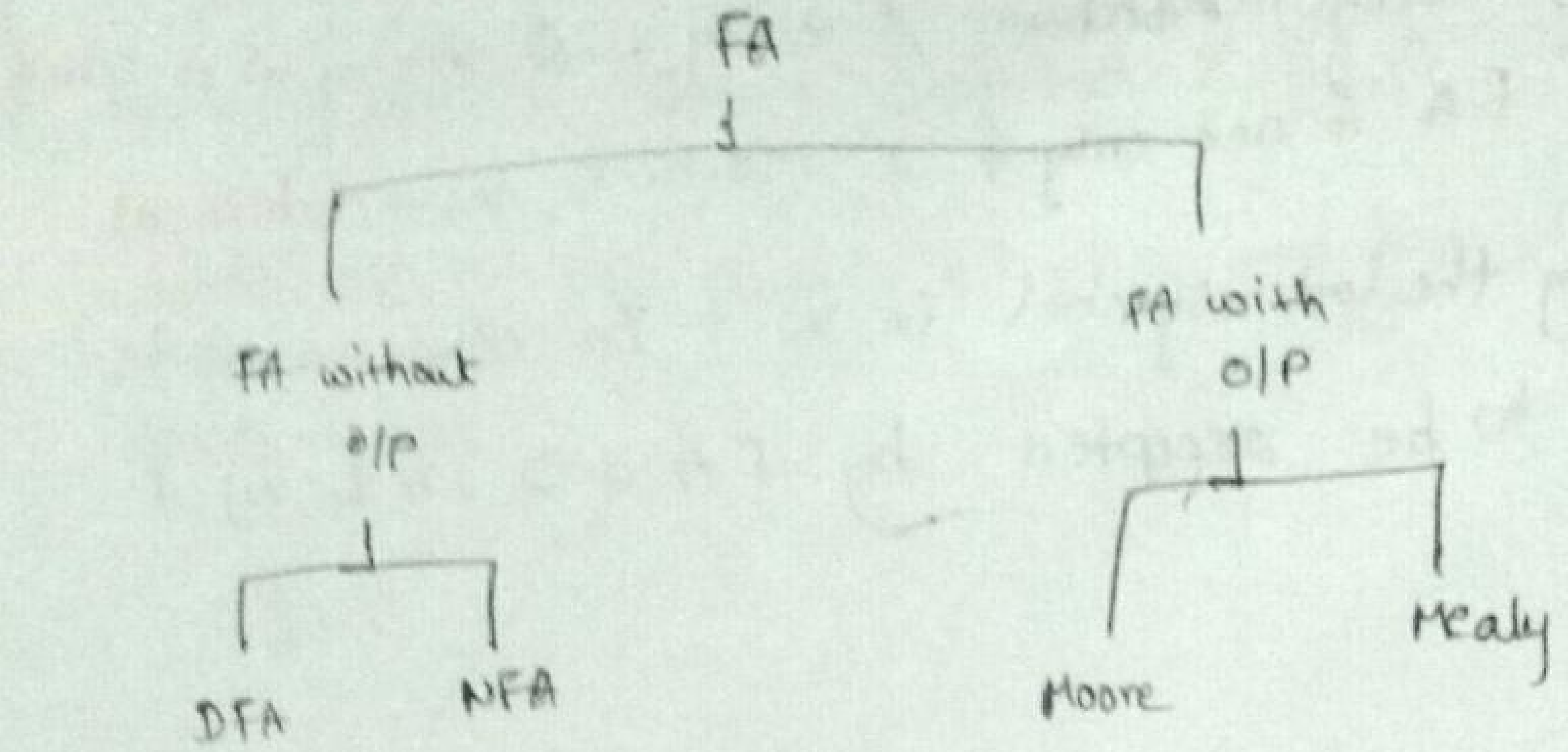
×b) 1000110

×c) 010110

×d) 100111001

$l(M) = \{11, 011, 110, 111, 0011, 0110 \ldots\}$

→ language of M can be defined as set of all strings over {0,1} with any zeroes followed by 2 consecutive 1's followed by any combination



DFA:

→ In DFA for i/p symbol there exist exactly one transition

→ The state table of DFA consists of single entities. The DFA never permits ϵ transitions.

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

NFA:

→ It is a 5 tuple system

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

δ: defined as

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

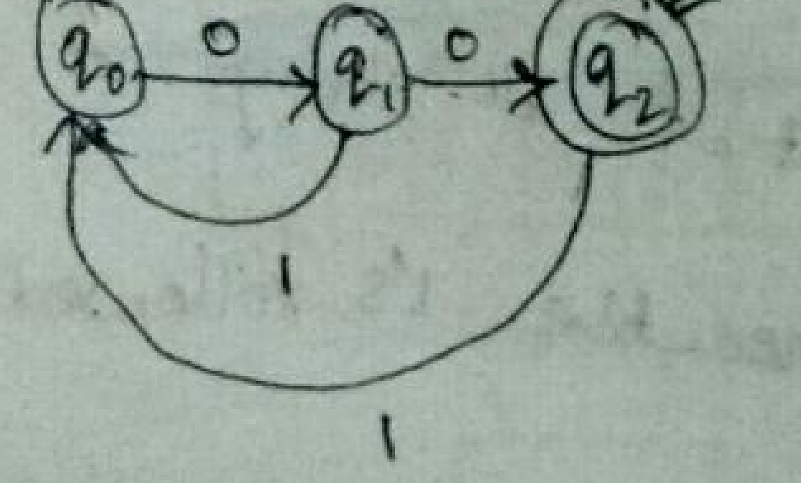
→ For each i/p symbol of NFA there may exist one or more transitions.

→ The state table of NFA consists of multiply defined entries. The NFA may also permits ϵ-transitions.

→ All NFA are generally translated into deterministic F.A for easy & secured implementation of code

i.e., developing a mechanism or logic for NFA is difficult than that of DFA. Hence we do translate NFA's to DFA's.

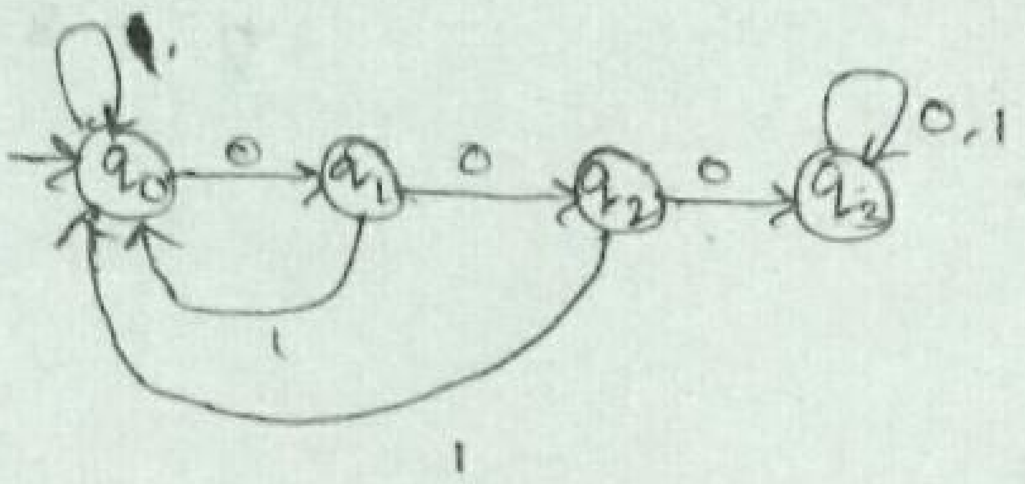
Eg: Design a DFA to recognise set of all strings over {0,1} ending with 00



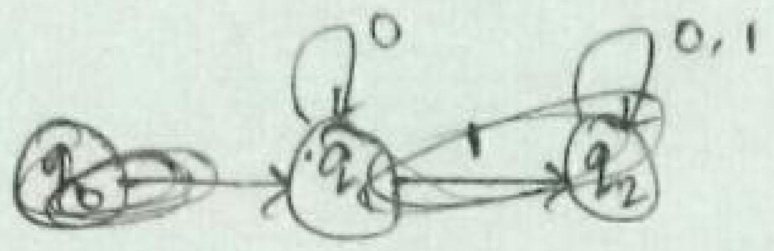
$L = \{00, 000, 100, 0000, 0100, 1000, 1100 \ldots\}$

Q: Design a DFA to recogonise set of all strings containing the Substring '000'.

$L = \{000, 1000, 0000, 0001, 10001, \ldots\}$.



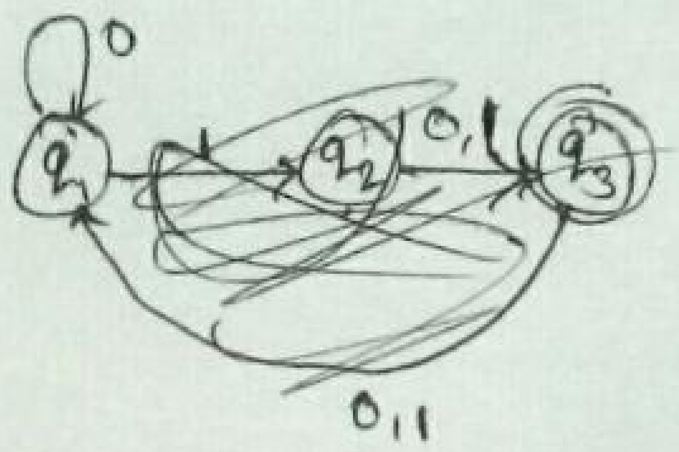
Q: DFA to accept set of strings over $\{0,1\}$ such that 10th symbol from right end is '1'.

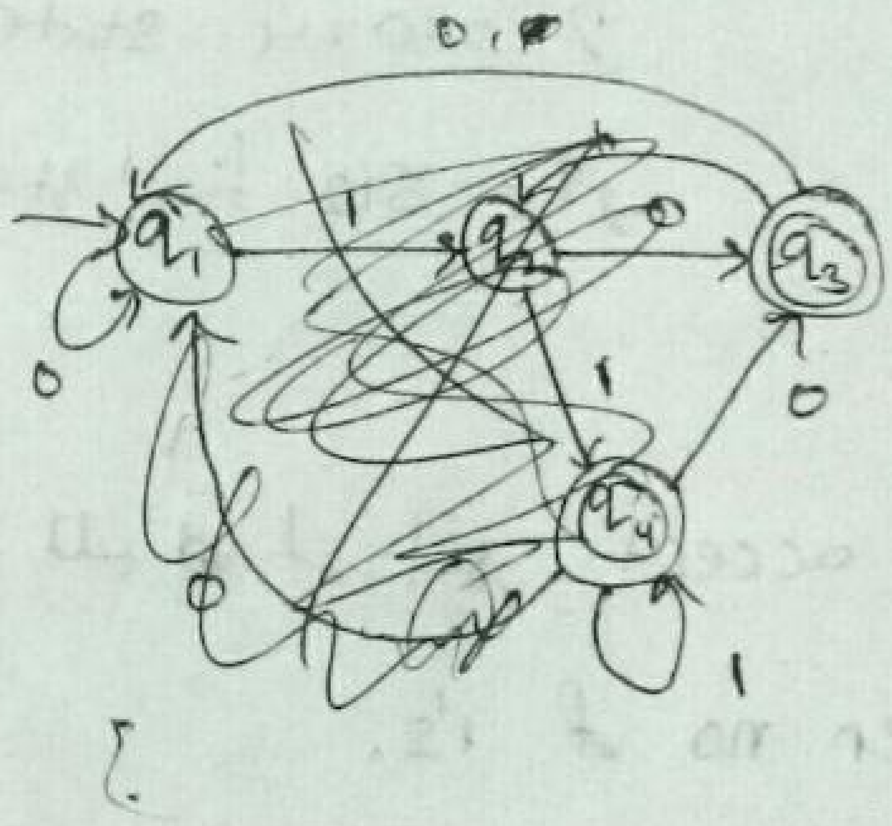


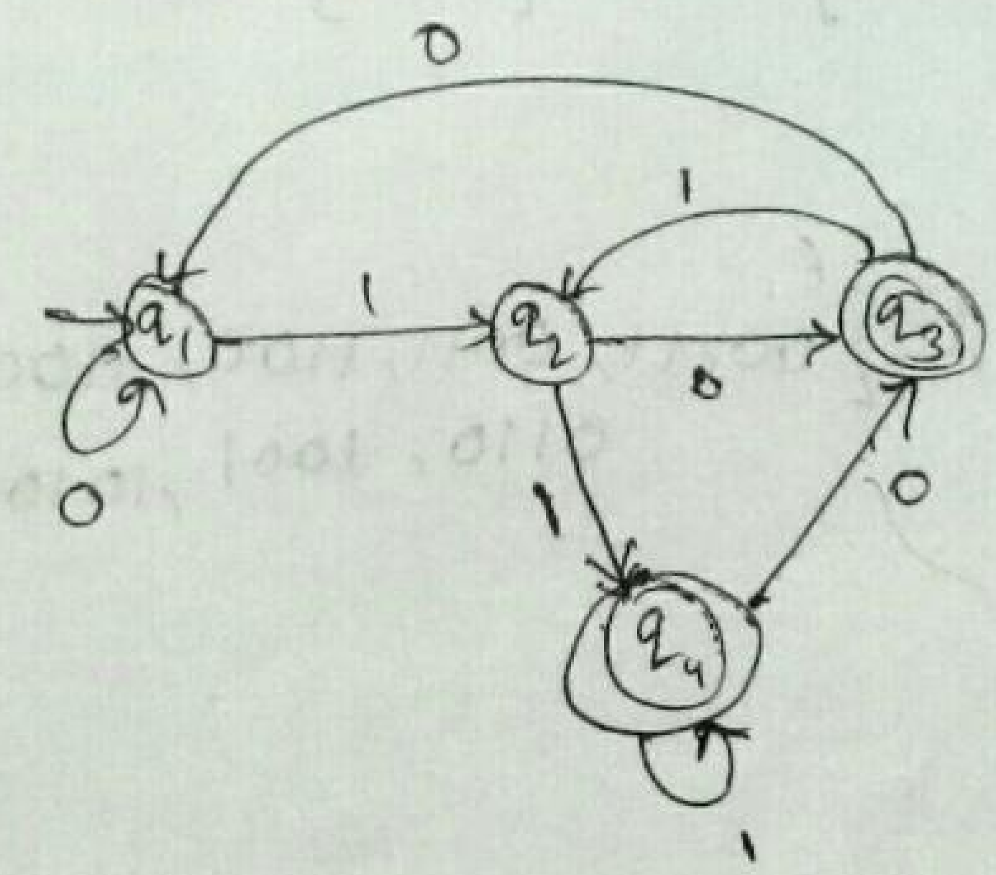
First design DFA - such that 2^{nd} symbol from right end is '1'.

$L = \{10, 11, 110, 111, 010, 011 \ldots\}$

11011







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

General Method:

2^{2-1} ⟶ corresponds to nth digit from right side

⟹ 2 final states

2^2 ⟶ states

lets design for 1 at 3rd place from right end.

$2^{3-1} = 4$ final states

$2^3 = 8$ states

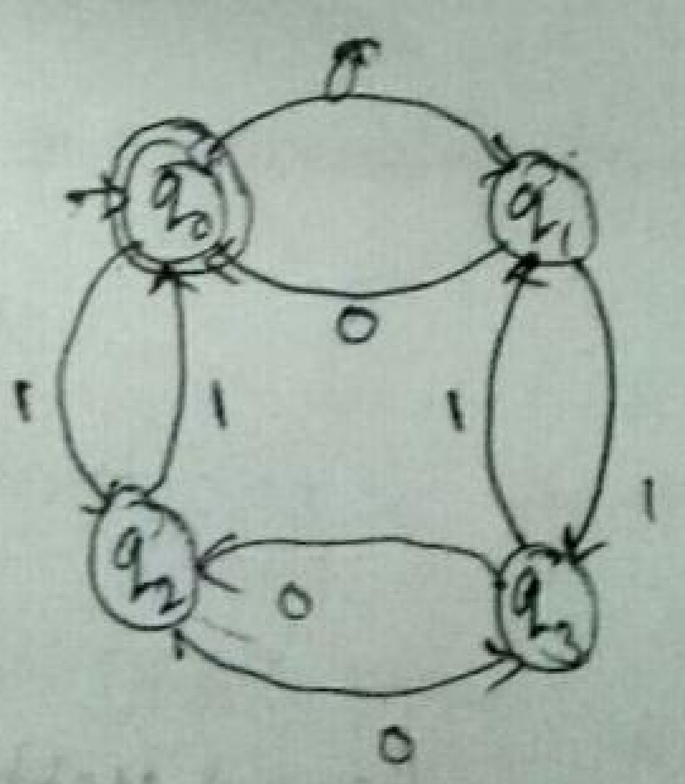
	0	1
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_5
q_3	q_6	q_7
q_4^*	q_0	q_1
q_5^*	q_2	q_3
q_6^*	q_4	q_5
q_7^*	q_6	q_7

for 1 at 10th place from right end

$2^{10} = 1024$ states

$2^9 = 512$ final state

Q: to accept set of all strings over $\{0,1\}$ with even no of 0's & even no of 1's.



$L = \{\epsilon, 00, 11, 0011, 1100, 110011, 001100, 000011, 0110, 1001, 10100101\}$

In above fig if q_1 is final state - odd 0's even 1's

q_2 is final state - even 0's odd 1's

q_3 is final state - odd 0's odd 1's