CS23010

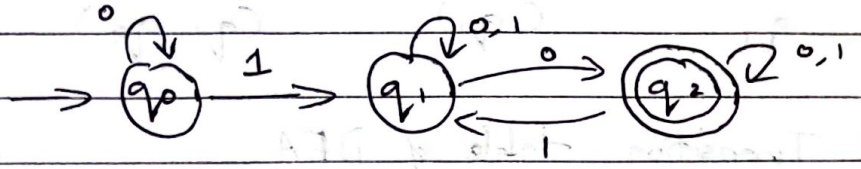
PRABHAT
PAGE NO. :
DATE : / /

Assignment 2

1) Convert NDFA to DFA : The diagrams are given below

⇒

i)



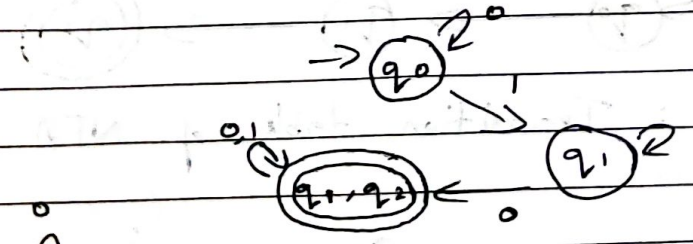
Step 1 : Transition table of NDFA

	0	1
→ q_0	q_0	q_1
* q_1	(q_1, q_2)	q_1
* q_2	(q_2)	(q_1, q_2)

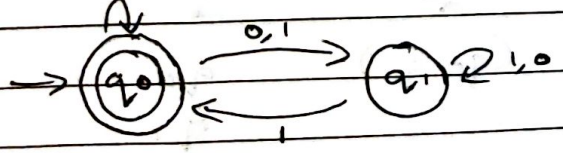
Step 2 : Transition of DFA

	0	1
→ q_0	q_0	q_1
q_1	(q_1, q_2)	q_1
(q_1, q_2)	(q_1, q_2)	(q_1, q_2)

Step 3 : Convert DFA transition to diagram



ii)



PRABHAT
PAGE NO. :
DATE : / /

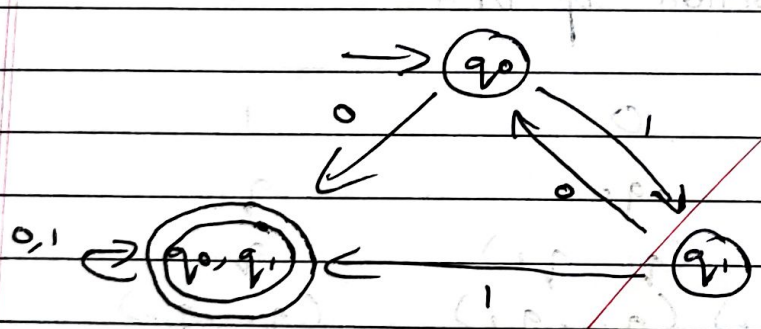
Step 1 : Transition table of NFA

	0	1
→ q_0	(q_0, q_1)	q_1
* q_1	(q_1)	(q_1, q_0)

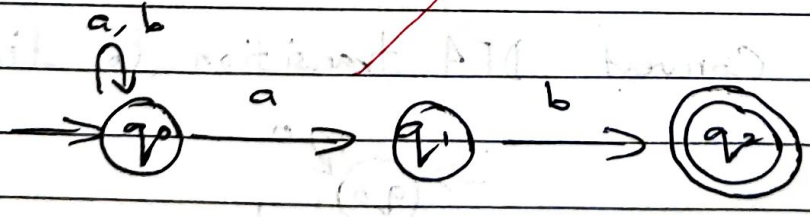
Step 2 : Transition table of DFA

	0	1
→ q_0	(q_0, q_1)	q_1
q_1	(q_0)	(q_0, q_1)
* (q_0, q_1)	(q_0, q_1)	(q_0, q_1)

Step 3 : Convert DFA transition table



iii)



Step 1 : Transition table of NFA

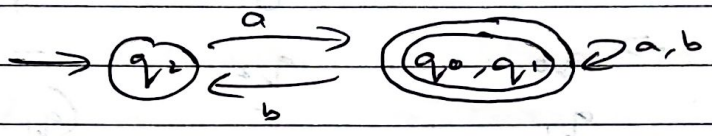
	a	b
→ q_0	(q_0, q_1)	q_0
* q_1	–	q_2
q_2	–	–

PRABHAT
PAGE NO :
DATE : / /

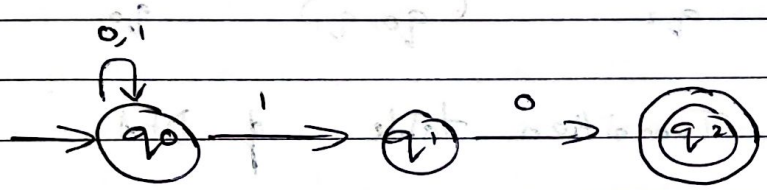
Step 2 : Transition of DFA

	a	b
→ q_0	(q_0, q_1)	q_0
* (q_0, q_1)	(q_0, q_1)	(q_0, q_1)

Step 3 : Conversion DFA transition to diagram



iv)



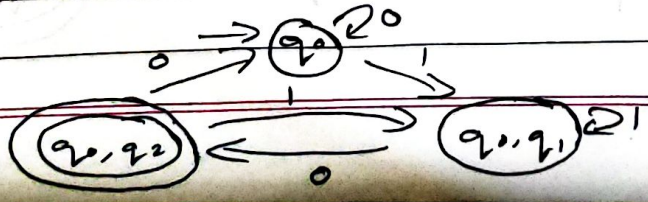
Step 1 : Transition of NFA

	0	1
→ q_0	q_0	(q_0, q_1)
q_1	q_2	–
* q_2	–	–

Step 2 : Transition of DFA

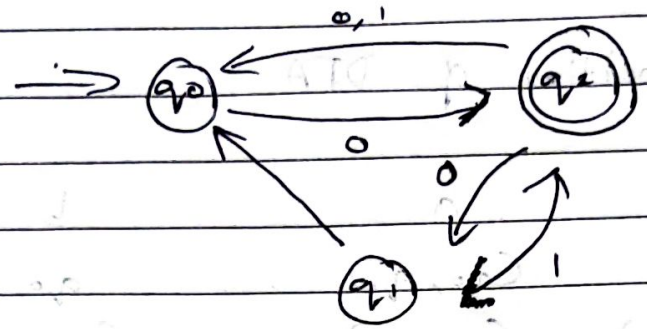
	0	1
→ q_0	q_0	(q_0, q_1)
(q_0, q_1)	(q_0, q_2)	(q_0, q_1)
* (q_0, q_2)	(q_0)	(q_0, q_1)

Step 3 : Conversion DFA transition to diagram



PRABHAT
PAGE NO. :
DATE : / /

v)



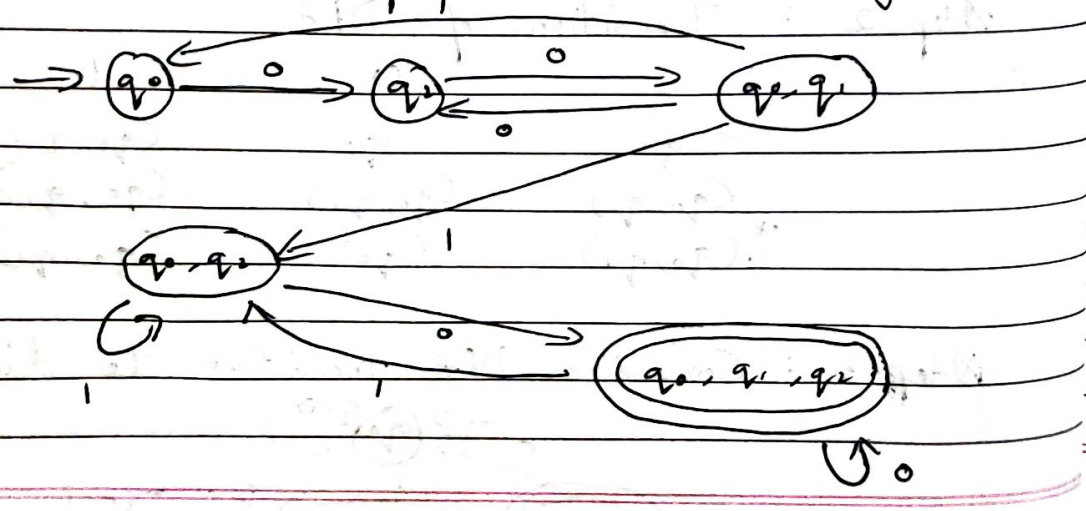
Step 1: Transition table of NFA

		0	1
→	q_0	q_2	–
	q_1	–	(q_1, q_2)
*	q_2	(q_0, q_1)	q_0

Step 2: Transition table of DFA

		0	1
→	q_0	q_2	–
	q_2	(q_0, q_1)	q_0
	(q_0, q_1)	(q_2)	(q_1, q_2)
	(q_0, q_2)	(q_2, q_1, q_0)	(q_0, q_2)
	(q_0, q_1, q_2)	(q_0, q_1, q_2)	(q_0, q_2)

Step 3: Conversion of DFA transition to diagram



PRABHAT
PAGE NO. :
DATE : / /

Q Difference between mealy and Moore machine

Mealy machine	Moore machine
* Output does depends on the present state as well as present input	* Output depends only upon the present state
* It places its output on the transaction	* Moore also places its output in the state
* Less no. of states are required	* More states are required
* Difficult to design	* Easy to design
* Has fewer or the same states as that of the moore machine	* Has more or the same states as that of the mealy machine.

Q Define the term Terminals and Non-Terminals.

→ Terminals-

Terminal symbol are those that are the components of the sentences generated using grammar and are represented using small case letters like a, b, c, d, etc.

Non-terminals

Non terminals symbols are those symbols that take part in the generation of the sentence but are not the component of the sentence. Non-terminal symbols are also called Auxiliary Symbols and Variables.

Eg: A, B, C, etc.

PRABHAT
PAGE NO. :
DATE : / /

Q Define grammar. Write components of grammar. Explain it with example.

→ Grammar: A grammar 'G' is defined as quadruple.

$G = \{V, T, P, S\}$

V = Variable (Only capital letter & it can be changed)

T = Terminal (Only small letter & it cannot be changed)

P = Production Rule

S = Start

Eg:

$S \to aSb / \varepsilon$

$S \to \varepsilon$ $\quad S \to \varepsilon$ $\quad S \to \varepsilon$

$S \to aSb$ $(S \to \varepsilon)$ $\quad S \to aaSbb$ $\quad S \to aaaSbbb$

$S \to ab$ $\quad S \to aabb$ $\quad S \to aaabbb$

∴ Language $= \{\varepsilon, ab, aabb, aaabbb, \ldots\}$

$= \{\varepsilon, ab, a^2b^2, a^3b^3, a^4b^4, \ldots a^n b^n\}$

Q How to generate language by grammar explain with example.

→ Set of all strings that can be derived from grammar is said to be language generated from the grammar.

Eg:

Consider the grammar

$G = (\{S, A\}, \{a, b\}, S, (S \to aAb, aA \to aaAb, A \to$

$S \to \underline{aA}b$ ... [By P rule] ... $S \to aAb$

$\to \underline{aaA}bb$ ... [$aA \to aaAb$]

$\to aaaAbbb$ ... [$aA \to aaAb$]

$\to aaabbb$ ... [by $A \to \varepsilon$]

PRABHAT
PAGE NO. :
DATE : / /

Final string aaabbb is derived from this grammar.

$G_2 = (\{S, A, B\}, \{a, b\}, S, \{S \to AB, A \to a, B \to b\})$

$S \to AB$

$\to ab$

$L(G_2) = \{ab\}$ ... Language generated by this

$G_3 = (\{S, A, B\}, \{a, b\}, S, \{S \to AB, A \to aA/a, B \to bB/b\})$

$S \to AB$

$\to ab$

$S \to AB$

$\to aAbB$ ... [by $A \to aA$, $B \to bB$]

$\to aabb$ ... [by $A \to a$, $B \to b$]

$S \to AB$

$\to aAb$ ... [by $A \to aA$; $B \to bB$]

$\to aabB$

$\to aabb$ ... [by $A \to a$, $B \to b$]

$S \to AB$

$\to aAB$ ... [by $A \to aA$, $B \to b$]

$\to aab$ ... [$B \to b$]

Depending upon choice we can generate many strings

$L(G_3) = \{a, a^2b^2, a^2b, ..\}$

Generalise Language

$L(G_4) = \{a^m b^n \mid m \geq 0 \text{ and } n \geq 0\}$

Q How to generate grammar by language. Explain with example?

-> Consider language and convert it into grammar which produce those languages.

Ex.

Lets consider $L = \{a^n b^m \mid n \geq 1 \text{ and } m \geq 0\}$

PRABHAT
PAGE NO. :
DATE : / /

language $L(a^n b^m)$ generate strings which having any power of n and m.

For $a^n = [a, aa, aaa \ldots]$

$A \rightarrow aA|a$

Consider if $m = 0$ ...

$\therefore b^m \Rightarrow b^0 \Rightarrow 1 \Rightarrow \varepsilon$

$b^m = \{\varepsilon, b, bb, bbb \ldots\}$

$B \rightarrow bB|\varepsilon$ ... This is grammar for b^m

Grammar is;

$S \rightarrow AB$

$A \rightarrow aA|a$

$B \rightarrow bB|\varepsilon$

* $L(a) = \{ab, aab, aabb, aaabb \ldots\}$

Generate more no. of a than b

$A \rightarrow aA|\varepsilon$

Generate equal no of a and b

$S \rightarrow aAb|asb$

$A \rightarrow aA|\varepsilon$

Q Explain Chomsky classification of grammar.

→ The chomsky hierarchy formulated by Noam Chomsky in 1956 is a classification of formal grammars that describes the generative power of different types of languages. It consists of four levels ordered from most to least restrictive:

1. Type-1: Context-sensitive language (CSL)
2. Type-2: Context-free Language (CFL)
3. Type-3 Regular Language

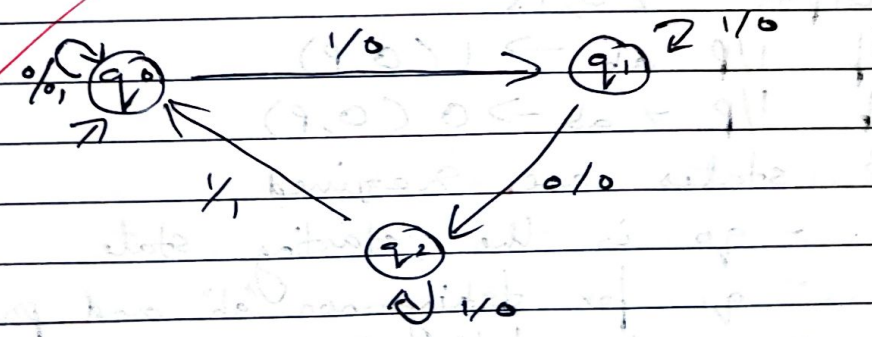
PRABHAT
PAGE NO. :
DATE : / /

Each level has specific rules and it's associated with a specific type of automation that can recognize the languages in that class.

Q Design a mealy machine that ~~diff~~ deflects the strings "101" in the input sequence and outputs 1 when "101" is detected and 0 otherwise.

→ I/P = {1, 0}
O/P = {1, 0}
if I/P '101' -> 1 (O/P)
if I/P ≠ 101 -> 0 (O/P)
3 states are required:
- q_0 is the starting state
- q_1 for taking '101' as I/P & O/P as 1
- q_2 for taking non '101' as I/P & O/P as 0.

L = {101, 1101, 1011, 1010}

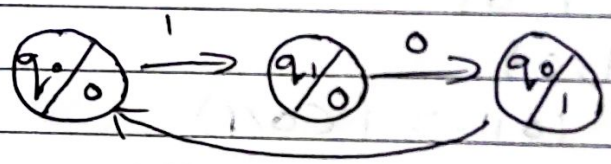


Q Design a Mealy machine that detects the string '110' in a given binary input sequence. The machine should give output 1 when the string '110' is detected and 0 otherwise.

→ I/P = {0, 1}
O/P = {0, 1}

if I/P = 110 → 1 (O/P)
if I/P ≠ 110 → 0 (O/P)
3 states are required :
- q_0 is the starting state
- q_1 for taking 110 as input to produce 1 as O/P
- q_2 for taking non 110 as input to produce 0 as O/P

$L = \{10|\}$

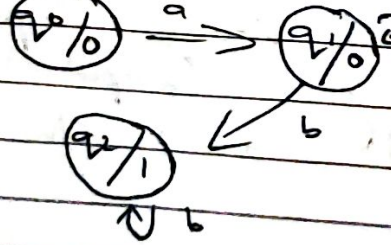


Q Construct a moore machine that takes set of all strings over $\{a,b\}$ as input and prints "1" as output for every occurence of 'ab' as a substring

→

Input = $\{a,b\}$
Output = $\{0,1\}$
If I/P ab → 1 (O/P)
If I/P ≠ ab → 0 (O/P)
3 states are required
- q_0 is the starting state
- q_1 for taking non "ab" and produce 0
- q_2 for taking "ab" and produce 1

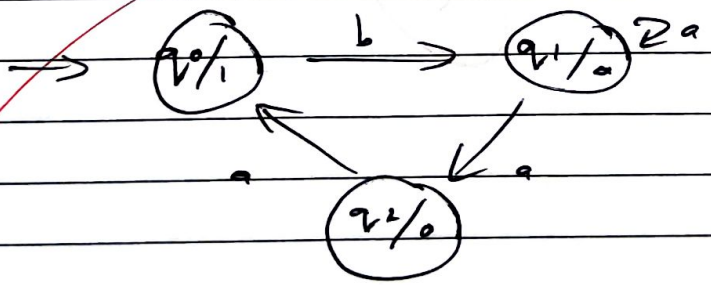
$L = \{ab, abab, aab, abb\}$



Q Construct a moore machine that takes set of all strings over {a, b} and count the no of occurence of substring 'baa'.

→ Input = {a, b}
Output = {0, 1}
If I/P = baa → 1 (O/P)
If I/P ≠ baa → 0 (O/P)
3 states are required:
- q_0 is the starting state
- q_1 for taking non 'baa' as input & generate 0 as (O/P)
- q_2 for taking 'baa' as I/P & generate as O/P

L = {baa, abaa, aabaa, bbaa, baaa}



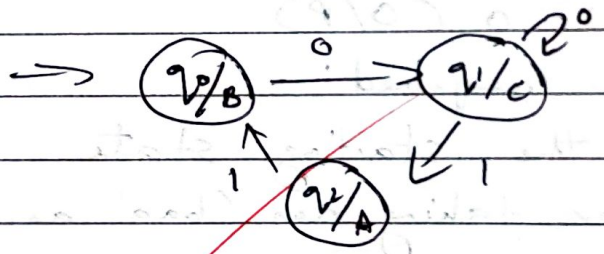
Q Construct a moore machines that takes set of all strings over {0, 1} and produces 'A' as output. If input ends with '01' or produce 'B' as output if input ends with '11' otherwise produce output as C.

→ Input = {0, 1}
Output = {A, B, C}
If I/P = '01' - A (O/P)
If O/P = '11' - B (O/P)
If I/P = random - C (O/P)

PRABHAT
PAGE NO. :
DATE : / /

3 states are required:
- q_0 being the starting state
- q_1 for taking inputs and generate A & B
- q_2 for taking inputs and generate C.

$L = \{01, 11, 10, 00\}$



17/02/2025

10