Previous Year Question & Answers

Subject: FLAT Branch-CSE Semester-5th Faculty: A.K. Rout

Long Questions And Answers:

Module-1

Q.1. Construct a DFA for the following:

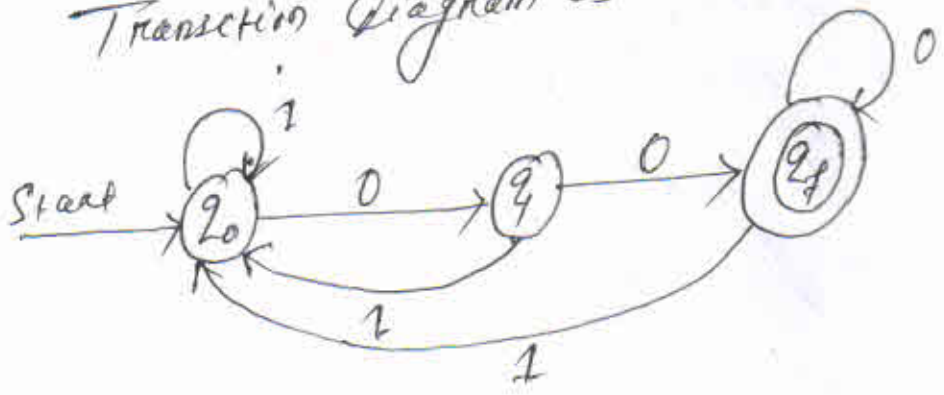
(i) The set of All String that end with 00 over $\Sigma = \{0,1\}$

(ii) The set of all string that Contain sub-string 010 over $\Sigma = \{0,1\}$

Ans: DFA $D = (Q, \Sigma, q_0, \delta, F)$.

Language $L = \{00, 000, 0000, \ldots, 100, 1100, 10100, \ldots\}$

Transition Diagram as



$Q = \{q_0, q_1, q_f\}$

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

$q_0 = \{q_0\}$

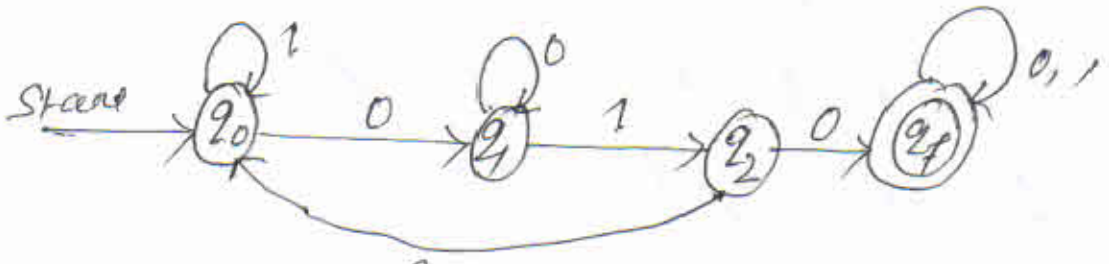
$F = \{q_f\}$.

(ii)

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

$L = \{010, 00010, 01000, 0101, 01011, 11010, \ldots\}$

Transition Diagram is



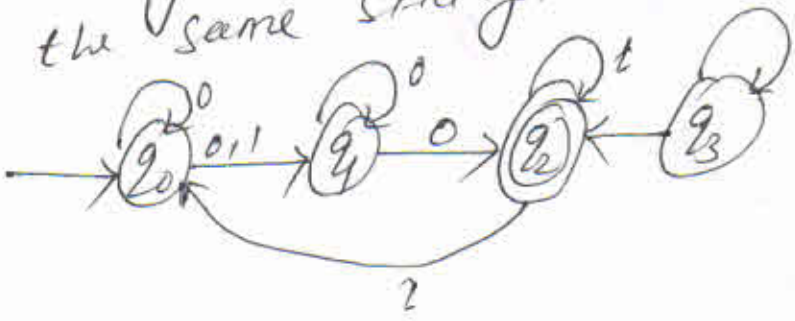
$Q = \{q_0, q_1, q_2, q_f\}$

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

$q_0 = \{q_0\}$

$F = \{q_f\}$

Q.2. Define NFA mathematically. Explain its significance and function. Convert the given FA into DFA equivalent. Explain the method used taking suitable example. Prove both accept the same string.



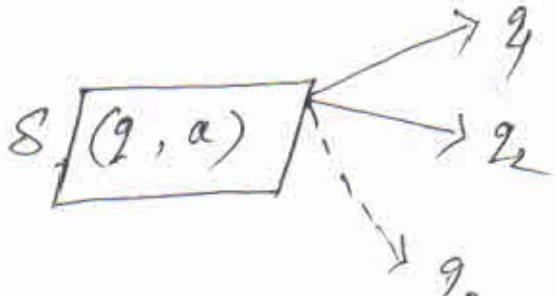
Answer: Mathematically NFA can be defined as

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

(Follow the class notes for mathematical definition)

Significance & function of NFA

→ An NFA is a machine to recognize a language.
→ It reads a string of input symbols and determines whether it is accepted or not.
→ The transition function of a NFA takes a input on no input and transit to one or more than one states



Where q is a state and 'a' is input symbol.

Conversion of NFA to DFA

Transition table of NFA is

δ / Σ	0	1
→ q_0	q_0, q_1	q_1
q_1	q_1, q_2	q_1
* q_2	ϕ	q_0, q_2
q_3	q_3	q_3

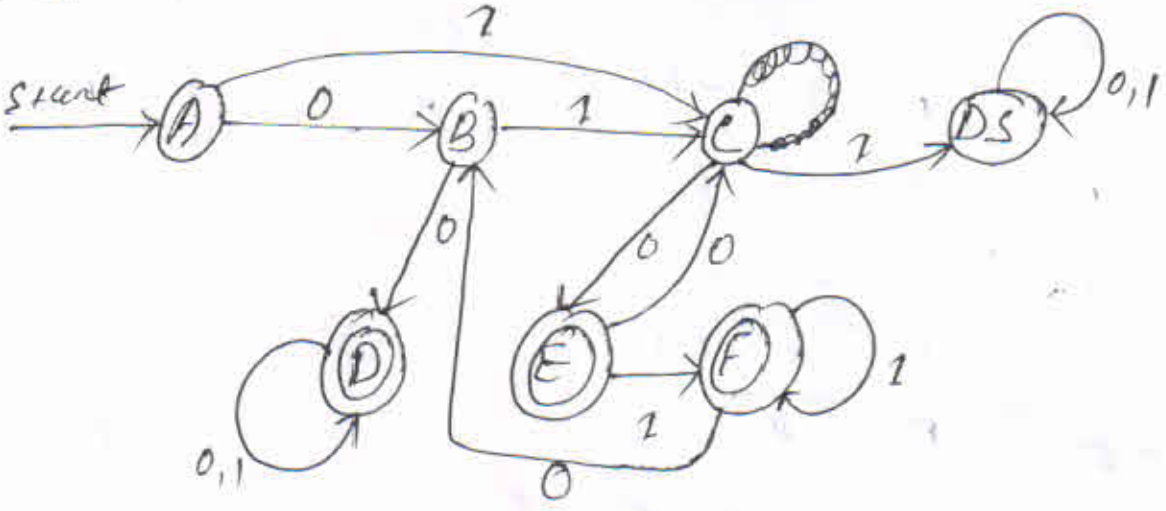
→ Treat the ϕ transition of NFA as Dead state of DFA

Now Draw the transition table of DFA by taking the transition of NFA which has more than one states with as new state.

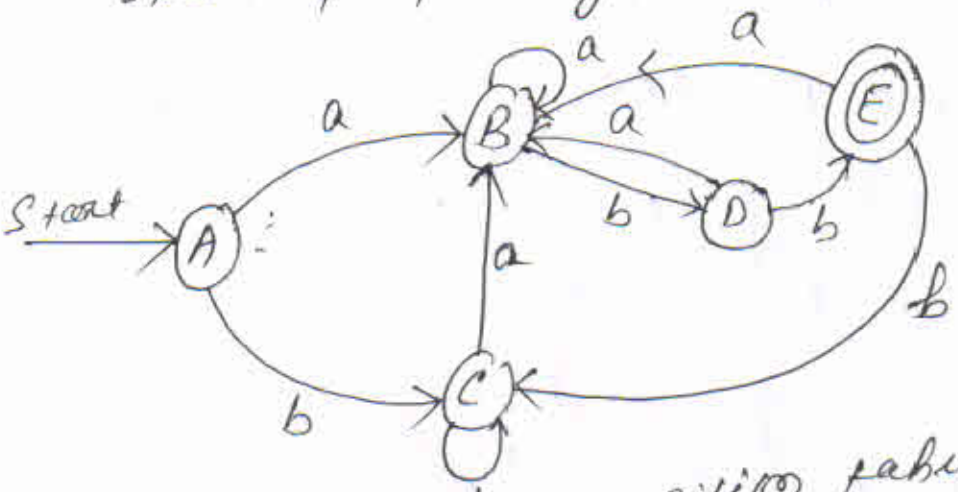
Transition Table of DFA

S / Σ	0	1	
→ q_0 (A)	$\{q_0, q_1\}$ B	$\{q_1\}$ C	$\{q_0, q_1\}$ and $\{q_1\}$ are new states
$\{q_0, q_1\}$ (B)	$\{q_0, q_1, q_2\}$ D	$\{q_1\}$ C	$\{q_0, q_1, q_2\}$ new state
$\{q_1\}$ (C)	$\{q_1, q_2\}$ E	DS	$\{q_1, q_2\}$ new state
* $\{q_0, q_1, q_2\}$ (D)	$\{q_0, q_1, q_2\}$ D	$\{q_0, q_1, q_2\}$ D	no new state
* $\{q_1, q_2\}$ (E)	$\{q_1\}$ C	$\{q_0, q_2\}$ F	$\{q_0, q_2\}$ new state
* $\{q_0, q_2\}$ (F)	$\{q_0, q_1\}$ B	$\{q_0, q_2\}$ F	no new state.
DS	DS	DS	DS end state transit within itself

→ The final states of DFA are those states which contains final state of NFA



(3) Write the algorithm to minimize the number of states of DFA. Apply this to minimize the states of following DFA.



Step-1: Draw the transition table of final and non-final states after removing unreachable, non-distinguishable and dead states.

→ There are no inaccessible states.

Table-1

δ/Σ	a	b	
→A	B	C	R1
B	B	D	R2
C	B	E	R3
D	B	E	R4
*E	B	C	

Table-2

δ/Σ	a	b
*E	B	C

Step-2: Now check the domain equivalence of each row of final and non-final states individually.

→ As there is only one final state so domain equivalence is not applicable. We apply it for only non-final states

→ In transition table $A = R_3$, so C is replaced by A if R_3 is removed or A is replaced by C if R_1 is removed.

→ Let R_3 is Removed then we get

S/Σ	a	b
→A	B	A
B	B	D
D	B	E

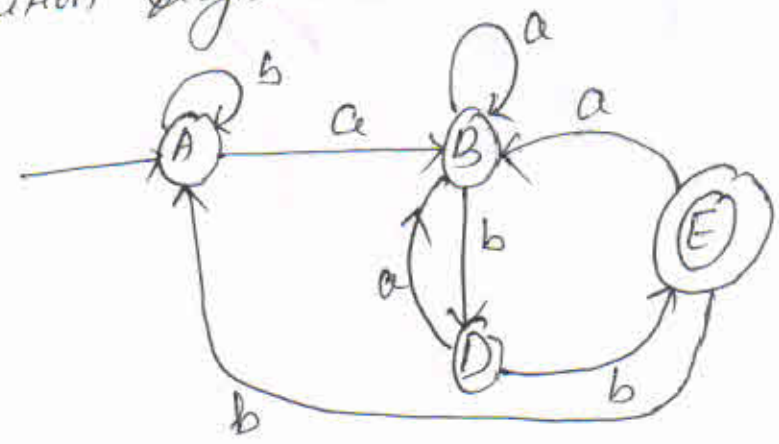
Step-3: Repeat step-2 until there is a Row equivalent.

→ As there is no Row equivalence stop

Step-4: Now combine both transition table.

S/Σ	a	b
→ A	B	A
B	B	D
D	B	E
* E	B	A

→ Transition Diagram is



(4) How an NDFA with epsilon-moves can be converted to DFA? Illustrate.

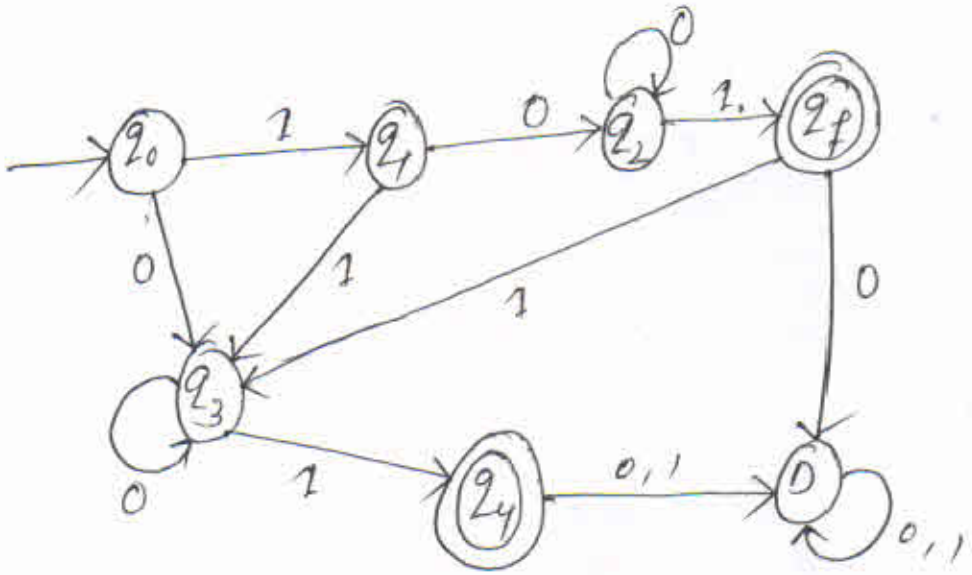
Solution: First convert NFA with ϵ to NFA without epsilon. Then follow the method to convert NFA to DFA.

NB: Follow the class notes.

(5) Construct a DFA with reduced states equivalent to re $10+(0+11)0^*1$

Solution:

Language $L = \{10(0^*)1, 00^*1, 110^*1, \ldots\ldots\}$



Q.8. what is the need to study automata theory?

Answer:

An Automata is defined as a system where energy, material and information are transformed, transmitted and used for performing some function without direct participation of man. The Study of Automata Theory is fruitful and futuristic.

There are several reason why the study of Automata and complexity is important and core part of computer science. Some important aspect of the Automata theory is given below :

1)It helps in making software for designing and checking the behavior of digital circuit.

2) The Lexical Analyzer of the typical compiler that is one of the component that breaks the input text into logical unit such as identifier, key words and punctuation.

3) Software for scanning large bodies of text such as collection of web pages in order to find out the occurrence of words, phrases or other patterns.

4) Automata theory is the key to software for verifying system of all types that have a finite number of distinct state such as communication protocol or protocol for secure exchange of information.

5) Automata theory is most useful of software for natural language processing.

Automata are incredibly useful for **applications** such as regular expressions, and learning about them makes it much easier to understand

Q.9. what is the central concepts of automata theory in toc?

Answer:

Automata Theory is a branch of computer science that deals with designing abstract self propelled computing devices that follow a predetermined sequence of operations automatically. An **automaton** with a finite number of states is called a Finite **Automaton**.

Central concepts of automata theory are alphabets ,strings ,languages

Alphabet s (Σ): Alphabets are set of symbols, which are always finite.

String: String is a finite sequence of symbols from some alphabet.

String is generally denoted as w and length of a string is denoted as |w|.

Empty string is the string with zero occurrence of symbols, represented as ε.

Number of Strings (of length 2) that can be generated over the alphabet {a, b} - - - a a, a b , b a, b b

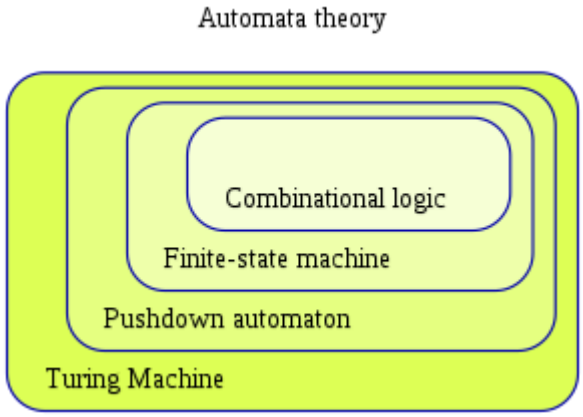
Length of String |w| = 2 Number of Strings = 4

Q.10.Explain the concept of basic machine. Also explain the properties and limitation of FSM.

Answer:

An **automaton** (**Automata** in plural) is an abstract self-propelled computing device which follows a predetermined sequence of operations automatically. An **automaton** with a finite number of states is called a Finite **Automaton** (FA) or Finite State **Machine** (FSM).

Automata theory is the study of abstract machines and automata, as well as the computational problems that can be solved using them. It is a theory in theoretical computer science.



Automata theory is closely related to formal language theory. An automaton is a finite representation of a formal language that may be an infinite set. Automata are often classified by the class of formal languages they can recognize, typically illustrated by the Chomsky hierarchy, which describes the relations between various languages and kinds of formalized logics.

capabilities of finite state machines

A Finite State Machine, or FSM, is a computation model that can be used to simulate sequential logic, or, in other words, to represent and control execution flow. Finite State Machines can be used to model problems in many fields, including mathematics, artificial intelligence, games or linguistics.

Limitations of finite state machines

- The expected character of deterministic finite state machines can be not needed in some areas like computer games.
- The implementation of huge systems using FSM is hard for managing without any idea of design.
- Not applicable for all domains.

imitations of finite state machines

- The expected character of deterministic finite state machines can be not needed in some areas like computer games.
- The implementation of huge systems using FSM is hard for managing without any idea of design.
- Not applicable for all domains.

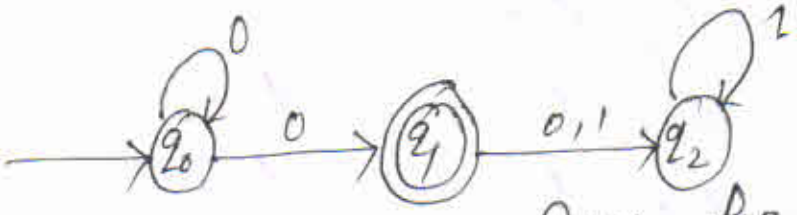
- Q.11.Explain detail about Myhill-Nerode theorem.
- Answer:see the answer from class notes.
- Q.11.Differentiate between DFA and NFA.
- Answer:

Basis of Difference	DFA	NFA
Reaction to symbols	For each symbolic representation of the alphabet, only a singular state transition can be attained in DFA.	No specifications are needed from the user with respect to how certain symbols impact the NFA.
Empty string transition	DFA is not capable of using an Empty String transition.	NFA can use an empty String transition.
Structure	DFA can be best described and understood as one machine.	NFA is like multiple small machines that are performing computational activities at the same time.
Rejection of string	DFA rejects the string in case it terminates in a state that is different from the accepting state.	NFA rejects the string in the event of all branches dying or refusing the string.
Backtracking	It is possible to use backtracking in DFA.	It is not possible to use backtracking at all times in the case of NFA.
Ease of construction	Given its complex nature, it is tougher to	NFA is more easily constructed in comparison

Basis of Difference	DFA	NFA
	construct DFA.	to DFA.
Supremacy	All DFAs are derived from NFAs.	All NFAs are not DFAs.
Transition functions	The number related to the next state is one.	The number related to the next state/ states is either zero or one.
Complexities of time	The total time required for running any input string in DFA is less than what it is in NFA.	The total time required for running any input string in NFA is larger than that in comparison to DFA.
Full form	The full form of DFA is Deterministic Finite Automata.	The full form of NFA is Nondeterministic Finite Automata (NFA).
Space requirement	More space allocation needed.	Less space needed.
The setting of the next possible set	The next possible state is clearly set in DFA.	In NFA, every single pair of input symbols and states may contain many possible next states.

-

(6) Convert the following NFA into an equivalent DFA



Solution: Follow the procedure for Question No-2.

(7) What do you mean by DFA and NDFA? Explain the Conversion of NFA to DFA.

Solu? DFA: Mathematically DFA can be defined as $D = (Q, \Sigma, q_0, F, \delta)$

where Q = non-empty set of states.

Σ = non-empty finite set of input symbols

q_0 = non-empty finite set of start symbol. $q_0 \in Q$.

F = non-empty finite set of final state $F \in Q$.

δ - δ is the transition function which takes two arguments one state and one input symbol and move to a single state

$$\boxed{\delta(q, a)} \longrightarrow q'$$

NDFA : NDFA mathematically defined as

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

Where Q = non-empty finite set of states

Σ = non-empty finite set of input symbols

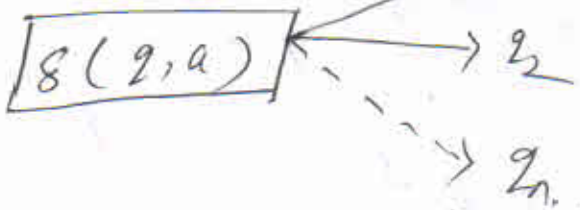
q_0 = non-empty finite set of initial state.

$q_0 \in Q$.

F = non-empty finite set of final state

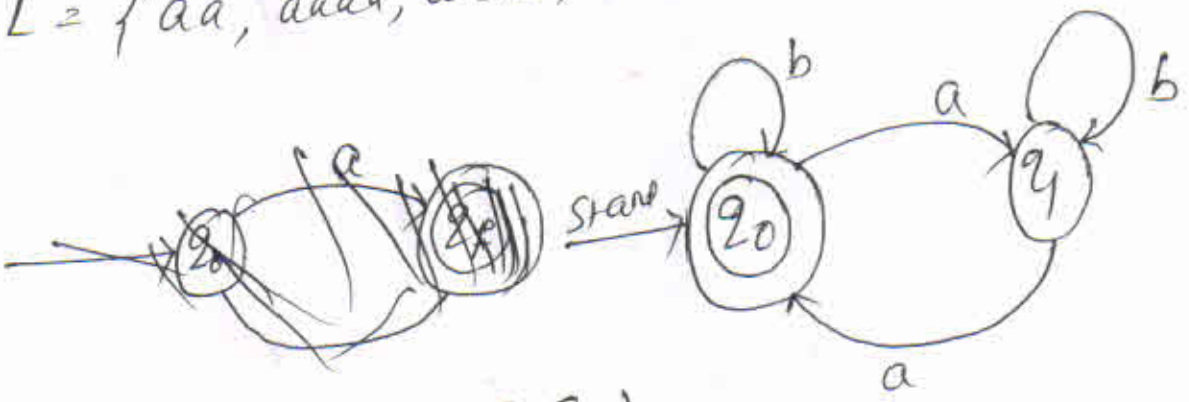
$F \in Q$

$\delta = \delta$ is the transition function which takes two arguement one state and an input or null and move to one or more than one states



Q.12. Design a DFA which accepts even number of a's over the alphabet $\{a, b\}$

Answer : $L = \{aa, aaaa, aba, abba, baa, bbaa, aabb, \ldots\}$



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

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

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

δ / Σ	a	b
q_0	q_1	q_0
q_1	q_0	q_1

Q.13. Construct a Finite Automata equivalent to the regular expression.

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

Answer: Follow the Answer of Question no. 5.

Q.14. For the following non-deterministic Finite Automata, make equivalent deterministic Finite automata

	a	b
$\rightarrow q_0$	q_0, q_1	q_2
q_1	q_0	q_1
q_2	–	q_0, q_1

Answer: The transition diagram for NFA is wrong question as there is no final state.