Assignment No. 01

1/07/23

Q.1 Explain Basic terms of automata theory.

→ 1) Alphabets :
- An alphabet is any finite set of symbols.

example :
- $\Sigma = \{a, b, c, d\}$ is an alphabet set where 'a', 'b', 'c', and 'd' are symbols.

2) String :
- A string is a finite sequence of symbols taken from Σ.

example :
- 'cabcad' it is a valid string on the alphabet set $\Sigma = \{a, b, c, d\}$.

3) Length of String :
- It is the number of symbols present in string. (denoted by $|s|$).

example :
- if s = 'abcd' then $|s| = 4$
- if $|s| = 0$ it is called as an empty string (denoted by ε).

4) Language :
- A language is a subset of Σ^* for some alphabet Σ. It can be finite or infinite.

example :
- If the language takes all possible strings of length 2 over $\Sigma = \{a, b\}$.

then $L = \{ab, aa, ba, bb\}$.

classmate

Date

Page

5) kleene Star :

- The kleene star Σ^* is a unary operator on set of symbols or strings. Σ gives infinite set of all possible strings of all possible length over Σ including empty string.

- Representation :

$\Sigma^* = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup$ .... when Σ_p is the set of all possible strings of length of p.

example :

- if $\Sigma = \{a, b\}$, $\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb \ldots\ldots\}$

6) kleene closure / Plus :

- The kleene plus $\Sigma+$ is the infinite set of all possible strings of all possible length over Σ excluding empty string.

Representation :

- $\Sigma^+ = \Sigma^1 \cup \Sigma 2 \cup \Sigma 3 \cup$ .... where Σp is the set of all possible strings of length of p.

example :

if $\Sigma = \{a, b\}$, $\Sigma^+ = \{a, b, aa, ab, ba, bb \ldots\ldots)$

Q.2 Define Finite State Machine.

→ • - FSM is a calculation model that can be executed with the help of hardware otherwise software.

• - This is used for creating sequential logic as well as a few computer programs.

• - FSMs are used to solve the problems in fields like mathematics, games, linguistics, and artificial intelligence.

• - In a system where specific inputs can cause specific changes in state that can be signified with the help of FSMs.

- Definition :
- An automaton with finite number of states is called a Finite Automaton (FA) or Finite state Machine (FSM).

- An automaton can be represented by 5 tuple ($Q, \Sigma, \delta, q_0, F$) where Q is a finite set of states.
- Q is a finite set of states.
- Σ is a finite set of symbols called the alphabet of the automaton.
- δ is transition function.
- q_0 is the initial state from where any input is processed ($q_0 \in Q$)
- F is a set of finite final state / states of Q ($F \subseteq Q$).

Q.3 Write a short note on construction DFA from a given language.

→ - The language consisting of strings starting with a particular substring.

- Following steps are constructing DFA from given language :

1) Calculate minimum number of DFA states. If length of substring is n then number of states are $n+2$.
2) Decides the strings for which DFA will construct.
3) Always prefer existing path.
4) Send all left possible combination to the dead state
5) Do not send all left possible combination over start state

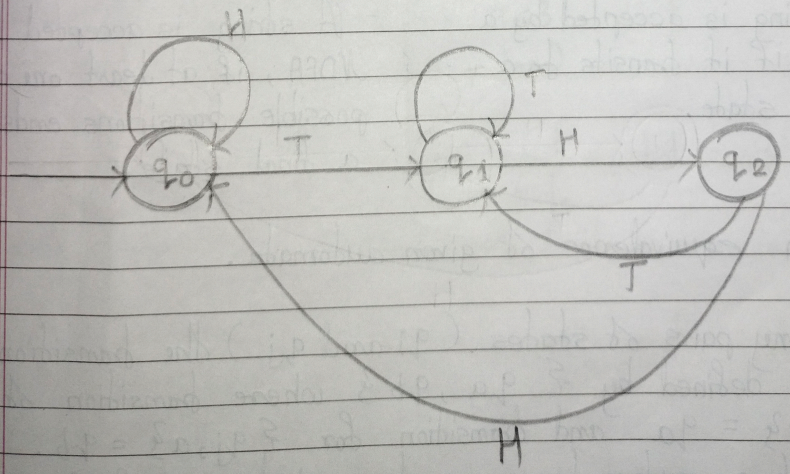
Example :

Draw a DFA for the language accepting strings ending with 'TH' over input alphabet $\Sigma = \{T, H\}$.

Solution :
step 1 : All strings of the language ends with substring "TH".
so, length of substring = 2
Thus, Minimum number of states required in the DFA = 2 + 1 = 3 ($\because$ DFA = n + 1)

step 2 : We will construct DFA for the following strings-

TH
TTH
THTH



Q.4 How DFA is differ from NDFA ?

→

DFA	NDFA
- The trasition from a state is to a single particular next state for each input symbol. Hence it is called deterministic.	- The transition from a state can be to multiple next state for each input symbol. Hence it is called non-deterministic.
- Empty string trasitions are not seen in DFA.	- NDFA permits empty string transition.
- Backtracking is allowed in DFA.	- In NDFA, backtracking is not allways possible.
- Requires more space.	- Require less space.
- A string is accepted by a DFA. if it transits to a final state.	- A string is accepted by a NDFA, if at least one of all possible transitions ends in a final state.

Q.5 Explain equivalence of given automata.

→ - For any pairs of states (q_i and q_j) the transition for input defined by $\{q_a, q_b\}$ where transition of $\{q_i, a\} = q_a$ and transition for $\{q_j, a\} = q_b$.
- The two automata are not equivalence if for a pair $\{q_a, q_b\}$ one is intermediate state and other state is final state otherwise it is equivalence.

Diagram A

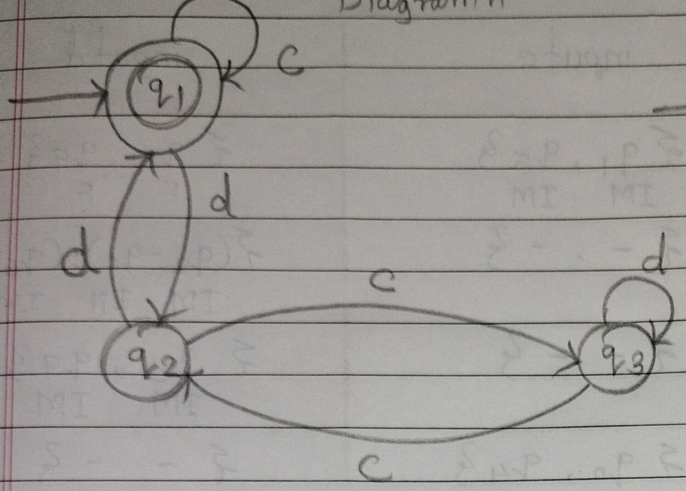
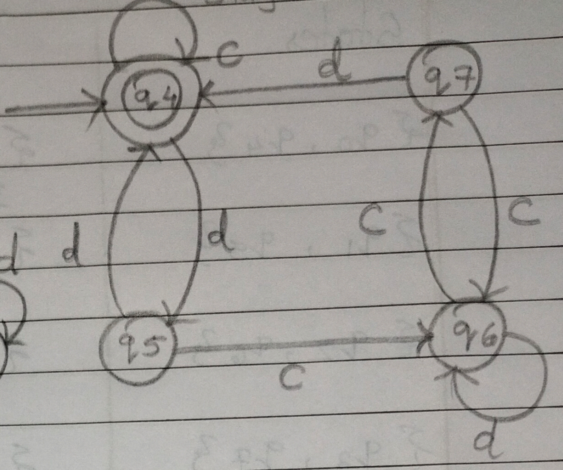


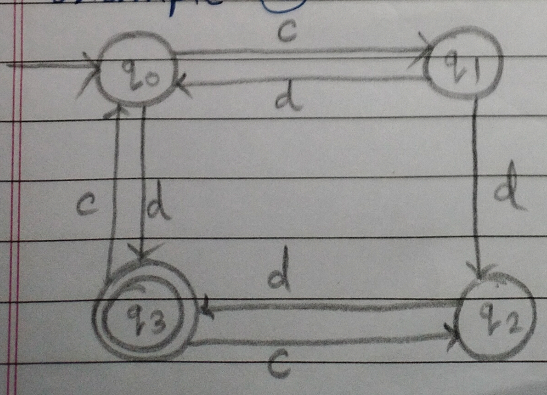
Diagram B

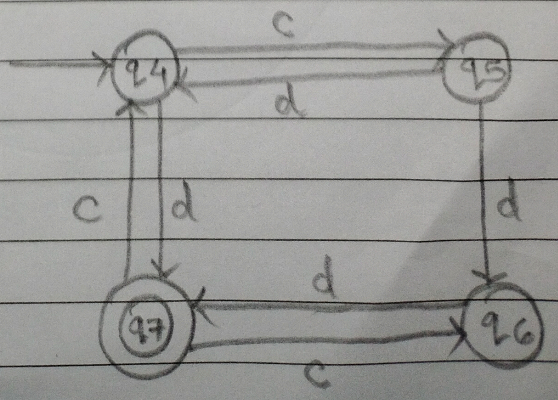


States	Input c	Input d
$\{q_1, q_4\}$	$\{q_1, q_4\}$ (final , final)	$\{q_2, q_5\}$ (Im , Im)
$\{q_2, q_5\}$	$\{q_3, q_6\}$ (Im, Im)	$\{q_1, q_4\}$ (Im , Im)
$\{q_3, q_6\}$	$\{q_2, q_7\}$ (Im, Im)	$\{q_3, q_5\}$ (Im , Im)
$\{q_2, q_7\}$	$\{q_3, q_6\}$ Im , Im)	$\{q_1, q_4)$ (final , final)

In above table for each pair input symbol c and got equal states like final or intermediate.
So we can say that both finite automata are equivalence.

Example ② :





States	input c	input d
$\{q_0, q_4\}$	$\{q_1, q_5\}$ IM IM	$\{q_3, q_7\}$ F F
$\{q_1, q_5\}$	$\{-, -\}$	$\{(q_2 - q_0), (q_6, q_4)\}$ IM IN IM IN
$\{q_2, q_6\}$	$\{- -\}$	$\{q_3, q_7\}$ IM IM
$\{q_3, q_7\}$	$\{q_0, q_4\}$ IN IN	$\{- -\}$