Name : Fardouse Lomat Jahan Rumpa

ID : 2219170041

Dept : B. Sc in CSE

Course title : Theory of computing

Course code : CSI 317

Semester : Summer 2023

Ans to the Qus NO: 01 (a)

Ans: Automata:

Theory of automata is a theoretical branch of computer science and mathmatical. It is the study of abstract machines and the computations problems that can be sloved using these machines. The abstract machine is called the automata. The main motivation behind developing the automata theory was to develop methods to describe and analysis the dynamic behaviour of discrete systems.

Automata is kind of machine which takes some string as input and this input goes through a finite number of states and may enter in the final state.

Ans to the Qus NO: 01 (b)

Ans: DFA: A Determinitic finite automata is a five-tuple automata. Following is the definition of DFA –

$M = (Q, \Sigma, \delta, q0, F)$

NFA: NFA also has five states same as DFA, but with different transtion function, as shown follows-

$$\delta : Q \times \Sigma \rightarrow 2^{Q}$$

Differences: The major difference between the DFA and the NFA are as follows—

Deterministic finite Automata	Non-Deterministic finite Automata
① Each transition leads to exatly one state called as deterministic	① A transition leads to a subset of states i.e. some transitions can be non-deterministic.
② Accepts input if the last state is in final	② Accepts input if one of the last states is in final.
③ Backtracking is allowed in DFA.	③ Backtracking is not always possible.
④ Requires more space.	④ Require less space

Ans to the Qus NO: 01 (c)

Ans: Turning machines are theoretical computal devices introduced by Alan Turing in 1936. They Consists of an infinite tape, a read/write head, and a set of rules that dictate the machine actions based on its current state and the Symbol being read. Turing machines can actions based on. Turing machine can Simulate any algorithm process..

Ans to the Qus NO: 02(a)

Ans: The pigeonhole principle states that if you try to distribute "pigeons" into "pigeonholes" and you have more pigeons than pigeonholes, at least one pigeonhole must contain more than one pigeon. The Letterbox principle is a related concept in Combina-tories, stating that if you place n+1 objects into n boxes, than at least one box must contain more than one object. Both principles deal with counting and the inevitability of repetitions in certain scenaries.

Ans to the Qus NO: 02(b)

Ans: The notation "$M = (Q, \Sigma, \delta, q0, F)$" represents the formal description of a turing machine, where each element has a specific meaning.

① Q: This represents the finite set of states that the turing machine can be in.

② Σ: This is the input alphabet, which consists of symbols that the Turing machine reads from the input tape.

.

③ δ : This is the transition function, which specifies how the turing machine transitions between states based on the current state. the symbol read from the tape, and the symb to be written on the tape, and

④ q^0 : This is the initial state of the turing machine.

⑤ F : This is the set of final or accepting states, indicating when the turing machine should hal and accept the input.

So, in the given notation " $M = (Q, \Sigma, \delta, q^0, F)$

→ Q repreents' the set of state.
→ Σ is the input alphabet.
→ q^0 is the transition function
→ F repprents the set of final states.

This notation is a concise and standardized way to describe the essential components and behaviour of a turing machi

Ans to the Qus NO: 02 (C)

Ans: state diagram & state table of "aabaab"

Here's the state diagram for it:

Creating transition table for M: state a b q0 q0 q' q1 q1 q1

$$q0 \xrightarrow{a} q1 \xrightarrow{a} q2 \xrightarrow{b} q3 \xrightarrow{a} q4 \xrightarrow{a} \xrightarrow{b} q5$$

in this diagram:

→ q0 is the start state.

→ q1, q2, q3, q4 and q5 are states.

representing the progression of the string "aabaab"

→ the transition are labeled with the input characters "a" or "b".

The string "aabaab" is accepted by transitioning from q0 to q1 with "a". then to q2 with "a". then to q5 with "a". and finally, the string is fully processed.

this state diagram specifically represents the language generated by the "aabaab".

State table of "aabaab"

current state	input	Next state
q^0	a	q^1
q^1	a	q^2
q^2	b	q^3
q^3	a	q^4
q^4	a	q^5
q^5	—	Reject

in this state table:

→ The current state column represents the current state of the automation.

→ The "input" column repprents the input symbol being read ("a" or "b")

→ The "Next state" Column repprents the state to transition to based on the current state and input.

The table indicates how the automation process the input string "aabaab".

Ans to the Qus NO: 03 (a)

Ans: Applications of DFA:

→ Vending Machines.
→ Traffic Lights
→ Video Games.
→ Text parsing.
→ Regular Expression Matching.
→ CPU Controllers.
→ Protocol Analysis.
→ Natural Language Processing
→ Speech Recognition.

Ans to the Qus NO: 03(c)

Ans: The statement that every integer "a" can be writte as "a = 2km" is not true for all integers. It appears t be attempting to express that every integer "a" can be factored into the product of an odd number "k" and a power of 2("m"), but this is not accurate.

In fact, the statement is vaild only for even integers. If "a" is an even integer, then it can indeed be expressed in the form "a = 2km". where " is an odd integer an "m" is a positive integer. This is because even integers are divisible by and they can be factored as such.

Ans to Qus NO: 03(b)

Ans: Union theorem:

The union theorem is a result from the 60s in computational complexity theory. It was published in 1969 by Ed Mccreight and Albert meyer.

Originally, it is stated for generated Blum complexity classes, but it is most relevant for DTIME, NTIME, DSPACE or NSPACE as stated in first edition from 1979 of the textbook of Hopcroft and ullman. This chapter was removed from newer editions, however,

The theorem for time complexity roughly states the following - Given a list of monotonically increasing time bound functions $t_1, t_2 \ldots$ where $t_{i+1} > t_i$ for $i \in N_{>0}$, there exists a time bound function t such that a problem is computable in time bounded by t if and only if there exists an i such that that problem is computable in time bounded by t_i.

The theorem can be applied to show that complexity classes like P are well-defined

Proof Union-theorem:

The automation $M_3 = (Q_3, \Sigma, \delta_3, q_3, F_3)$ runs M_1 and M_2 in "parallel" on a string w.

In the end, the final state (r_1, r_2) "knows" if $w \in L_1$ (via $r_1 \in F_1$?) and if $w \in L_2$ (via $r_2 \in F_2$?)

The accepting states F_3 of M_3 are such that $w \in L(M_3)$ if and only if $w \in L_1$, or $w \in L_2$ for: $F_3 = \{(r_1, r_2) \mid r_1 \in F_1 \text{ or } r_2 \in F_2\}$.