Biyani Institute of Science & management
First Internal Examination 2019-20
MCA IIISem
Paper :-Theory of Computation (Set A)

Duration 1 hr 30 mins Max Marks 20

1. What do you understand by the term “graph”? (1)
   A **graph** is a pictorial representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are represented by points termed as vertices, and the links that connect the vertices are called edges.
2. Define “Depth of a tree”. (1)
   The height of a node is the number of edges on the longest path between that node and a leaf. The height of a **tree** is the height of its root node.
   The **depth** of a node is the number of edges from the **tree's** root node to the node.
3. Explain “Path” and “circuits” in graph with example. (3)
   Path is a route along edges that start at a vertex and end at a vertex. Circuit is a path that begins and ends at the same vertex
   Usually a path in general is same as a walk which is just a sequence of vertices such that adjacent vertices are connected by edges. Think of it as just traveling around a graph along the edges with no restrictions.
   Some books, however, refer to a path as a "simple" path. In that case when we say a path we mean that **no vertices are repeated**. We do not travel to the same vertex twice (or more).
   A **cycle** is a **closed** path. That is, we start and end at the same vertex. In the middle, we do not travel to any vertex twice.
   It will be convenient to define trails before moving on to circuits. Trails refer to a walk where **no edge is repeated**. (Observe the difference between a trail and a simple path)
   Circuits refer to the **closed** trails, meaning we start and end at the same vertex.

4. Define “Degree of a vertex” with example. (3)
   In graph theory, the **degree** (or **valency**) of a vertex of a graph is the number of edges incident to the vertex, with loops counted twice.[1] The degree of a vertex is denoted or . The **maximum degree** of a graph *G*, denoted by Δ(*G*), and the **minimum degree** of a graph, denoted by δ(*G*), are the maximum and minimum degree of its vertices. In the graph on the right, the maximum degree is 5 and the minimum degree is 0. In a regular graph, all degrees are the same, and so we can speak of *the* degree of the graph.
5. Define “Non Deterministic Finite Automaton” with example. (6)
   In NDFA, for a particular input symbol, the machine can move to any combination of the states in the machine. In other words, the exact state to which the machine moves cannot be determined. Hence, it is called **Non-deterministic Automaton**. As it has finite number of states, the machine is called **Non-deterministic Finite Machine** or **Non-deterministic Finite Automaton**.
   Formal Definition of an NDFA

An NDFA can be represented by a 5-tuple (Q, ∑, δ, q_0, F) where –

- **Q** is a finite set of states.
- **∑** is a finite set of symbols called the alphabets.
- **δ** is the transition function where δ: Q × ∑ → 2^Q
  (Here the power set of Q (2^Q) has been taken because in case of NDFA, from a state, transition can occur to any combination of Q states)
- **q_0** is the initial state from where any input is processed ($q_0 \in Q$).
- **F** is a set of final state/states of Q ($F \subseteq Q$).

Graphical Representation of an NDFA: (same as DFA)

An NDFA is represented by digraphs called state diagram.

- The vertices represent the states.
- The arcs labeled with an input alphabet show the transitions.
- The initial state is denoted by an empty single incoming arc.
- The final state is indicated by double circles.

Example

Let a non-deterministic finite automaton be →

- Q = {a, b, c}
- ∑ = {0, 1}
- q_0 = {a}
- F = {c}

6. Construct a deterministic finite automaton equivalent to (6)
   M = ({q0, q1, q2, q3}, {0, 1}, δ, q0, {q3})
   Where δ is given by the following table

State	a	b
→q0	q0, q1	q0
q1	q2	q1
q2	q3	q3
(q3)		q2

Let $Q = \{q_0, q_1, q_2, q_3\}$. Then the deterministic automaton M_1 equivalent to M is given by

$$M_1 = (2^Q, \{a, b\}, \delta, [q_0], F)$$

where F consists of:

$$[q_3], [q_0, q_3], [q_1, q_3], [q_2, q_3], [q_0, q_1, q_3], [q_0, q_2, q_3], [q_1, q_2, q_3]$$

and

$$[q_0, q_1, q_2, q_3]$$

and where δ is defined by the state table given by Table 3.7.

TABLE 3.7 State Table of M_1 for Example 3.8

State/Σ	a	b
$[q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$
$[q_0, q_1, q_3]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$

BISMA
First Internal Examination 2019-20
MCA IIISem
Paper :- Theory of Computation (Set B)

Duration 1 hr 30 mins — Max Marks 20

1. What do you understand by the term “tree”? (1)
   A tree data structure can be defined recursively (locally) as a collection of nodes (starting at a root node), where each node is a data structure consisting of a value, together with a list of references to nodes (the "children"), with the constraints that no reference is duplicated, and none points to the root.
2. Write any two properties of a tree. (1)
   Alternatively, a tree can be defined abstractly as a whole (globally) as an ordered tree, with a value assigned to each node. Both these perspectives are useful: while a tree can be analyzed mathematically as a whole, when actually represented as a data structure it is usually represented and worked with separately by node (rather than as a set of nodes and an adjacency list of edges between nodes, as one may represent a digraph, for instance).
3. Define the following terms w.r.t. a tree:- (3)
   i) Length of a path
      A path is a trail in which all vertices (except possibly the first and last) are distinct. A trail is a walk in which all edges are distinct. A walk of length in a graph is an alternating sequence of vertices and edges, , which begins and ends with vertices. The no. Of vertices in a path is its length.
   ii) Height of a tree
      The **height** of a node is the number of edges on the longest path between that node and a leaf. The **height of a tree** is the **height** of its root node. The depth of a node is the number of edges from the**tree's** root node to the node. A forest is a set of n ≥ 0 disjoint **trees**.
4. Define concatenation of two strings with example. (3)
   In formal language theory and computer programming, **string concatenation** is the operation of joining character stringsend-to-end. For example, the concatenation of "snow" and "ball" is "snowball". In some but not all formalisations of concatenation theory, also called string theory, string concatenation is a primitive notion.
5. Define “Transition System” with example. (6)
   A transition graph or a transition system is a finite directed labelled graph inwhich each vertex (or node) represents a state and the directed edges indicate the transition of a state and the edges are labelled with input J output.
   A typical transition system is shown in Fig. 3.5. In the figure, the initialstate is represented by a circle with an arrow pointing towards it, the final state by two concentric circles, and the other states are represented by just a circle.
   The edges are labelled by input/output (e.g. by 1/0 or 1/1). For example, if thesystem is in the state *qo*and the input 1 is applied, the system moves to

state *qJ*as there is a directed edge from *qo*to *qJ*with label 1/0. It outputs O.

6. Find a deterministic acceptor equivalent to (6)
M = ({q0, q1, q2}, {a,b}, δ, q0, {q2})
Where δ is given by the following table

State	a	b
→q0	q0, q1	q2
q1	q0	q1
(q2)		q0, q1

The deterministic automaton M_1 equivalent to M is defined as follows:

$$M_1 = (2^Q, \{a, b\}, \delta, [q_0], F')$$

where

$$F = \{[q_2], [q_0, q_2], [q_1, q_2], [q_0, q_1, q_2]\}$$

We start the construction by considering $[q_0]$ first. We get $[q_2]$ and $[q_0, q_1]$. Then we construct δ for $[q_2]$ and $[q_0, q_1]$. $[q_1, q_2]$ is a new state appearing under the input columns. After constructing δ for $[q_1, q_2]$, we do not get any new states and so we terminate the construction of δ. The state table is given by Table 3.5.

TABLE 3.5 State Table of M_1 for Example 3.7

State/Σ	a	b
$[q_0]$	$[q_0, q_1]$	$[q_2]$
$[q_2]$	$\emptyset$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_0]$	$[q_0, q_1]$