Govt. Engg. College Jhalawar

Sub:- TOC MM- 10 Hr.-01

Q1. What is automation? Differentiate the DFA and NDFA. (3)

Q2. Prove that for any transition function and for ant two input strings x and y, $\delta(q,xy)=\delta(\delta(q,x),y)$. (2)

Q3. Construct a grammar G accepting the set L of all srtings over { a,b } having more a's than b's. (2)

Q4. Construct a nondeterministic finite automaton accepting {ab,ba}, and use it to find a deterministic automaton accepting the same set. (3)

Scanned by CamScanner

Govt. Engg. College, Jhalawar

Subject :- TOC

Q1. What is automation? Differentiate the DFA and NDFA.

Ans An Automation is defined as a system where energy, materials and information are transformed, transmitted and used for performing some operation/functions without direct participation of man. Examples are automation is automatic machine tools, automatic parking machines etc.

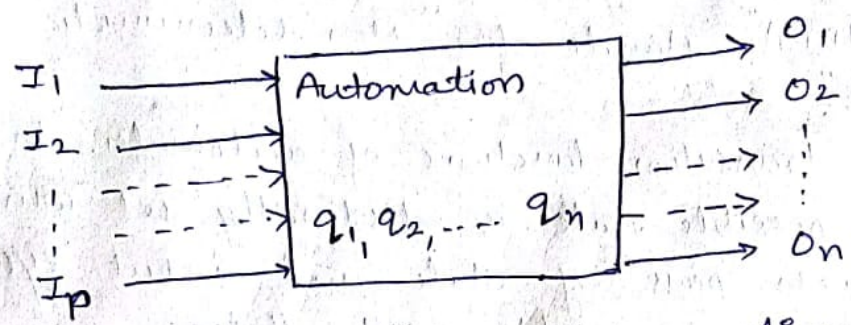


Fig:- Model of discrete automation.

The characteristics of automation are now described.

(i) I/P (Input) :- At each of the discrete instants of time $t_1, t_2 \ldots t_n$ the input values $I_1, I_2, \ldots I_p$, each of which can take a finite number of fixed values from the input alphabet Σ, are applied to the input side of the model.

(ii) Output (O/P) :- $O_1, O_2, \ldots O_n$ are the output of the model, each of which can take a finite number of fixed values from an output O.

(iii) State :- At any instant of time the automation can be in one of the state $q_1, q_2, \ldots q_n$.

(iv) State Relation :- The next state of an automation at any time time is determined by the present state and the present Input.

Scanned by CamScanner

(v) output Relation :- The output is related to the either state only or to both the input and the state. It should be noted that at any instant of time the automation is in some state. On reading an input symbol, the automation moves to a next state which is given by the state relation.

Difference b/w DFA and NDFA.

(1) "DFA" stands for "Deterministic finite automata" while "NDFA" stands for "Non deterministic finite automata".

(2) Both the transition functions of automata. In DFA the next possible state is distinctly set while in NDFA each pair of state and input symbol can have many possible next states.

(3) DFA cannot use empty string state while NDFA can use empty string transition.

(4) NDFA is easier to construct while it is more difficult to construct DFA.

(5) Backtracking is allowing in DFA while in NDFA it may or may not be allowed.

(6) DFA requires more space while NDFA requires less space.

(7) While DFA can be understand one machine and a DFA machine can be constructed for every I/P and O/P.

(8) NDFA can be understood as ~~one machine and a DFA machine can be~~ several little machines that computer together, and there is no possibility of constructing an NDFA machine for every input and output.

Scanned by CamScanner

Q2 prove that for any transition function δ and for any two input string x and y, $\delta(q, xy) = \delta(\delta(q,x)\, y)$.

Ans By the Method of Induction on $|y|$, i.e. length of y.

Basis: When $|y| = 1$, $y = a = \Sigma$

$$L.H.S = \delta(q, xa)$$
$$= \delta(\delta(q,x), a)$$
$$= R.H.S.$$

Assume the result for all strings x and string y with $|y| = n$. Let y be a string of length $n+1$ Write $y = y_1 a$ where $|y_1| = n$.

$$L.H.S. = \delta(q, xy_1a) = \delta(q, x_1 a), \quad x_1 = xy_1$$
$$= \delta(\delta(q, x_1), a)$$
$$= \delta(\delta(q, xy_1), a)$$
$$= \delta(\delta(\delta(q,x), y_1), a)$$

$$R.H.S. = \delta(\delta(q,x), y_1 a)$$
$$= \delta(\delta(\delta(q,x), y_1), a)$$

$L.H.S = R.H.S$ Hence proved.

Q3 Construct a grammar G accepting the set L of all strings over $\{a, b\}$ having more a's than b's.

Ans for generating strings with a's, but with no b's.

$$S \to a \mid as \mid sa$$

for generating strings with a's and b's (more a than b's)

$$S \to bSS \mid SbS \mid SSb.$$

So, G as follow

$G = (\{S\}, \{a, b\}, P, S)$ where P consists of

$S \to a \mid as \mid sa \mid bSS \mid SbS \mid SSb$

G accept all the strings over $\{a, b\}$ having more a's than b's.

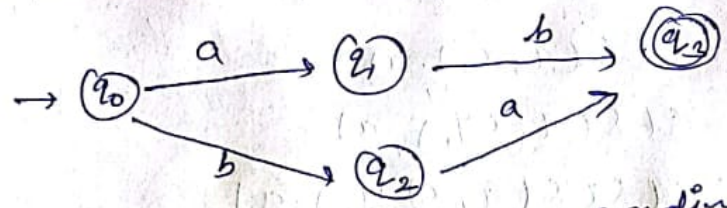
Scanned by CamScanner

Q4 Construct a NDFA accepting {ab, ba} and use it to find a deterministic automation (DFA) accepting the same set.

Ans The state of NDFA accepting {ab, ba} is defined by the table

State/Σ	a	b
q_0	q_1	q_2
q_1	—	q_3
q_2	q_3	—
(q_3)	—	—

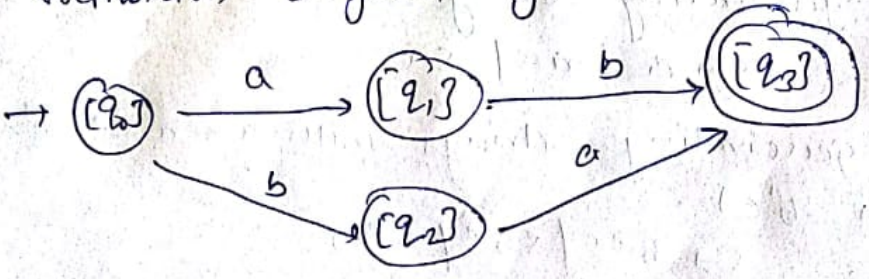
The transition diagram of NDFA



The state table of the corresponding DFA is defined by table

State	a	b
$[q_0]$	$[q_1]$	$[q_2]$
$[q_1]$	ϕ	$[q_3]$
$[q_2]$	$[q_3]$	ϕ
$[q_3]$	ϕ	ϕ
ϕ	ϕ	ϕ

The transition diagram of DFA



Scanned by CamScanner