CS & IT ENGINEERING

Theory of Computation

Regular Expression

Lecture No.- 01

By- Venkat sir

Recap of Previous Lecture

Topic

NFA

Topic

Topics to be Covered

Topic Finite Automaton & Regular Languages.

Topic Pushdown Automata & Context free Languages.

Topic Turing Machine & Recursive Enumerable Languages.

Topic Undecidability.

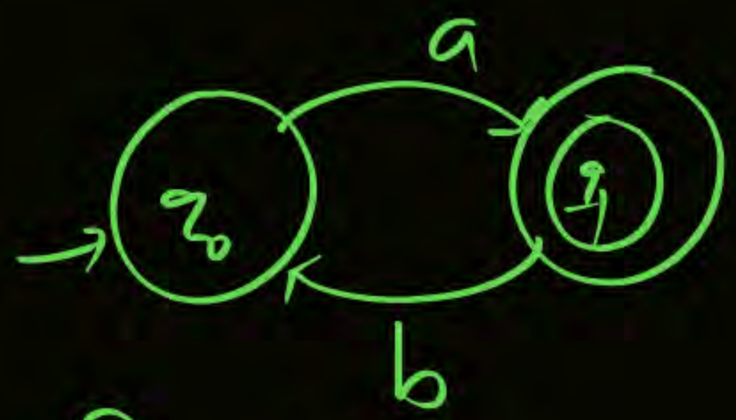
NFA is formally define as:- $\boxed{\{Q, \Sigma, q_0, F, \delta\}}$

DFA $Q \times \Sigma \to Q$

NFA $Q \times \Sigma \to 2^Q$

Q	-	Finite number of states (set of state)
Σ	-	Input alphabet
q_0	-	initial state
F	-	Set of final states
δ	-	transition function

$$\boxed{Q \times \Sigma \to 2^Q}$$



1 → Every NFA is DFA? → <u>false</u>

2 → Every DFA is NFA → <u>True</u>

Topic : Non-Deterministic Finite Automaton

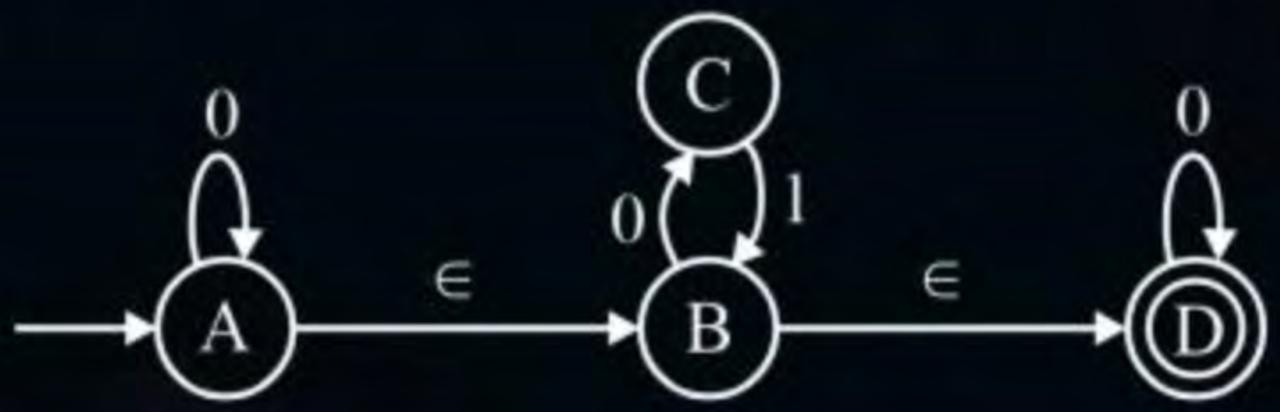
- Construction of NFA is easy than DFA.
- Minimization not possible for NFA
- Complementation not possible for NFA
- NFA from the given state on the given input string multiple state possibility may be exist.
- Language recognization is easy in DFA compare to NFA.
- In NFA, for valid string also automata may halt in non-final state.
- In NFA for the valid even though multiple non-final transition exist for one final state transition should exit.
- All DFA are NFA but all NFA need not be DFA.

Expressive power

alg

DFA = NFA

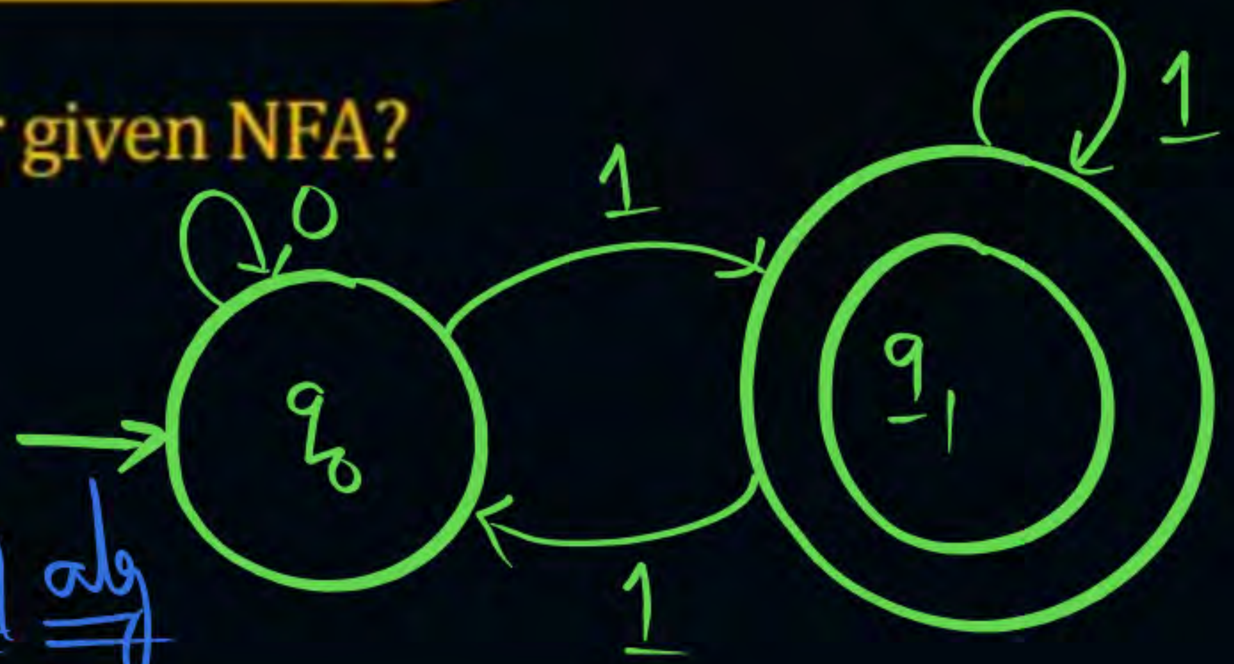
#Q. Construct an equivalent DFA for the following Ɛ-NFA



Topic : NFA to DFA Conversion

what is the equivalent DFA for given NFA?

NFA



Subset Construction alg

① Transition table NFA

	0	1
q_0	q_0	q_1
q_1	—	$\{q_0, q_1\}$

NFA

2

	0	1
$\rightarrow q_0$	q_0	q_1
q_1	$\_$	$\{q_0, q_1\}$

$$\{q_1\} \cup \{q_0, q_1\} = \{q_0, q_1\}$$

DFA

4

	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_{Dead}	$[q_0 q_1]$
q_{Dead}	q_{Dead}	q_{Dead}
$[q_0 q_1]$	q_0	$[q_0 q_1]$

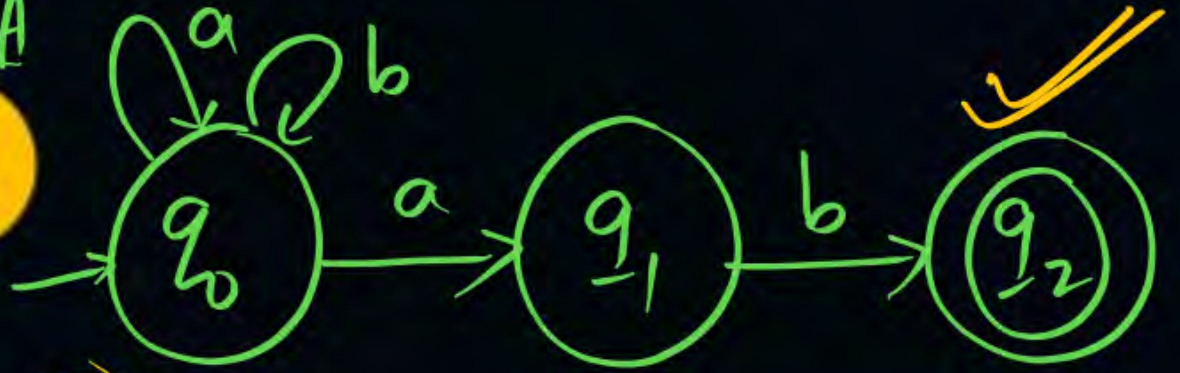
① Initial State is Same

② for no transition in NFA, Dead state is taken in DFA.

NFA		DFA
zero	$\longrightarrow$	Dead state
multiple	$\longrightarrow$	new state

Topic : NFA to DFA Conversion

NFA



$\{q_0, q_1\} \cup \{\phi\} = \{q_0 q_1\}$

NFA

3

	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0
q_1	—	q_2
q_2	—	—

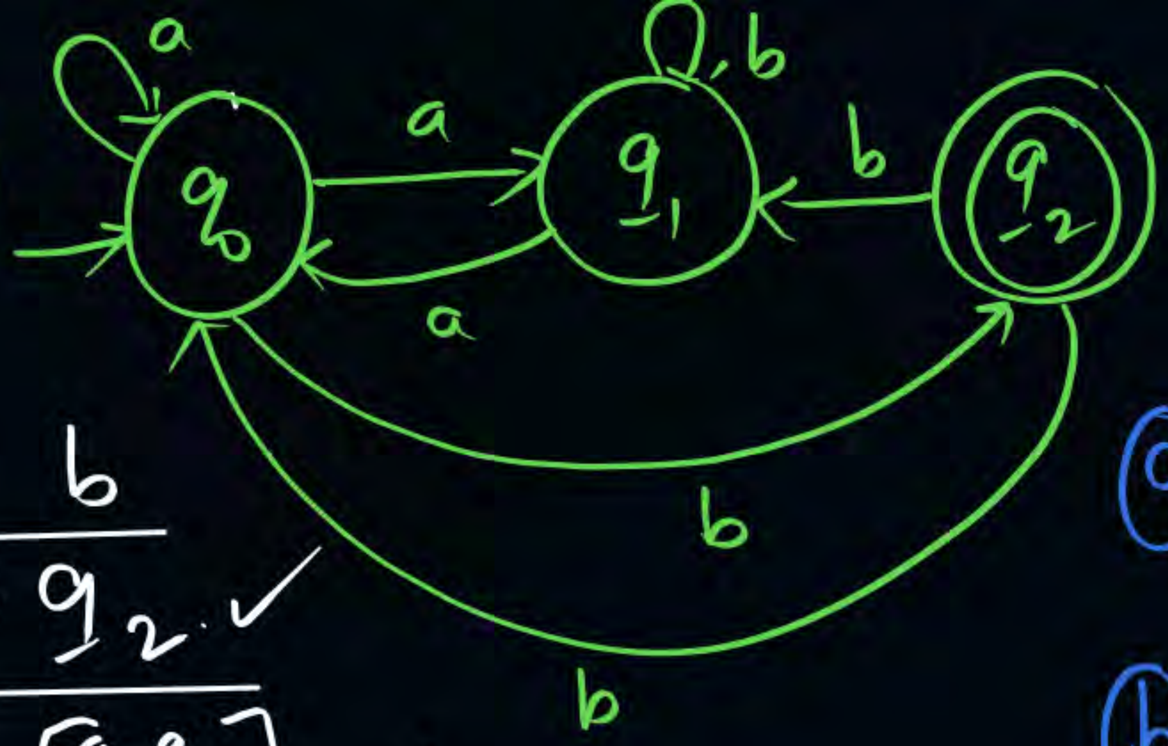
DFA

3

	a	b
$\rightarrow q_0$	$[q_0 q_1]$	q_0
$[q_0 q_1]$	$[q_0 q_1]$	$[q_0 q_2]$
$[q_0 q_2]$	$[q_0 q_1]$	q_0

Topic : NFA to DFA Conversion

NFA



NFA

	a	b
q_0	$\{q_0, q_1\}$	q_2
q_1	q_0	q_1
q_2		$\{q_0, q_1\}$

DFA

	a	b
$\rightarrow q_0$	$[q_0 q_1]$	q_2
$[q_0 q_1]$	$[q_0 q_1]$	$[q_1 q_2]$
q_2	q_{Dead}	$[q_0 q_1]$
$[q_1 q_2]$	q_0	$[q_0 q_1]$
q_{Dead}	q_{Dead}	q_{Dead}

(a) 3

(b) 4

(c) 5

(d) 6

NFA

	a	b
q_0	$\{q_0, q_1\}$	q_2
q_1	q_0	q_1
q_2	–	$\{q_0, q_1\}$

Topic : NFA to DFA Conversion

How many states in DFA?

NFA

(a) 3

~~(b) 4~~

(c) 5

(d) 6

NFA $\rightarrow$ DFA

$2 \rightarrow 4$

$3 \rightarrow 3$

$3 \rightarrow 5$

$3 \rightarrow 4$

$n \rightarrow \{ \overset{\text{min}}{1} \text{ --- } 2^n \}$

NFA DFA

$5 \rightarrow 1 \text{ -- } 2^5$

Topic : NFA to DFA Conversion

Note:- NFA to DFA conversion Algorithm does not affect language of automata i.e. if language accept by a NFA is L then while converting that NFA into DFA the resultant DFA also accepts.

Topic : ϵ -NFA

$Q \times \Sigma \rightarrow 2^n$

DFA

NFA

ϵ-NFA

NOTE: Construction of $\in$ - NFA is easy than NFA

$\boxed{\{Q, \Sigma, q_0, F, \delta\}}$

Q	-	Finite number of states (set of state)
Σ	-	Input alphabet
q_0	-	initial state
F	-	Set of final states
δ	-	transition function

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

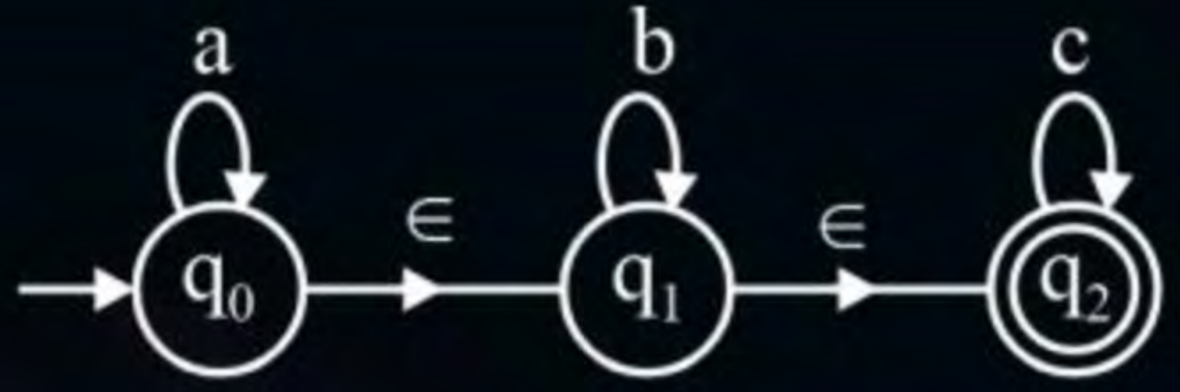
NFA

$n \longrightarrow$

DFA

2^n

ϵ-NFA



THANK - YOU