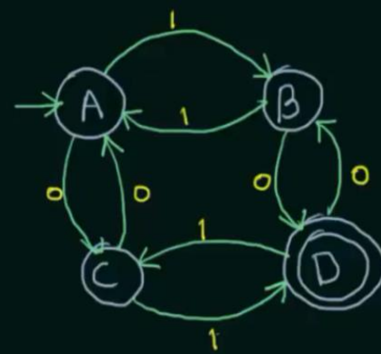
NFA - Non-deterministic Finite Automata

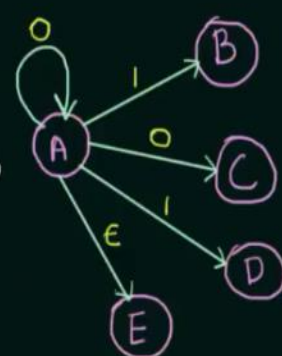
Deterministic Finite Automata
↓
DETERMINISM

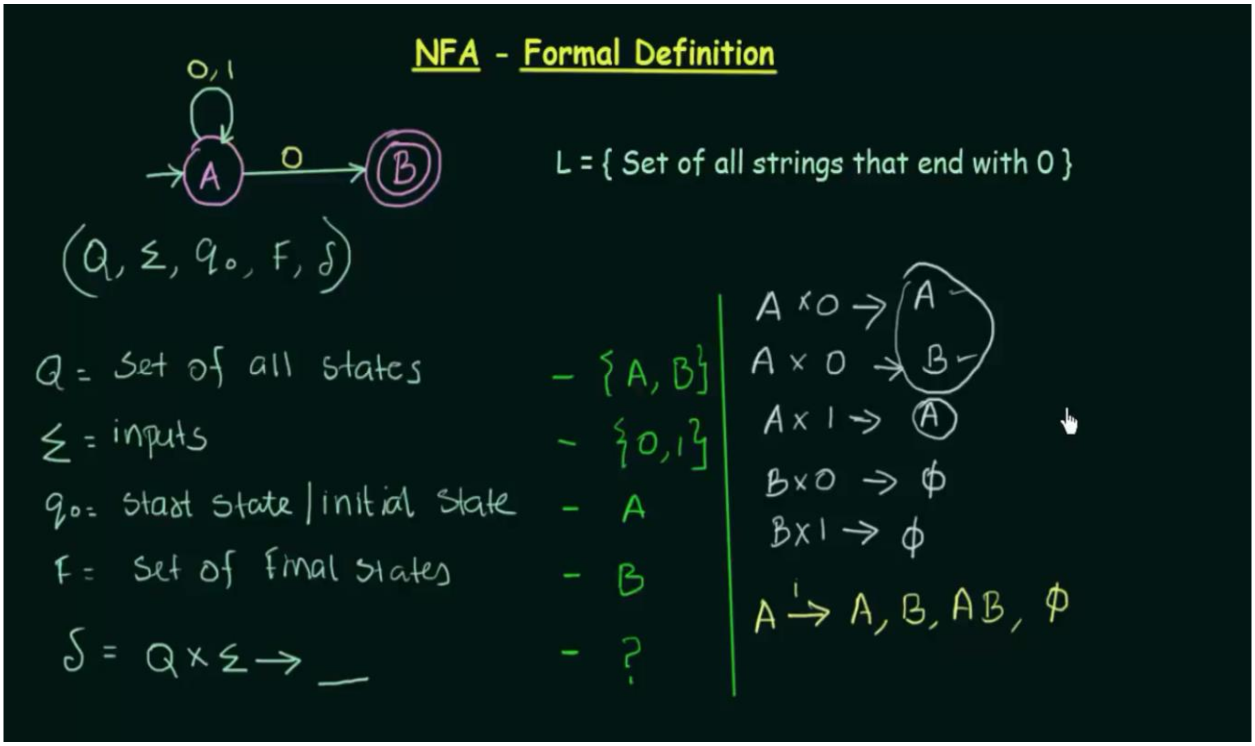
- In DFA, given the current state we know what the next state will be
- It has only one unique next state
- It has no choices or randomness
- It is simple and easy to design

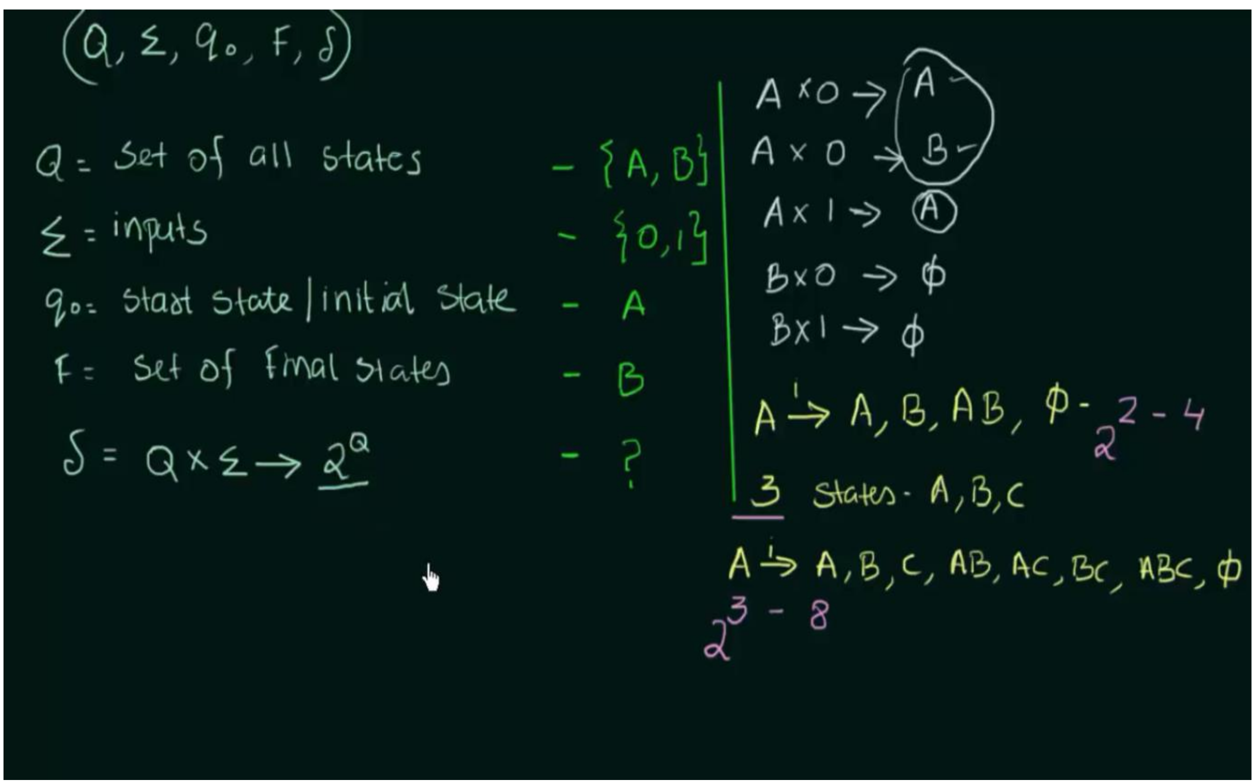


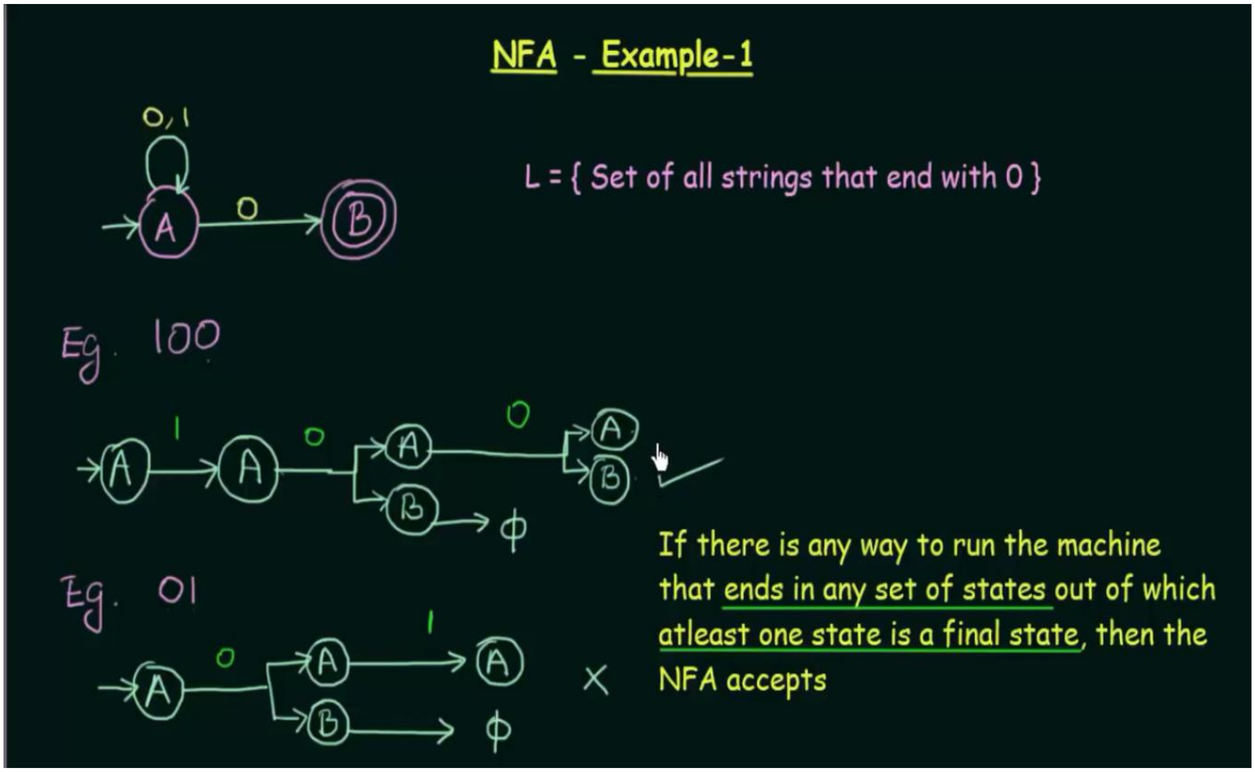
Non-deterministic Finite Automata
↓
NON-DETERMINISM

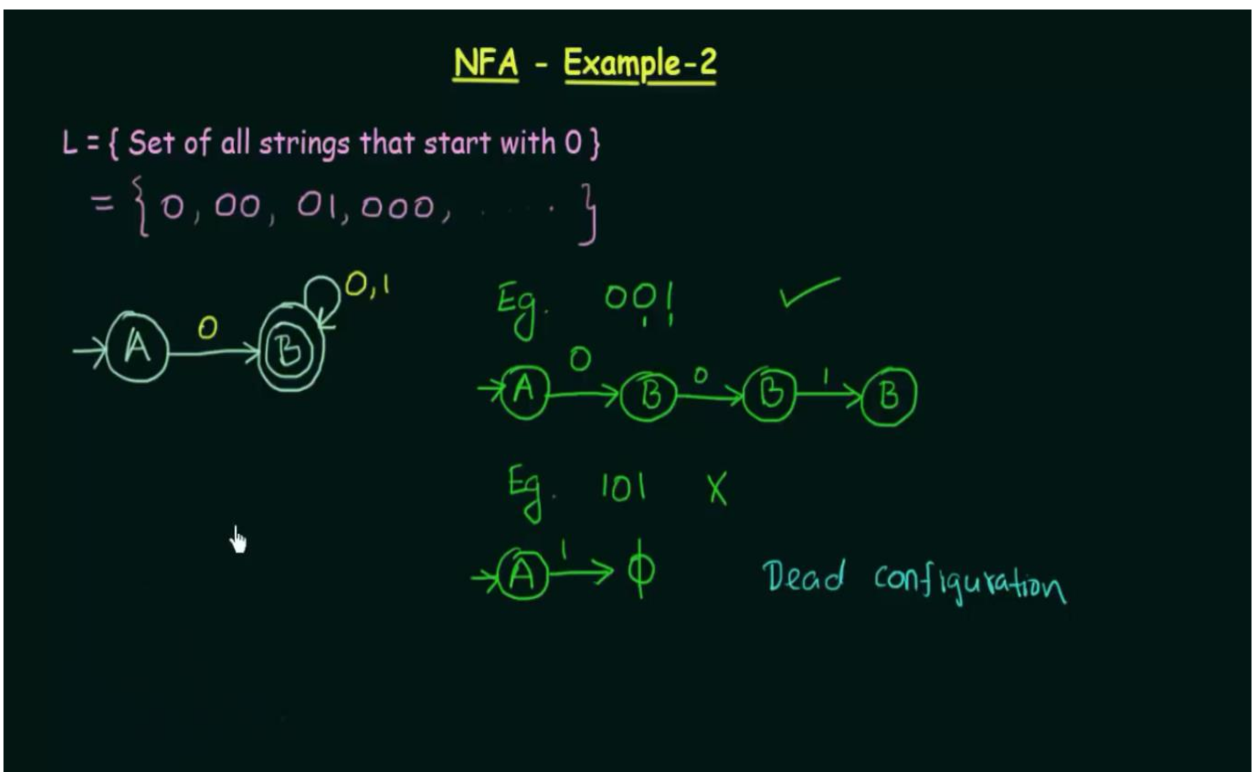
- In NFA, given the current state there could be multiple next states
- The next state may be chosen at random
- All the next states may be chosen in parallel

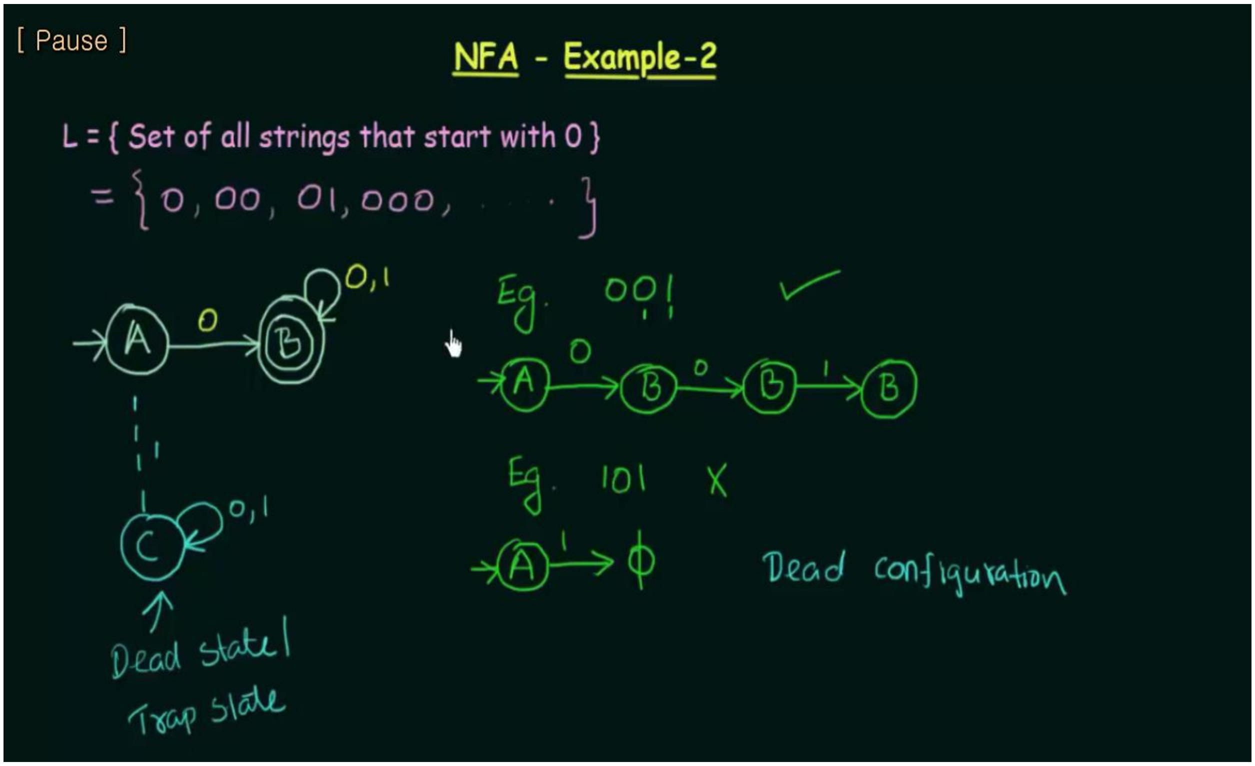


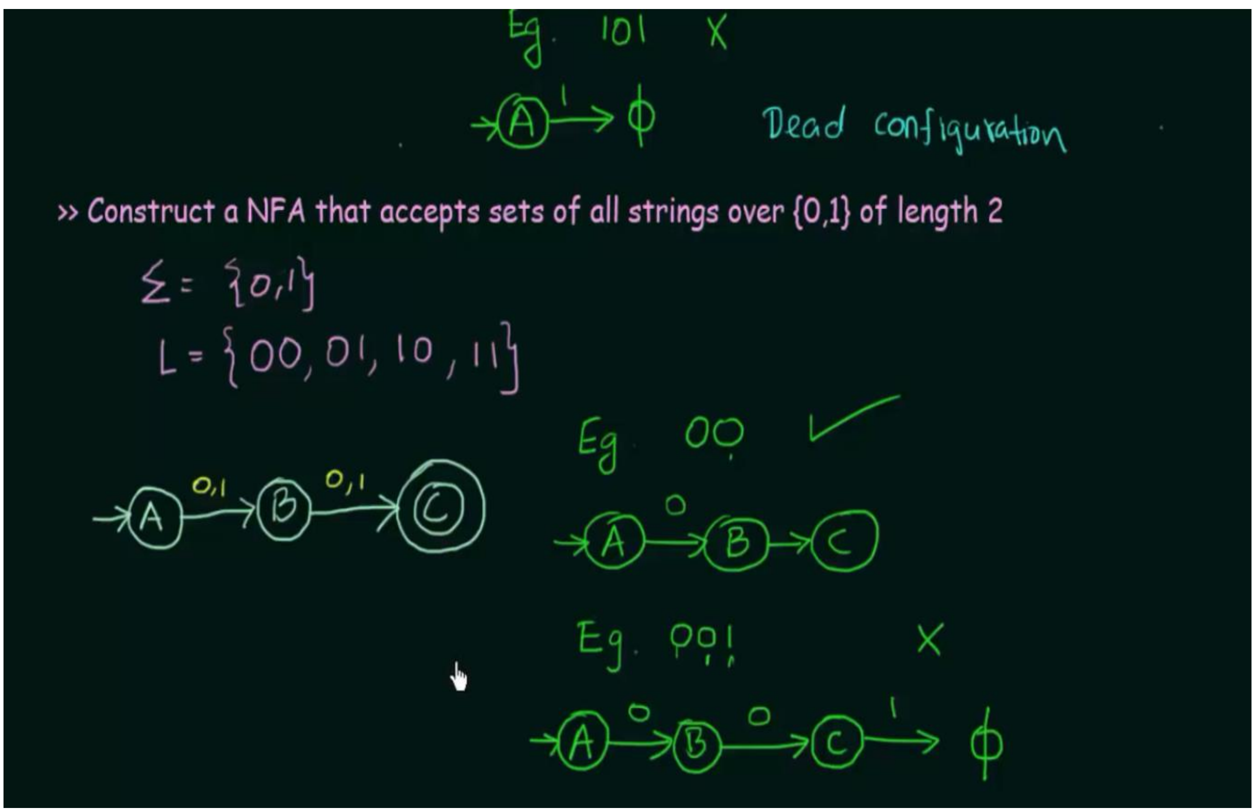

NFA - Formal Definition
0,1
A
0
B
L = { Set of all strings that end with 0 }
(Q, Σ, q0, F, δ)
Q = Set of all states - {A, B}
Σ = inputs - {0,1}
q0 = start state | initial state - A
F = set of final states - B
δ = Q × Σ → ___ - ?
A × 0 → A
A × 0 → B
A × 1 → A
B × 0 → ϕ
B × 1 → ϕ
A →1 A, B, AB, ϕ


(Q, Σ, q0, F, δ)
Q = Set of all states - {A, B}
Σ = inputs - {0,1}
q0 = start state | initial state - A
F = set of final states - B
δ = Q × Σ → 2^Q - ?
A × 0 → A
A × 0 → B
A × 1 → A
B × 0 → ϕ
B × 1 → ϕ
A →1 A, B, AB, ϕ - 2^2 - 4
3 states - A, B, C
A →1 A, B, C, AB, AC, BC, ABC, ϕ
2^3 - 8


NFA - Example-1
0,1
A
0
B
L = { Set of all strings that end with 0 }
Eg. 100
1
0
0
A
A
A
A
B
B
ϕ
If there is any way to run the machine that ends in any set of states out of which atleast one state is a final state, then the NFA accepts
Eg. 01
0
1
A
A
A
B
ϕ
X

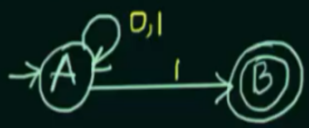

NFA - Example-2
L = { Set of all strings that start with 0 }
= { 0, 00, 01, 000, . . . }
A
0
B
0,1
Eg. 001
0
0
1
A
B
B
B
Eg. 101 X
A
1
ϕ
Dead configuration


[Pause]
NFA - Example-2
L = { Set of all strings that start with 0 }
= {0, 00, 01, 000, ...}
0,1
0
A
B
0,1
C
Dead state /
Trap state
Eg. 001 ✓
0
A B 0 B 1 B
Eg. 101 X
A 1 ϕ
Dead configuration


Eg. 101 X
A 1 ϕ
Dead configuration
» Construct a NFA that accepts sets of all strings over {0,1} of length 2
Σ = {0,1}
L = {00, 01, 10, 11}
0,1
0,1
A B C
Eg. 00 ✓
0
A B C
Eg. 001 X
A 0 B 0 C 1 ϕ

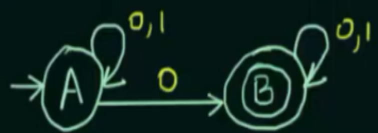
NFA - Example-3

Ex 1) L1 = { Set of all strings that ends with '1' }

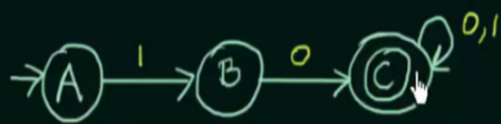


01, 001, 0001, 0*1, 1,
101, 1101,

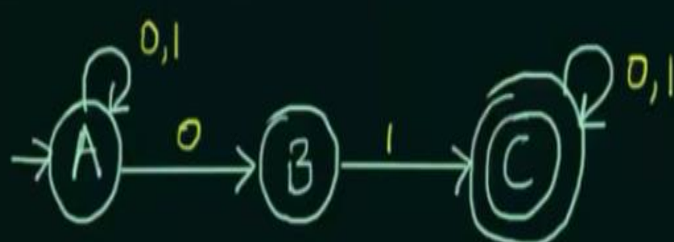
Ex 2) L2 = { Set of all strings that contain '0' }



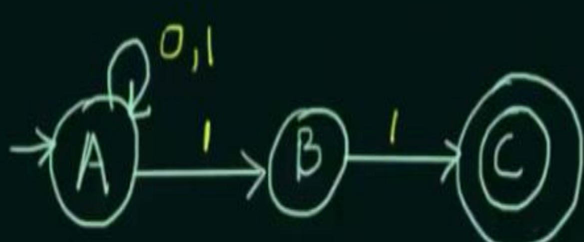
Ex 3) L3 = { Set of all strings that starts with '10' }



Ex 4) L4 = { Set of all strings that contain '01' }



Ex 5) L5 = { Set of all strings that ends with '11' }



Conversion of NFA to DFA

Every DFA is an NFA, but not vice versa

But there is an equivalent DFA for every NFA

DFA

$\delta = \underline{Q \times \Sigma \rightarrow Q}$

NFA

$\delta = \underline{Q \times \Sigma \rightarrow 2^Q}$

NFA $\cong$ DFA

L = { Set of all strings over (0,1) that starts with '0' }

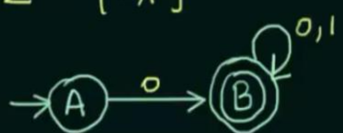
$\Sigma = \{0,1\}$

NFA $\cong$ DFA

L = { Set of all strings over (0,1) that starts with '0' }

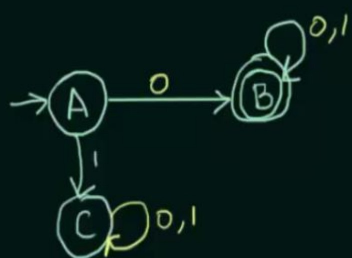
$\Sigma = \{0,1\}$

NFA



	0	1
A	B	ϕ
B	B	B

DFA



	0	1
A	B	C
B	B	B
C	C	C

C - Dead state /

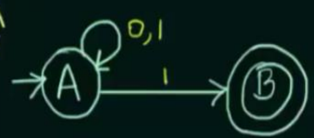
Trap state

Conversion of NFA to DFA - Examples (Part 1)

L = { Set of all strings over (0,1) that ends with '1' }

$\Sigma = 0, 1$

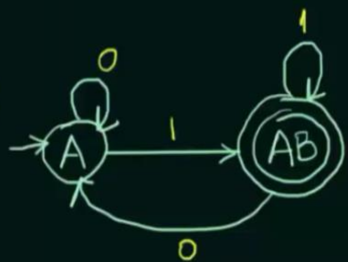
NFA



	0	1
A	{A}	{A, B}
B	ϕ	ϕ

Subset construction method

DFA



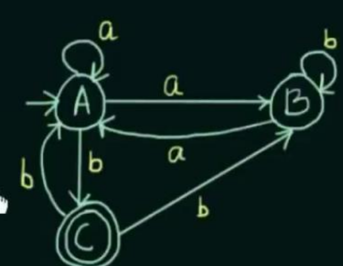
	0	1
A	{A}	{AB}
AB	{A}	{AB}

AB - Single State

Conversion of NFA to DFA - Examples (Part-2)

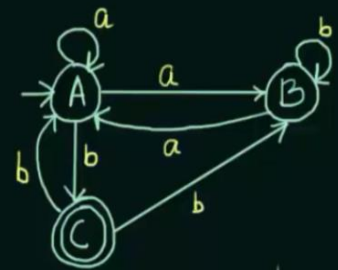
Find the equivalent DFA for the NFA given by M= [{A,B,C}, (a,b), δ, A, {C}] where δ is given by:

	a	b
→A	A, B	C
B	A	B
C	–	A, B

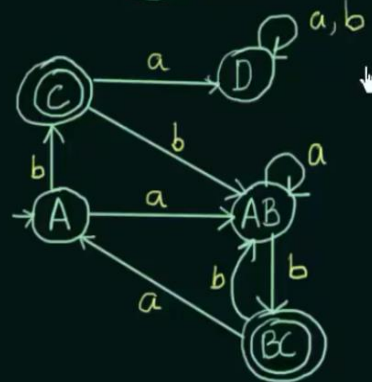


Find the equivalent DFA for the NFA given by M= [{A,B,C}, (a,b), δ, A, {C}] where
δ is given by:

	a	b
→A	A, B	C
B	A	B
C	–	A, B



	a	b
→A	AB	C
AB	AB	BC
BC	A	AB
C	D	AB
D	D	D

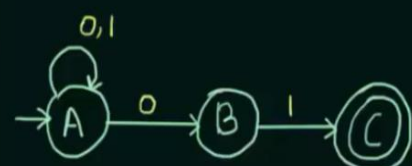


Conversion of NFA to DFA - Examples (Part-3)

Given below is the NFA for a language

L = { Set of all strings over (0,1) that ends with '01' }. Construct its equivalent DFA

NFA



	0	1
→A	A, B	A
B	Φ	C
C	Φ	Φ

	0	1
→A	AB	A
AB	AB	AC
AC	AB	A

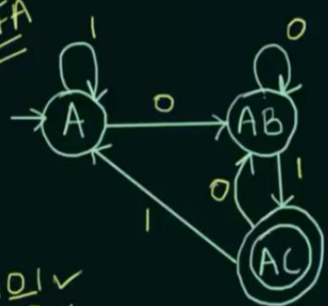
Given below is the NFA for a language

L = { Set of all strings over (0,1) that ends with '01' }. Construct its equivalent DFA

NFA

	0	1
→A	A, B	A
B	ϕ	C
C	ϕ	ϕ

DFA



	0	1
→A	AB	A
AB	AB	AC
AC	AB	A

1101 ✓
1110 ×

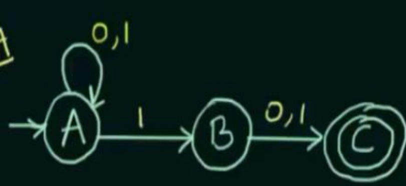
[Pause]

Conversion of NFA to DFA - Examples (Part-4)

Design an NFA for a language that accepts all strings over {0,1} in which the second last symbol is always '1'. Then convert it to its equivalent DFA.

Eg. 1010
110
1101010

NFA

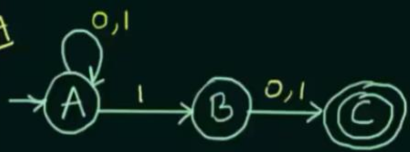


	0	1
→A	A	A, B
B	C	C
C	ϕ	ϕ

second last symbol is always '1'. Then convert it to its equivalent DFA.

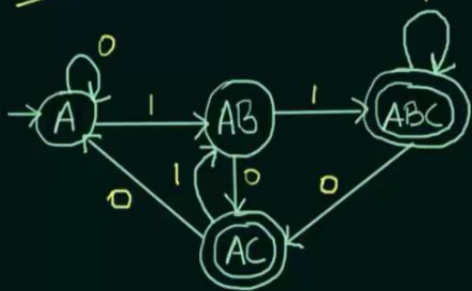
Eg. 1010 ✓
110 ✓
1101010 ✓

NFA



	0	1
→A	A	A,B
B	C	C
C	ϕ	ϕ

DFA



	0	1
→A	A	AB
AB	AC	ABC
AC	A	AB
ABC	AC	ABC