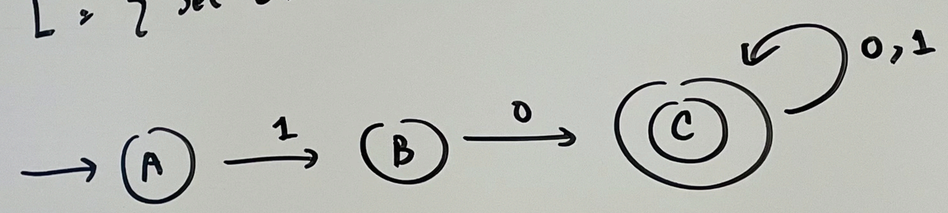
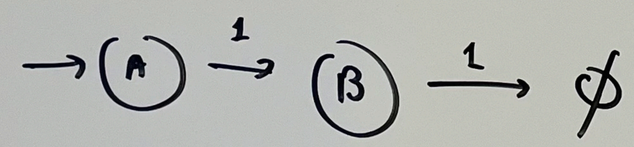
Non-Deterministic Finite Automata (NFA)

* Example 6

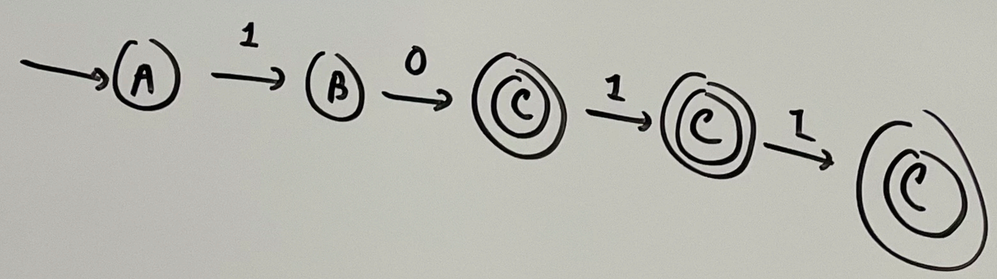
L = { Set of all strings that start with '10' }



Eg: 11 ×



Eg: 1011 ✓



* Example 7

L = {

* Proof for

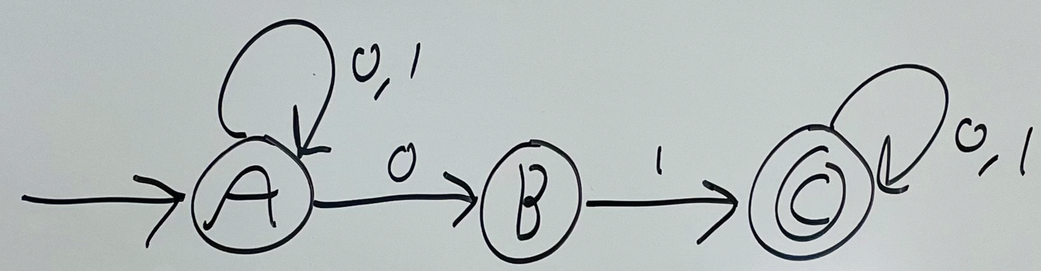
(NFA)

0'}

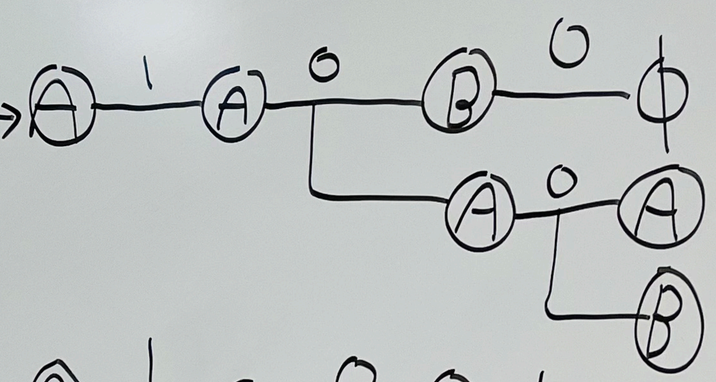
* Example 7

L = { Set of all strings that contain '01'}

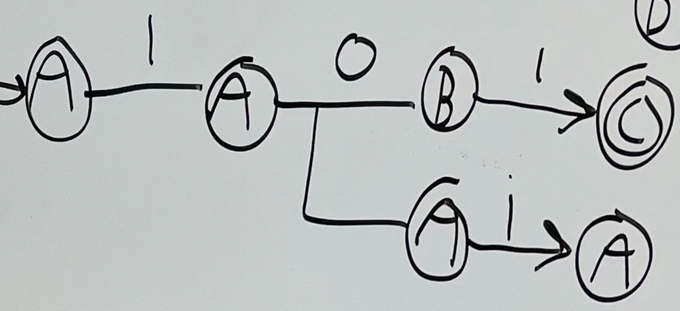
* Proof for 100 , 101



× Eg 100 →



✓ Eg 101 →



NFA

#1 "end wi

0,1

end wi

#4 →

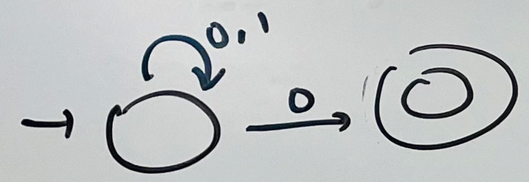
#3 length

→

Only

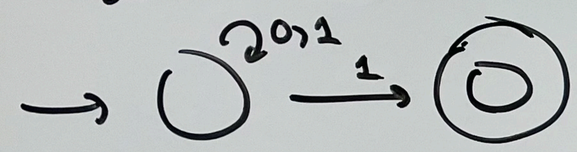
NFA Design Patterns

#1 "end with 0'

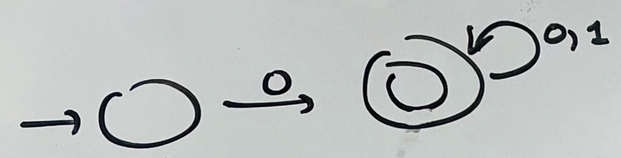


end with 1

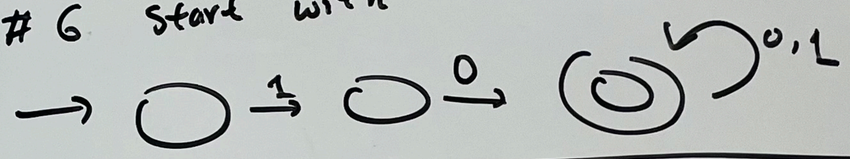
#4



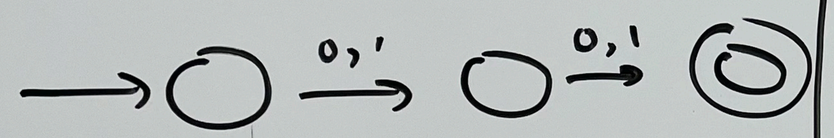
#2 Start with 0



6 Start with '10'

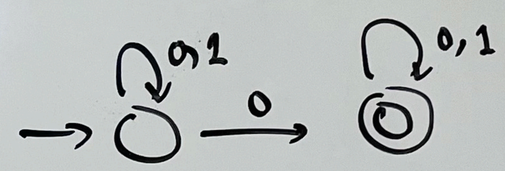


#3 length 2

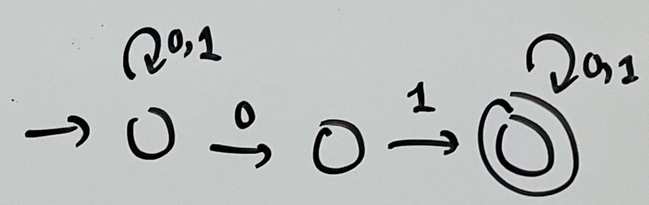


Only focus on accepting valid strings

5 Contain 0



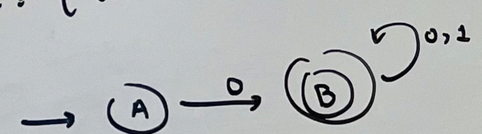
7 Contain 01



Conversion of NFA to DFA

Example 1 L : { Set of all strings over $\Sigma = \{0,1\}$ that starts with '0' }

NFA :



State Transition Table

NFA :

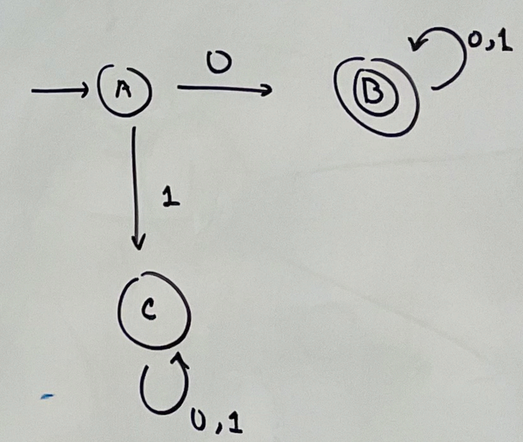
States	Inputs 0	Inputs 1
A	B	ϕ
* B	B	B

State Transition Table

DFA :

States	0	1
A	B	C
* B	B	B
C	C	C

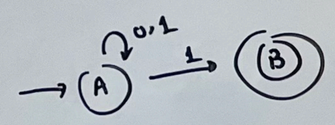
DFA



★ Example 2

L = { Set of all strings over $\Sigma = \{0,1\}$ that ends with '1' }

NFA:



State Transition Table

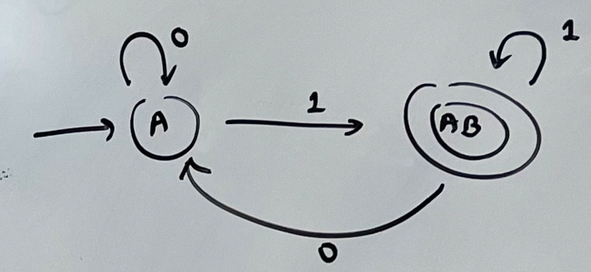
NFA:

	0	1
A	{A}	{A,B}
★ B	ϕ	ϕ

→ DFA

State Transition Table

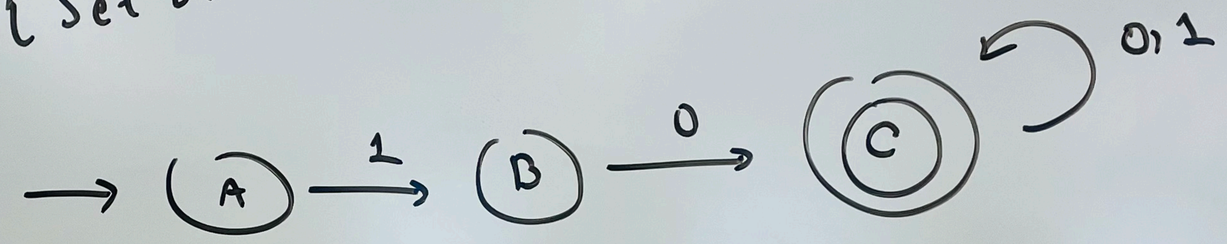
	0	1
→ A	A	AB
★ AB	A	AB



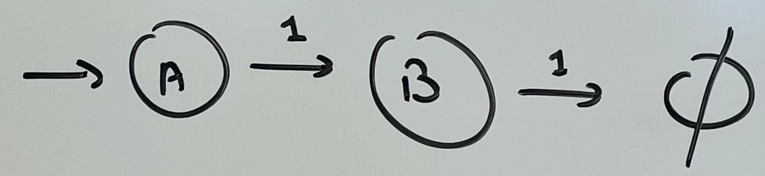
Non Deterministic Finite Automata (NFA)

★ Example 6

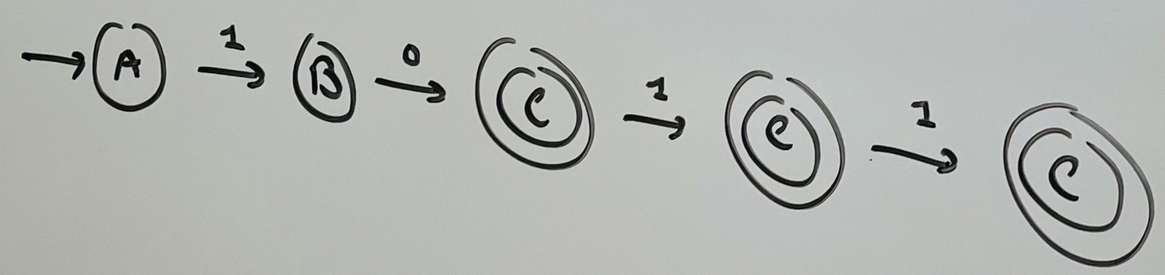
L = { Set of all strings that start with '10' }



Eg: 11 ✗



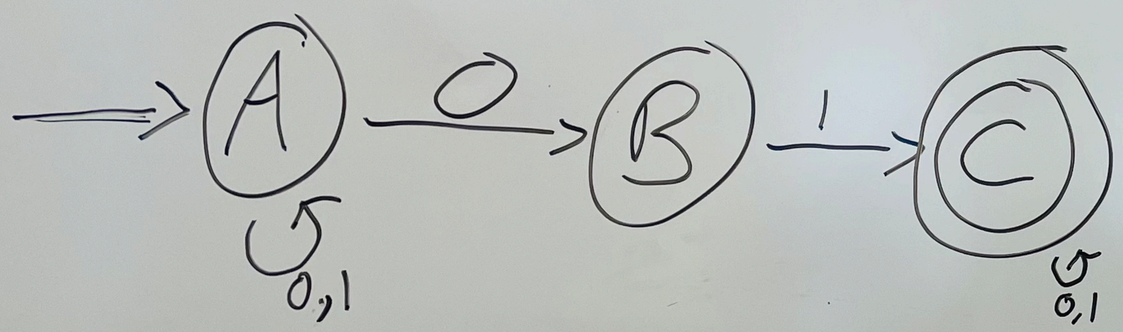
Eg: 1011 ✓

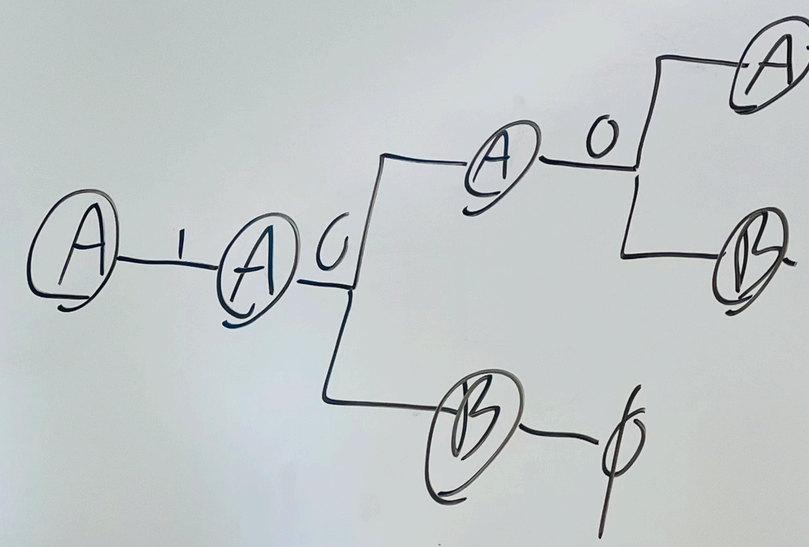


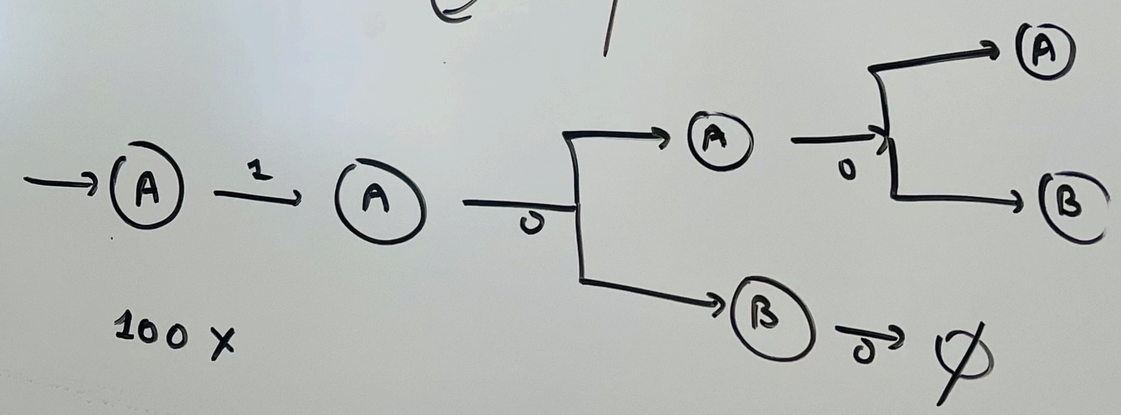
* Example 7

L = { Set of all strings that contain '01' }

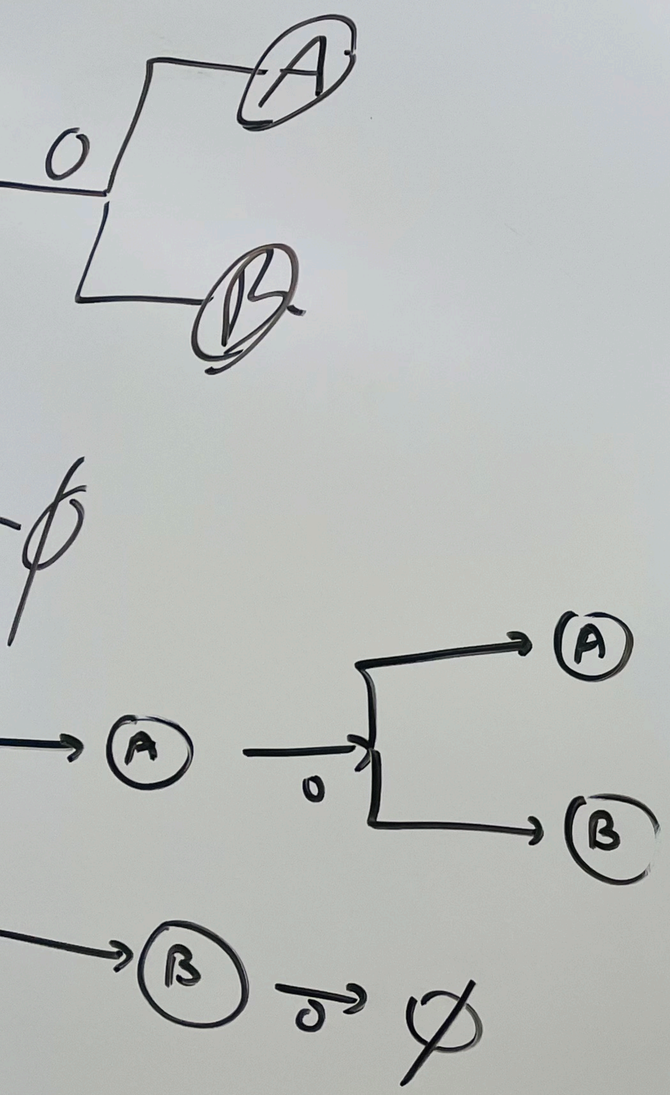
* Proof for 100 , 101 — are they valid?



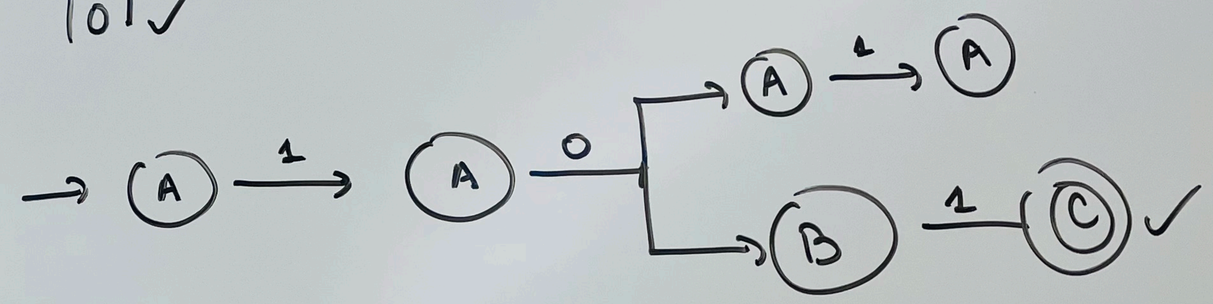




100 X

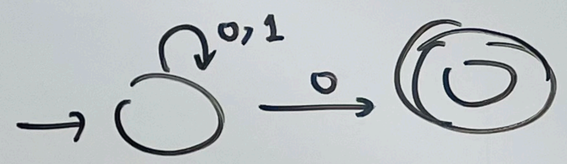


101 ✓

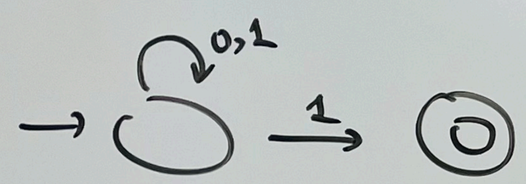


NFA Design Patterns

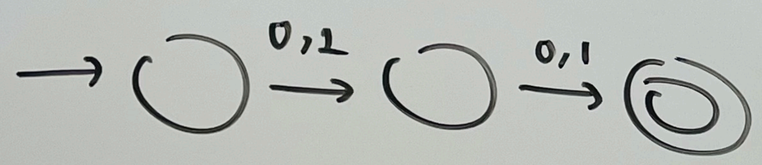
#1 end with "0"



#4 end with "1"

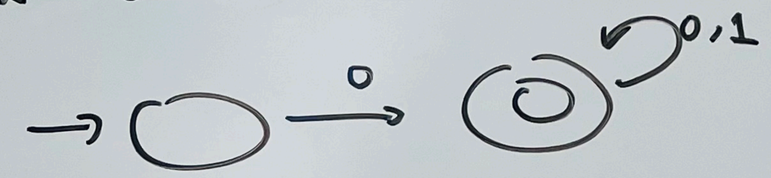


#3 length 2

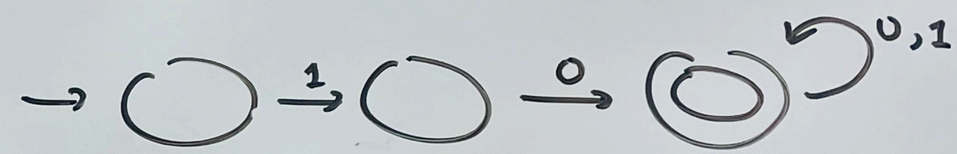


Only focus on accepting valid strings.

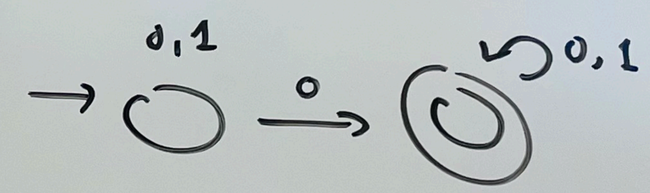
#2 Start with "0"



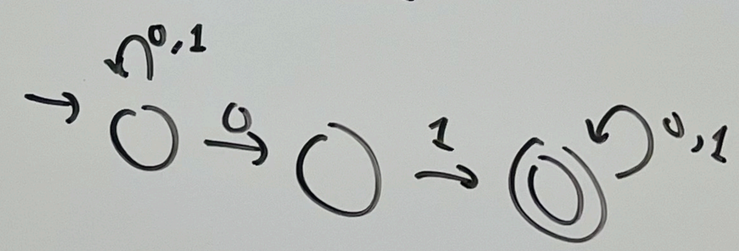
6 Start with '10'



#5 Contain 0



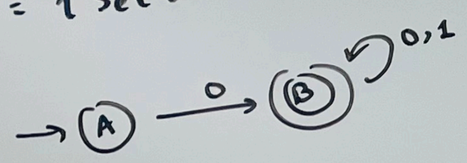
#7 Contain 01



★ Conversion of NFA to DFA

Example 1: L = { Set of all strings over $\Sigma = \{0,1\}$ that starts with '0' }

NFA:



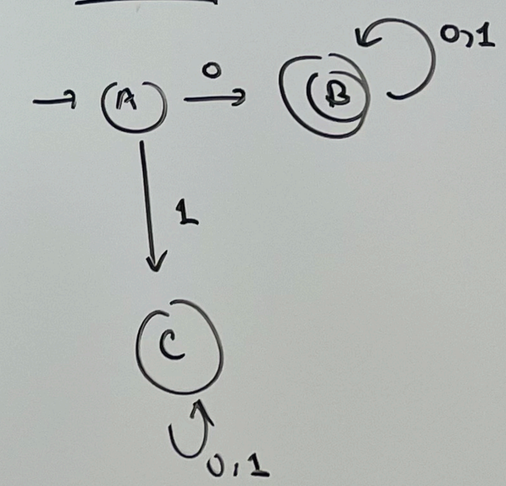
State Transition Table

States	Inputs 0	Inputs 1
→ A	B	∅
* B	B	B

→ DFA :

	0	1
→ A	B	C
* B	B	B
C	C	C

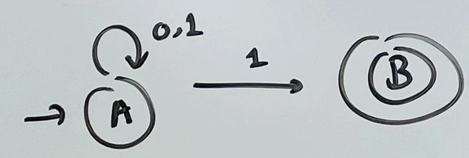
DFA



* Example 2

L = { Set of all strings over $\Sigma = \{0,1\}$ that ends with 1 }

NFA:



State Transition Table

NFA:

	0	1
→ A	{A}	{A, B}
* B	ϕ	ϕ

→ DFA:

	0	1
→ A	A	AB
* AB	A	AB

$\Sigma = \{0,1\}$ that ends with 1 }

3

ble

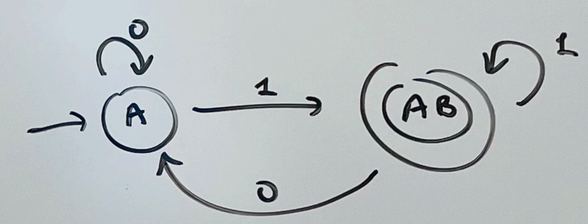
, B}

ϕ

→ DFA:

	0	1
→ A	A	AB
* AB	A	AB

DFA:



Conversion of NFA to DFA using the Subset Construction Method

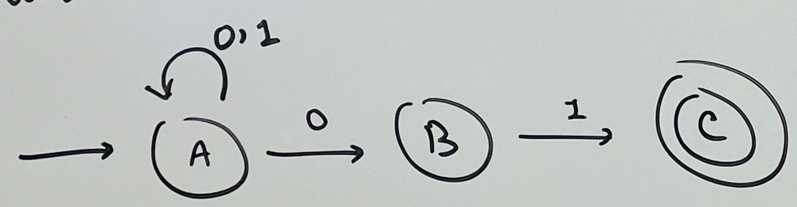
★ Example 3

Given below is the NFA for the language

L = { Set of all strings over $\Sigma = \{0, 1\}$ that ends with '01' }

Construct its equivalent DFA.

NFA:



State Transition Table

NFA:

States	Inputs 0	Inputs 1
→ A	{A, B}	{A}
B	ϕ	{C}
★ C	ϕ	ϕ

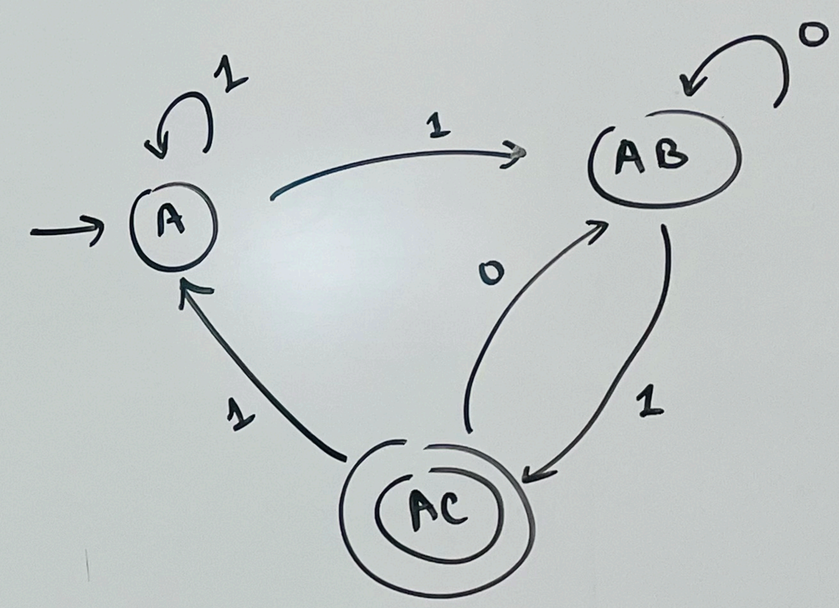
DFA

State Transition Table

DFA :

	0	1
→ A	{AB}	{A}
AB	{AB}	{AC}
★ AC	{AB}	{A}

DFA

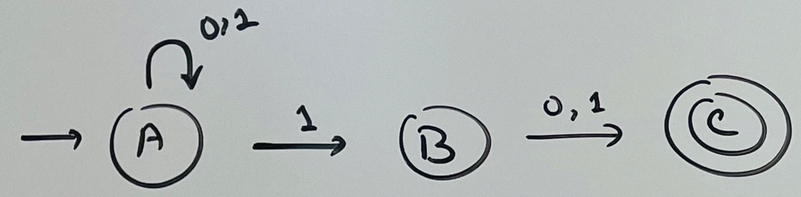


★ Example 4

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

$\{ \underline{1}0, \underline{1}1, 100\underline{1}1, 000\underline{1}0, 010\ldots\underline{1}\,0^{0/1} \}$

NFA:



State Transition Table

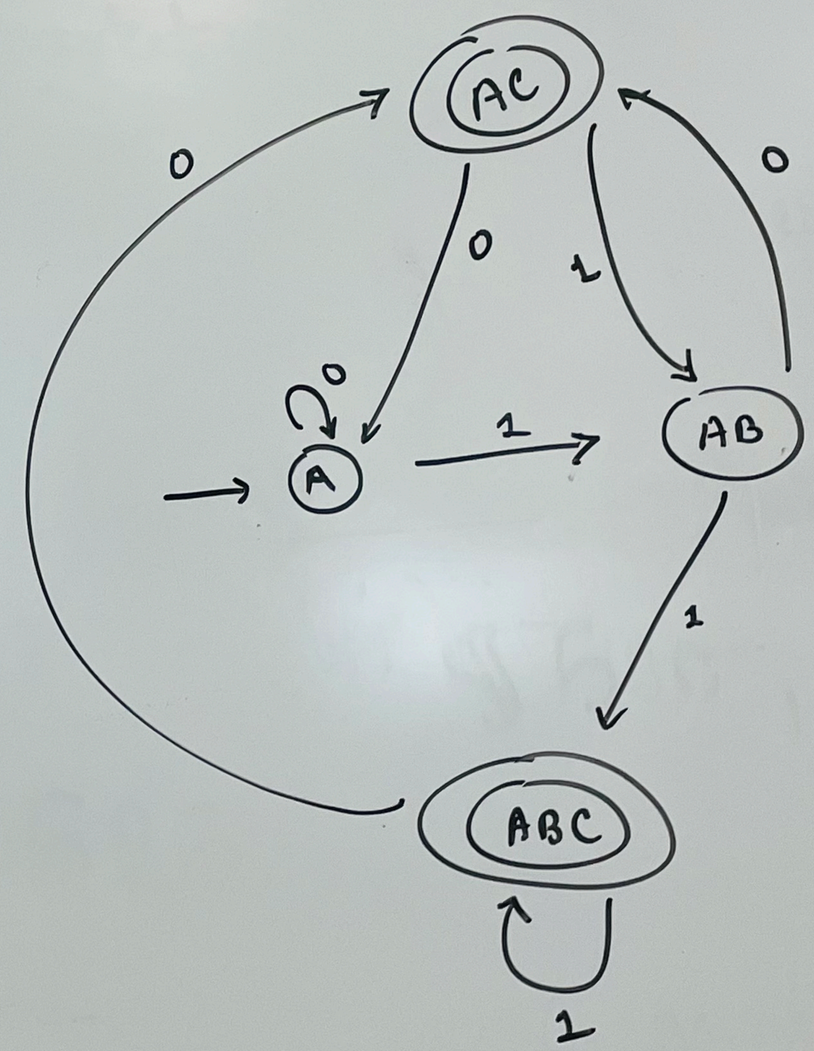
States	0	1
→ A	{A}	{A, B}
B	{C}	{C}
* C	∅	∅

NFA: →

State Transition Table

→ DFA:

	0	1
→ A	{A}	{AB}
AB	{AC}	{ABC}
★ AC	{A}	{AB}
★ ABC	{AC}	{ABC}



★ Conversion of NFA to DFA using Subset Construction Method

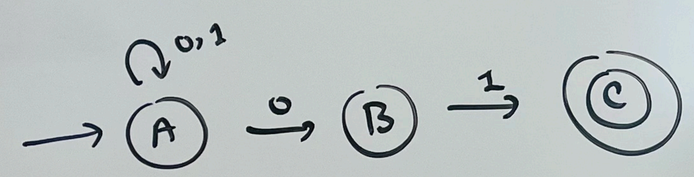
★ Example 3

Given below is the NFA for a language

L = { set of all strings over $\Sigma = \{0, 1\}$ that ends with '01' }

Construct its equivalent DFA.

NFA:



State Transition Table

NFA:

States	Inputs 0	Inputs 1
→ A	{A, B}	{A}
B	∅	{C}
★ C	∅	∅

State Transition Table

DFA:

States	Inputs 0	Inputs 1
→ A	{AB}	{A}
AB	{AB}	{AC}
★ AC	{AB}	{A}

Construction Method

uage

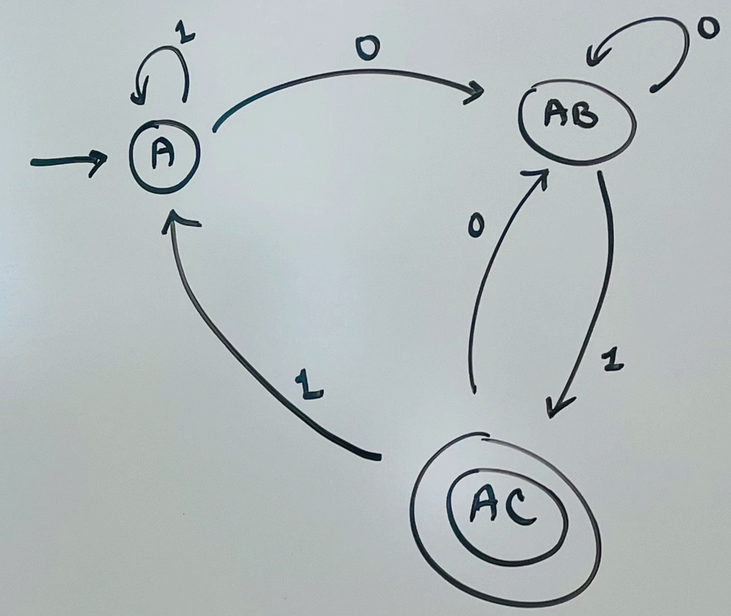
} that ends with '01' }

State Transition Table

	Inputs	
States	0	1
→ A	{AB}	{A}
AB	{AB}	{AC}
* AC	{AB}	{A}

PA:

DFA:



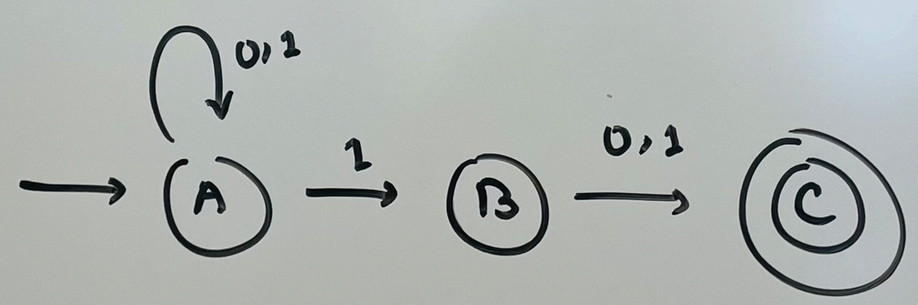
Please Erase!
ll Boards After
ch Lecture

★ Example 4

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

{ 10 , 11 , 100 11 , 0,1 1 0,1 }

NFA :



State Transi

NFA :

States	T
→ A	{A
B	{
★ C	

s all
second last
o its

0,1

NFA:

State Transition Table

States	Inputs 0	Inputs 1
→ A	{A}	{A,B}
B	{C}	{C}
* C	∅	∅

→

DFA:

State Transition Table

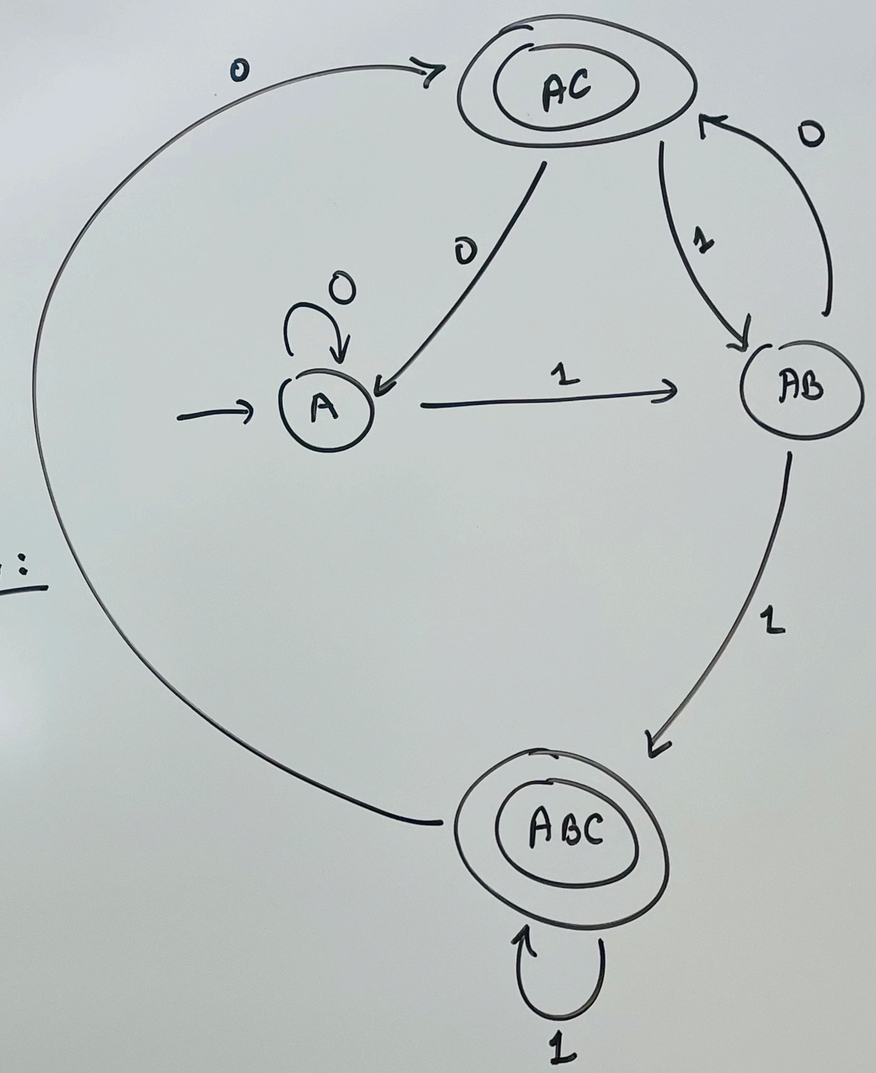
States	0	1
→ A	{A}	{AB}
AB	{AC}	{ABC}
* AC	{A}	{AB}
* ABC	{AC}	{ABC}

DFA:

State Transition Table

States	0	1
→ A	{A}	{AB}
AB	{AC}	{ABC}
★ AC	{A}	{AB}
★ ABC	{AC}	{ABC}

DFA:

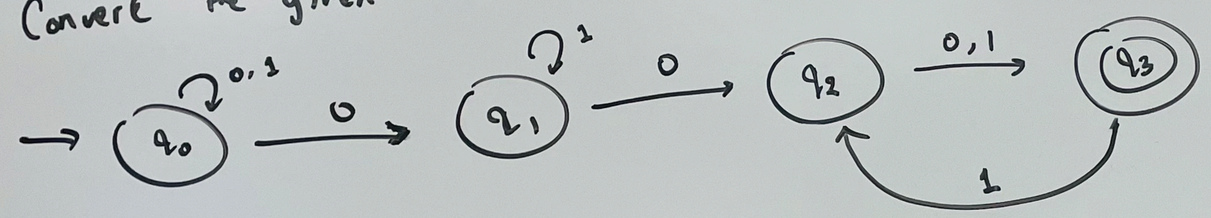


C	ϕ	A, B
C	D	AB
D	D	D

Conversion of NFA to DFA using the subset construction method

★ Example 5

Convert the given NFA to DFA :



State Transition Table

NFA:

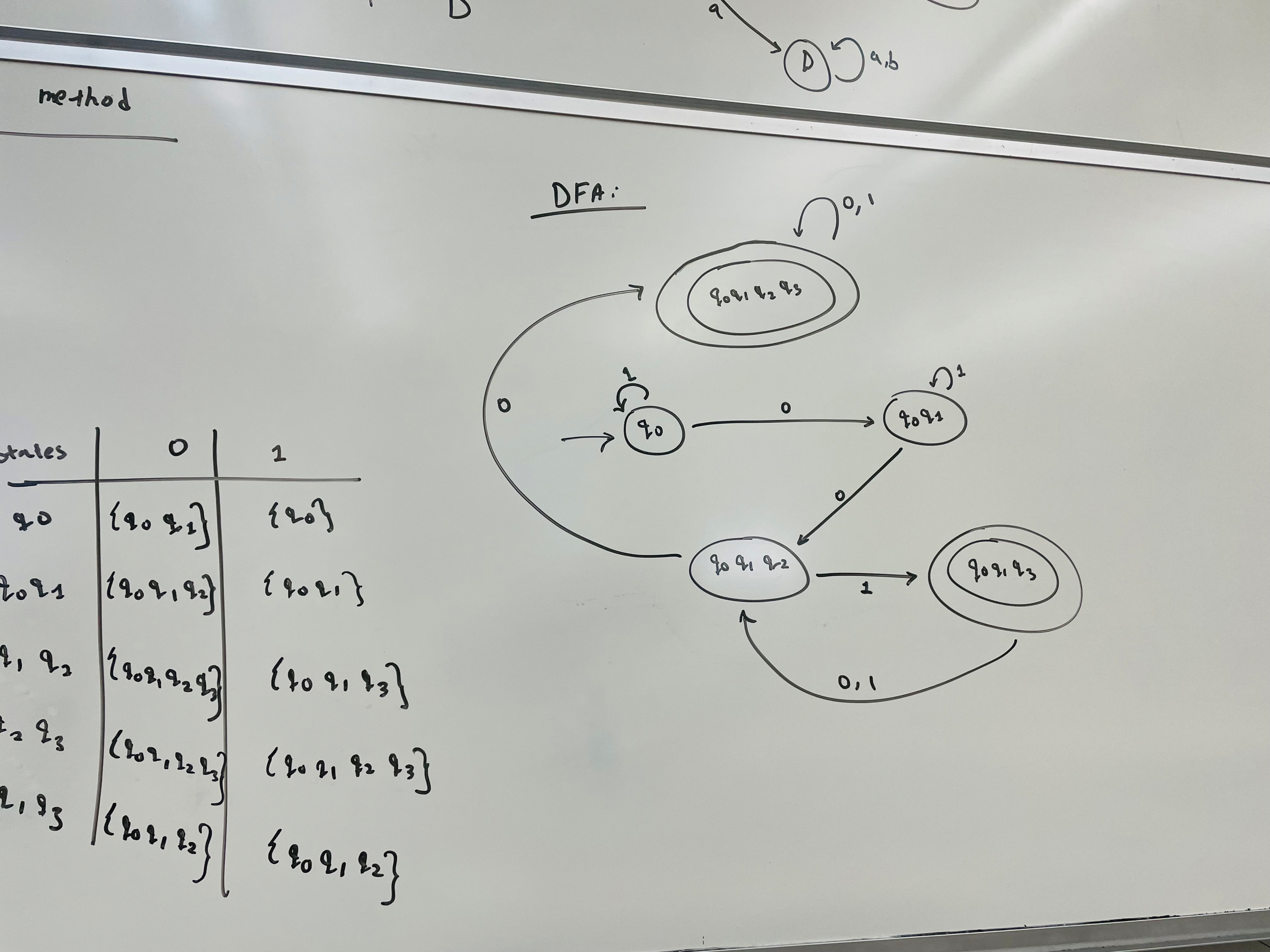
States	0	1
→ q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_1\}$
q_2	$\{q_3\}$	$\{q_3\}$
★ q_3	ϕ	$\{q_2\}$

→ DFA:

States	0	1
→ q_0	$\{q_0 q_1\}$	$\{q_0\}$
$q_0 q_1$	$\{q_0 q_1 q_2\}$	$\{q_0 q_1\}$
$q_0 q_1 q_2$	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_3\}$
★ $q_0 q_1 q_2 q_3$	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_2 q_3\}$
★ $q_0 q_1 q_3$	$\{q_0 q_1 q_2\}$	$\{q_0 q_1 q_2\}$

DFA:

0



states	0	1
q_0	$\{q_0 q_1\}$	$\{q_0\}$
$q_0 q_1$	$\{q_0 q_1 q_2\}$	$\{q_0 q_1\}$
$q_1 q_2$	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_3\}$
$q_2 q_3$	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_2 q_3\}$
$q_1 q_3$	$\{q_0 q_1 q_2\}$	$\{q_0 q_1 q_2\}$

Special Case

Find the equivalence DFA to NFA given by:

$$M = [\ \underbrace{\{A, B, C\}}_{\text{States}},\ \underbrace{(a, b)}_{\text{Inputs}},\ \underbrace{\delta}_{\text{delt transition function}},\ \underbrace{A}_{\text{starting state}},\ \underbrace{\{C\}}_{\text{final accepting state}}\ ]$$

→ formal way of representing NFA

where δ is given by

NFA:

	Inputs	
	a	b
→ A	A, B	C
B	A	B
C	$\emptyset$	A, B

→ DFA:

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

y :

, A , {C}] → formal way of representing NFA

on
ction

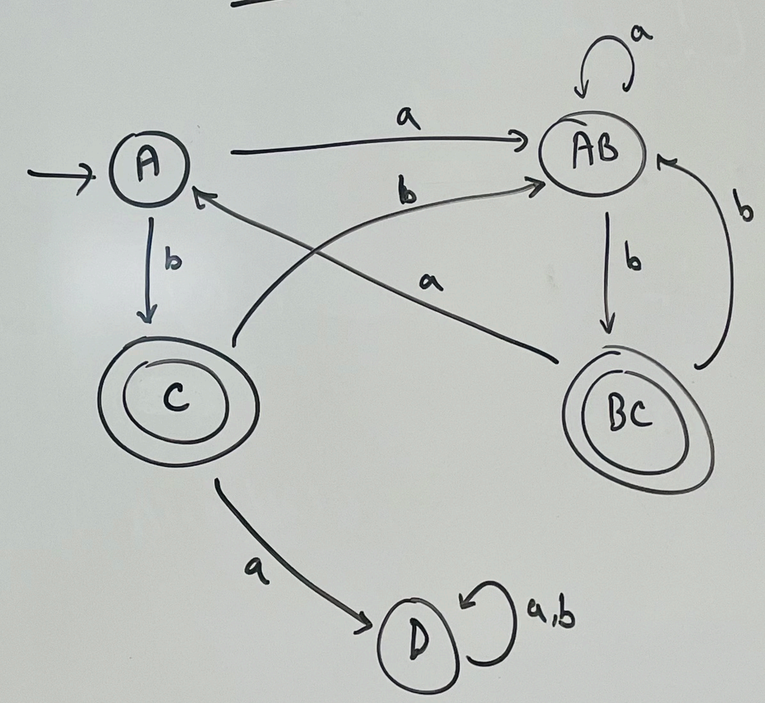
A: starting state

{C}: final accepting state

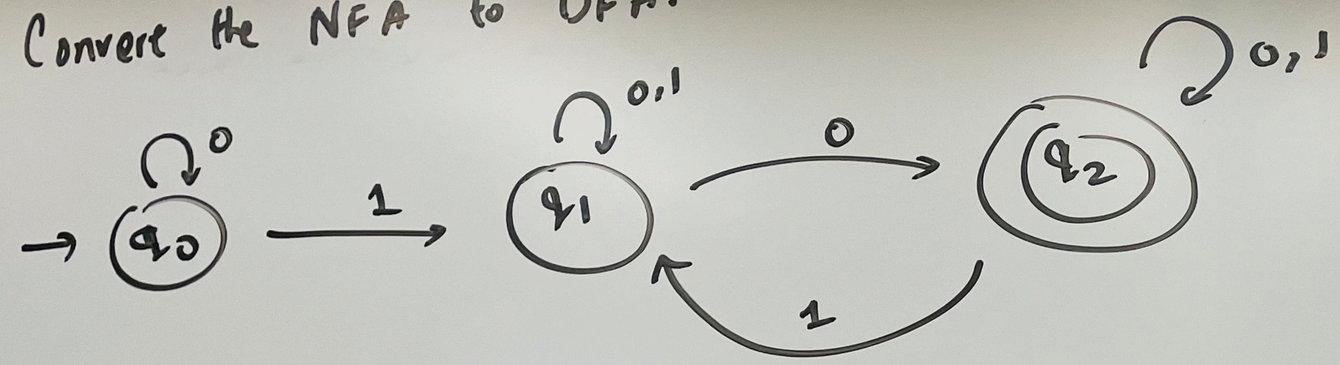
DFA:

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

DFA:



★ Convert the NFA to DFA:



NFA

	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_1\}$
q_1	$\{q_1, q_2\}$	$\{q_1\}$
★ q_2	$\{q_2\}$	$\{q_1, q_2\}$

✓

DFA

	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_1\}$
q_1	$\{q_1, q_2\}$	$\{q_1\}$
★ $q_1 q_2$	$\{q_1, q_2\}$	$\{q_1, q_2\}$

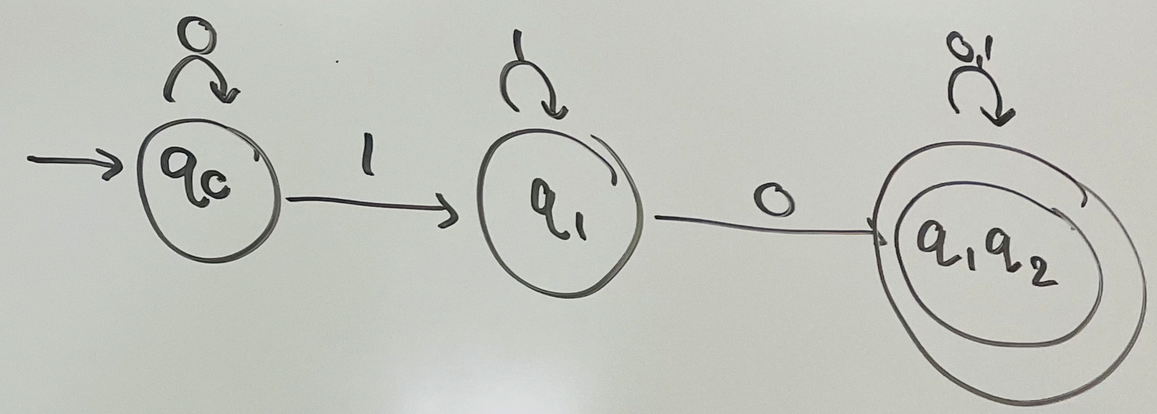
✓

) 0, 1

DFA	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_1\}$
q_1	$\{q_1, q_2\}$	$\{q_1\}$
$\star\ q_1 q_2$	$\{q_1, q_2\}$	$\{q_1, q_2\}$

✓

DFA

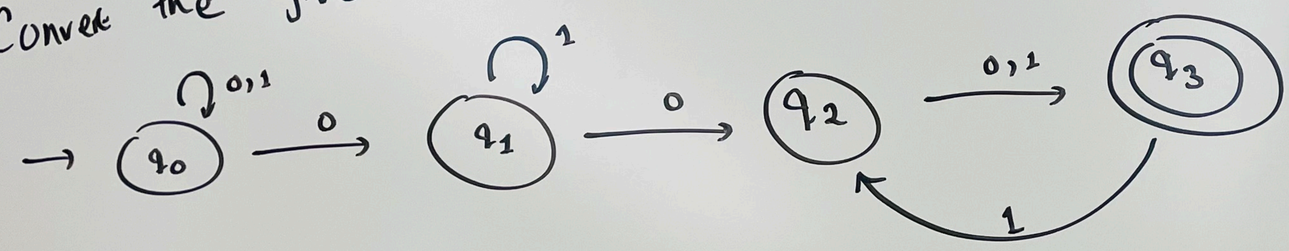


✓

D | D | D

Conversion of NFA to DFA using the Subset Construction Method

Convert the given NFA to DFA:



State Transition Table

NFA:

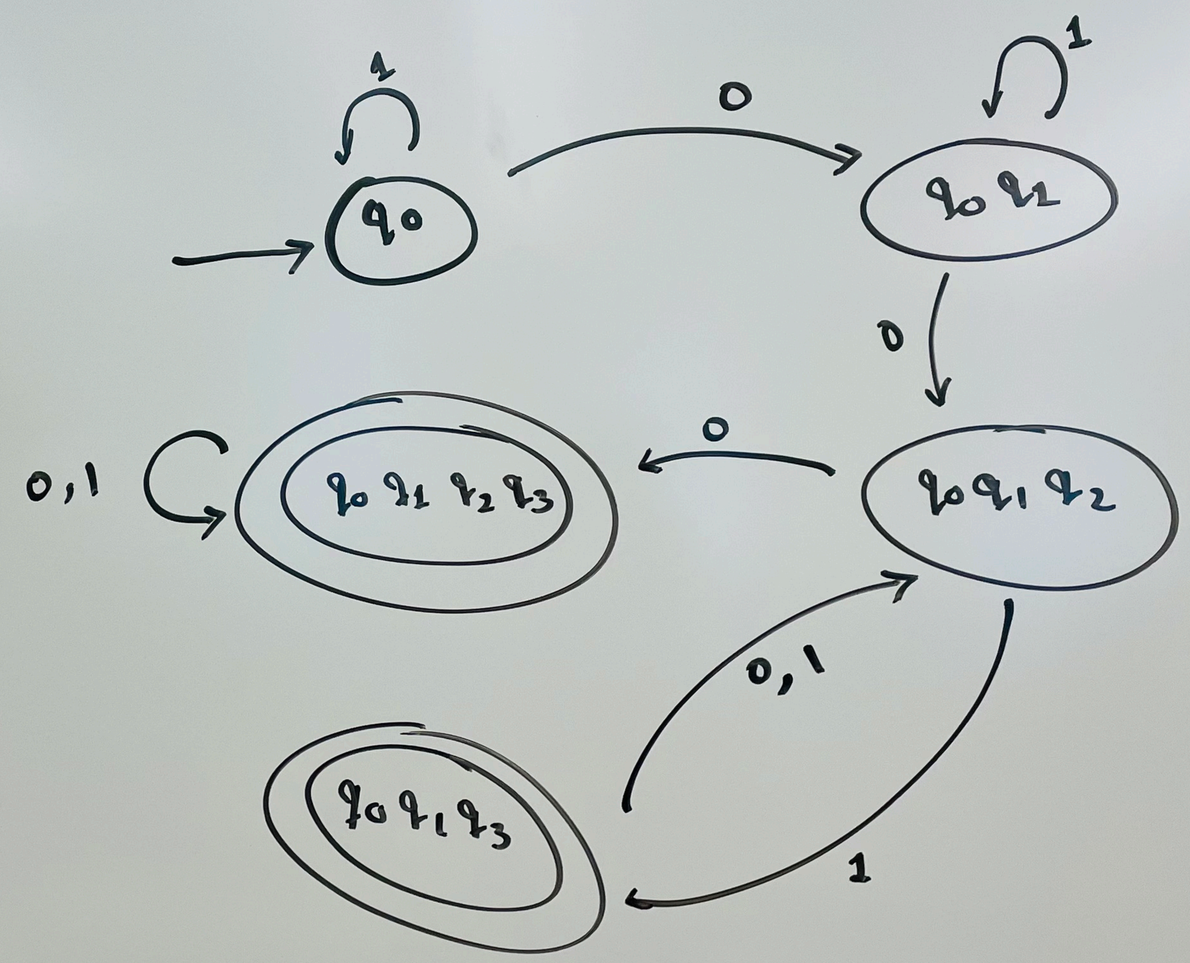
States	Inputs 0	Inputs 1
→ q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_1\}$
q_2	$\{q_3\}$	$\{q_3\}$
* q_3	$\emptyset$	$\{q_2\}$

→

States	Inputs 0	Inputs 1
→ q_0	$\{q_0 q_1\}$	$\{q_0\}$
$q_0 q_1$	$\{q_0 q_1 q_2\}$	$\{q_0 q_1\}$
$q_0 q_1 q_2$	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_3\}$
* $q_0 q_1 q_2 q_3$	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_2 q_3\}$
* $q_0 q_1 q_3$	$\{q_0 q_1 q_2\}$	$\{q_0 q_1 q_2\}$

tion Method

States	Inputs 0	1
q_0	$\{q_0 q_1\}$	$\{q_0\}$
q_1	$\{q_0 q_1 q_2\}$	$\{q_0 q_1\}$
q_2	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_3\}$
q_3	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_2 q_3\}$
	$\{q_0 q_1 q_2\}$	$\{q_0 q_1 q_2\}$



$\boxed{\text{Special Case}}$

Find the equivalent DFA to NFA given by

$$M = [\underbrace{\{A, B, C\}}_{\text{States}}, \underbrace{(a, b)}_{\text{inputs}}, \underbrace{\delta}_{\text{delt transition function}}, \underbrace{A}_{\text{Starting state}}, \underbrace{\{C\}}_{\text{final accepting state}}] \rightarrow$$

formal way of representing NFA

where δ is given by

NFA:

	a	b
→ A	A, B	C
B	A	B
C	$\emptyset$	A, B

DFA:

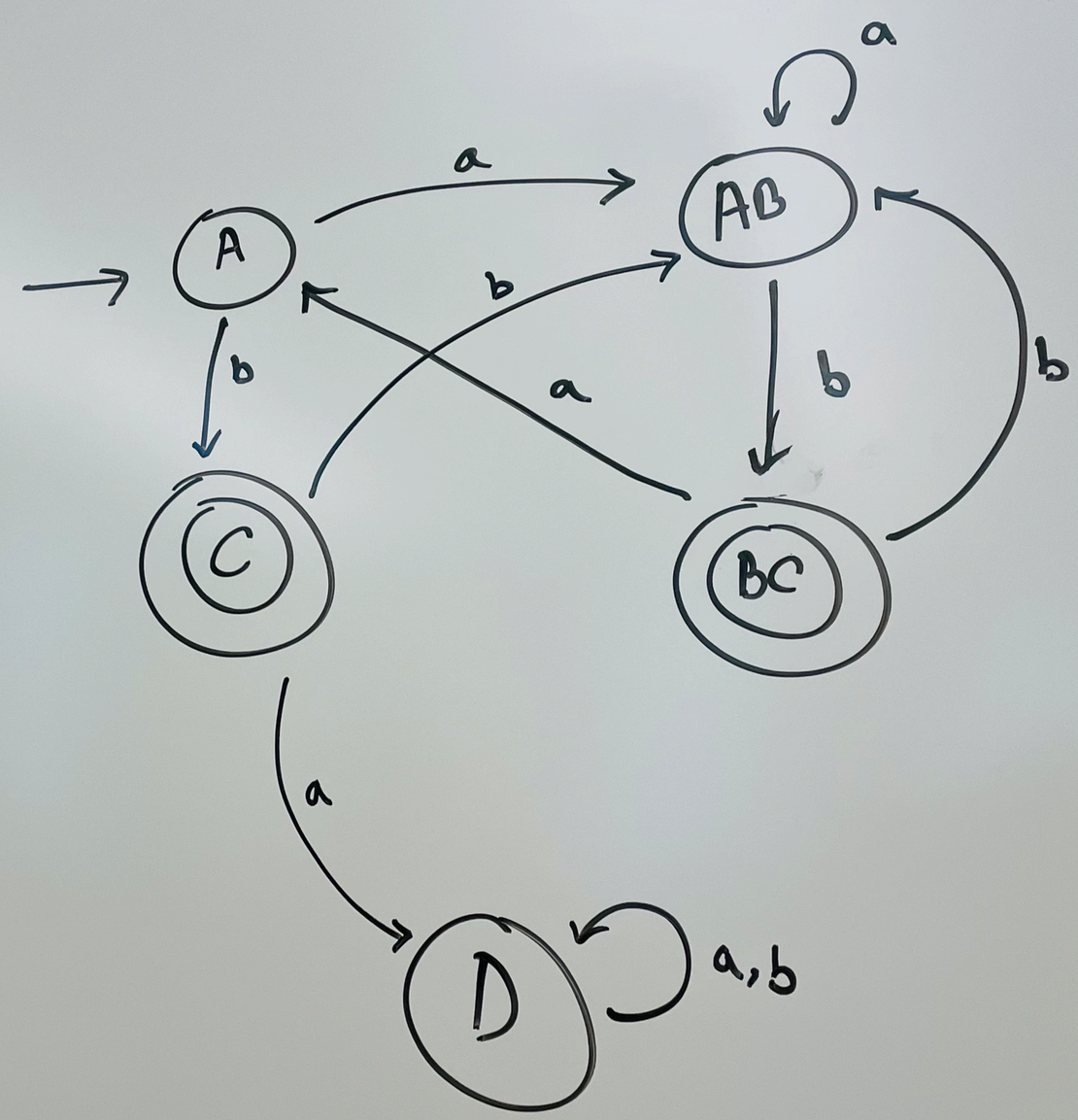
states	a	b
→ A	AB	C
AB	AB	BC
* BC	A	AB
* C	D	AB
D	D	D

A, {C}] → formal way of representing NFA

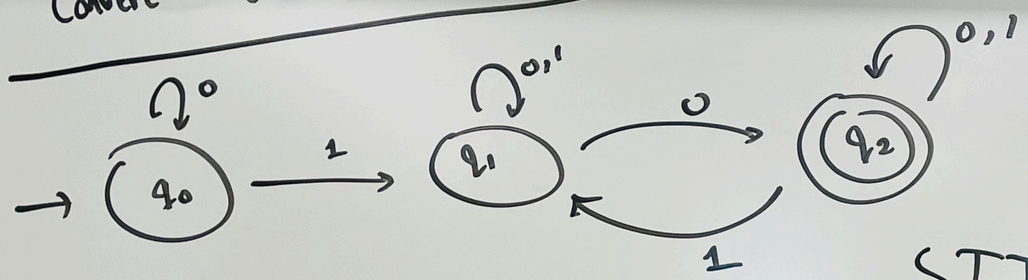
A: Starting state

{C}: final accepting state

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



★ Convert the NFA to DFA



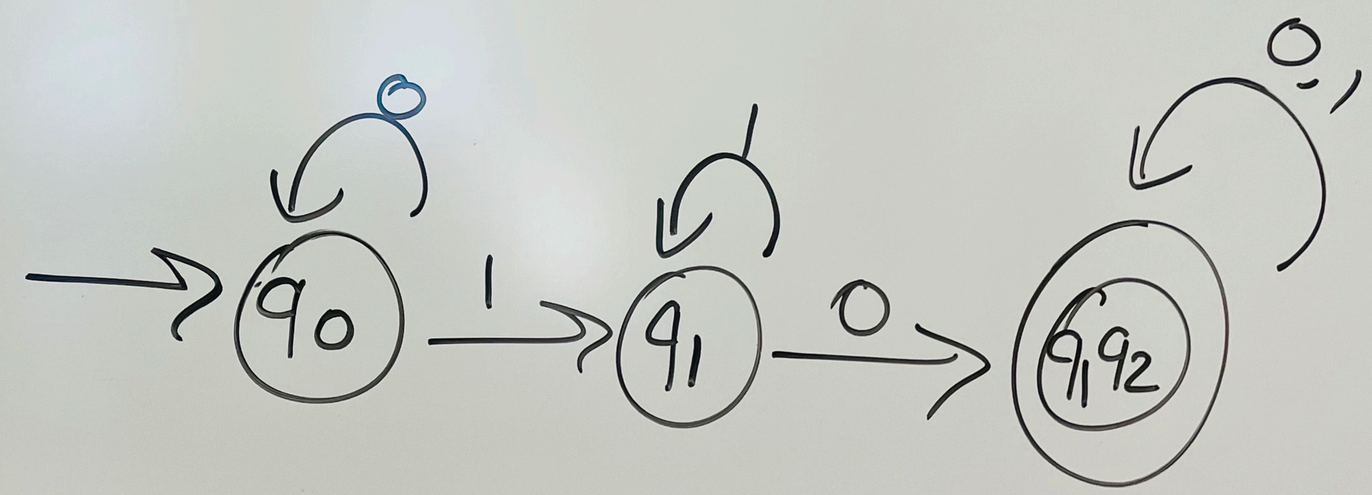
STT NFA

States	Inputs 0	Inputs 1
→ q_0	q_0	q_1
q_1	$\{q_1, q_2\}$	q_1
* q_2	q_2	$\{q_2, q_1\}$

STT DFA

States	Inputs 0	Inputs 1
→ q_0	q_0	q_1
q_1	$\{q_1, q_2\}$	$\{q_1\}$
* $q_1 q_2$	$\{q_1, q_2\}$	$\{q_1, q_2\}$

Please Erase!
(Boards After)

STT DFA

	Inputs	
States	0	1
$\rightarrow q_0$	q_0	q_1
q_1	$\{q_1, q_2\}$	$\{q_1\}$
$* \; q_1 q_2$	$\{q_1, q_2\}$	$\{q_1, q_2\}$