NFA → DFA Conversion

- Process of NFA to DFA conversion
- Practice problems

1. Problem: Draw an NFA that accepts all the strings ending with "1" over Σ {0,1} and convert this NFA to its corresponding DFA.
2. Problem: Draw an NFA of all binary strings in which the 2nd last bit is 1 and convert this NFA to its corresponding DFA.
3. Problem: Draw an NFA that ends with 01 and convert this NFA to its corresponding DFA.
4. Problem: Convert this NFA to its corresponding DFA.
5. Problem: Convert this NFA to its corresponding DFA.
6. Problem: Convert this NFA to its corresponding DFA.
7. Problem: Convert this NFA to its corresponding DFA.
8. Problem: Convert this NFA to its corresponding DFA.
9. Problem: Convert this NFA to its corresponding DFA.

NFA → DFA Conversion

❑ Converting NFA to DFA Algorithm

There are four basic steps for the conversion of NFA to DFA.

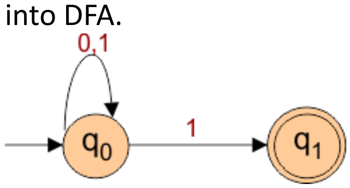
- **Step 01:** Draw an NFA Graph or diagram **if it is not given**.
- **Step 02:** Draw the NFA transition Table.
- **Step 03:** Convert the NFA transition Table to the DFA transition Table.
  - If any **new state** appears while converting the NFA to DFA then it will be added to the state column of the DFA Table.
  - If "x" was the final state in NFA, then all those states will be the final in DFA where "x" exists.
- **Step 04:** Convert the DFA table to the DFA Diagram.

Example 1: NFA to DFA Conversion

Problem: Draw an NFA that accepts all the strings **ending with "1" over Σ {0,1} and** convert this NFA to its corresponding DFA.

Step 01: Draw NFA Graph

The following is the NFA graph for conversion into DFA.



Step 02: Draw the NFA transition Table.

The NFA transition table of the above NFA is given below

States	0	1
→ q0	q0	q0q1
*q1	-	-

NFA Transition Table

Step 03: Conversion of NFA To DFA transition Table

Initially, the NFA table has three states**: q0 and q1** Any new state is added to the DFA states column.

First, Select the **first two rows** of the **NFA transition table**, which will become the first two rows of the **DFA transition table**. It is given below

States	0	1
q0	q0	q0q1

The q0q1 is a new state and will be added to the states column. Following is the Transitions for **q0q1.**

δ([q0q1], 0) = δ(q0, 0) ∪ δ(q1, 0)
= {q0} ∪ {ϕ}
= **{q0}**

δ([q0q1], 1) = δ(q0, 1) ∪ δ(q1, 1)
= {q0q1} ∪ {ϕ}
= **{q0q1}**

NFA → DFA Conversion

The transition of q0q1 against **input 0** is q0, and the transition against **input 1** is q0q1. Hence, the updated DFA table is given below.

States	0	1
q0	q0	q0q1
q0q1	q0	q0q1

DFA Transition Table

At this stage, **all newly generated states** are executed **successfully** for their transitions. So, the DFA table is ready.

Important: As q1 was the final state in NFA Table. That's why, all those states will be the **final states** where q1 is present. Simply mark with "*" to represent the final state.

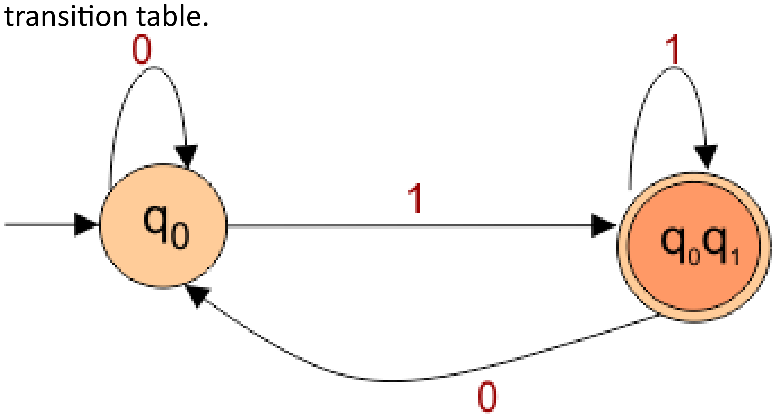
So, the following is the updated table of DFA with its final states

States	0	1
→ q0	q0	q0q1
*q0q1	q0	q0q1

DFA Transition Table

Step 4: Now draw DFA according to the DFA transition table

The DFA transition table contains **two states (q0, q0q1)** in its state column. Thus, the desired DFA machine graph will contain these similar states, and transitions will be made according to the DFA transition table.



DFA According to Given NFA
With Final State

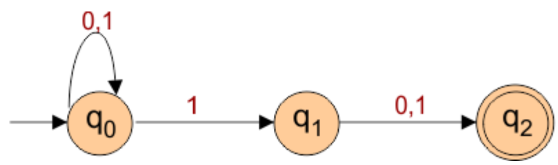
NFA → DFA Conversion

Example 02: NFA to DFA Conversion

Problem: Draw an NFA of all binary strings in which the **2nd last bit is 1 and** convert this NFA to its corresponding DFA.

Step 01: Draw NFA Graph.

The following is the NFA graph that has to be converted into DFA.



Step 02: Draw the NFA transition Table.

The NFA transition table of the above NFA is given below

States	0	1
→ q0	q0	q0q1
q1	q2	q2
*q2	-	-

NFA Transition Table

Step 03: Conversion of NFA To DFA transition Table

Initially, the NFA table has three states**: q0, q1, and q2.** Any new state is added to the DFA states column. **First,** Select the **first two rows** of the **NFA transition table**, which will become the first two rows of the **DFA transition table**. It is given below

States	0	1
q0	q0	q0q1

The q0q1 is a new state and will be added to the states column. Following is the Transitions for **q0q1.**

δ([q0q1], 0) = δ(q0, 0) ∪ δ(q1, 0)
= {q0} ∪ {q2}
= **[q0q2] // new state generated**
δ([q0q1], 1) = δ(q0, 1) ∪ δ(q1, 1)
= {q0q1} ∪ {q2}
= **{q0q1q2} // new state generated**

The transition of q0q1 against **input 0** is q0q2, and the transition against **input 1** is q0q1q2. Hence, the updated DFA table is given below.

States	0	1
q0	q0	q0q1
q0q1	q0q2	q0q1q2

q0q2 and q0q1q2 are the new states for the states column. We need to find the transitions for **both** newly generated states. Following is the Transitions for **q0q2.**

δ([q0q2], 0) = δ(q0, 0) ∪ δ(q2, 0)
= {q0}
= [q0] **// already found in DFA table**
δ([q0q2], 1) = δ(q0, 1) ∪ δ(q2, 1)
= {q0q1}
= {q0q1}
= [q0q1] **// already found in DFA table**

The transition of q0q2 against **input 0** is q0, and the transition against **input 1** is q0q1. Hence, the updated DFA table is given below.

NFA → DFA Conversion

States	0	1
q0	q0	q0q1
q0q1	q0q2	q0q1q2
q0q2	q0	q0q1

Following is the Transitions for **q0q1q2.**

δ([q0q1q2], 0) = δ(q0, 0) ∪ δ(q1, 0) ∪ δ(q2, 0)

= {q0} ∪ {} ∪ {q2}

= [q0q2] **// already found in DFA table**

δ([q0q1q2], 1) = δ(q0, 1) ∪ δ(q1, 1) ∪ δ(q2, 1)

= {q0} ∪ {q1} ∪ {q2}

= [q0q1q2] **// already found in DFA table**

The transition of q0q1q2 against **input 0** is q0q2, and the transition against **input 1** is q0q1q2. Hence, the updated DFA table is given below.

States	0	1
q0	q0	q0q1
q0q1	q0q2	q0q1q2
q0q2	q0	q0q1
q0q1q2	q0q2	q0q1q2

DFA Transition Table

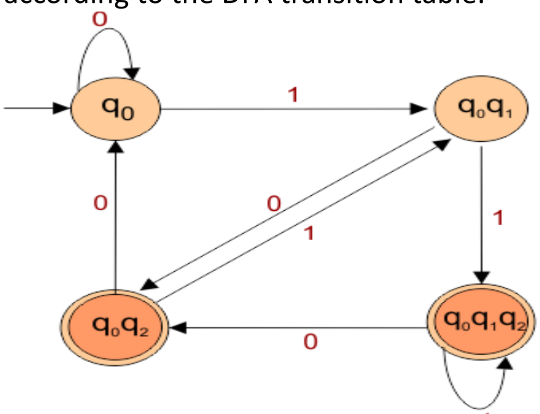
At this stage, **all newly generated states** are executed **successfully** for their transitions. So, the DFA table is ready. **Important:** As **q2** was the final state in NFA, all those states will be the final in DFA where q2 exists. Therefore, the DFA Table with all its final states is given under,

States	0	1
→ q0	q0	q0q1
q0q1	q0q2	q0q1q2
*q0q2	q0	q0q1
*q0q1q2	q0q2	q0q1q2

DFA Transition Table

Step 4: Now draw DFA according to the DFA transition table

The DFA transition table contains **four states (q0, q0q1, q0q2, q0q1q2)** in its state column. Thus, the desired DFA machine graph will contain these similar states, and transitions will be made according to the DFA transition table.



DFA According to Given NFA
With Final States

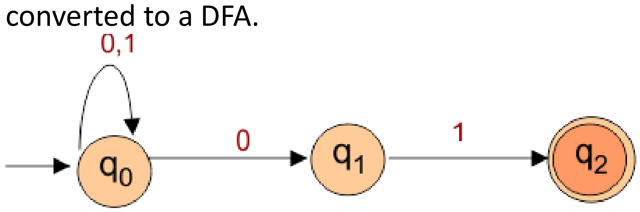
NFA → DFA Conversion

Example 3: NFA to DFA Conversions

Problem: Draw an NFA that ends with 01 **and** convert this NFA to its corresponding DFA.

Step 01: Draw NFA Graph

The following is the NFA graph that needs to be converted to a DFA.



Step 02: Draw the NFA transition Table.

The following is the NFA transition table which is derived from the given NFA.

States	0	1
→ q0	q0q1	q0
q1	ϕ	q2
*q2	ϕ	ϕ

NFA Transition Table

Step 03: Conversion of NFA To DFA transition Table

Initially, the NFA table has three states**: q0 and q1** Any new state is added to the DFA states column.

First, Select the **first two rows** of the **NFA transition table**, which will become the first two rows of the **DFA transition table**. It is given below

States	0	1
q0	q0q1	q0

The q0q1 is a new state (as it does not exist in the NFA table) and will be added to the states column of the DFA table. Following is the Transitions for **q0q1.**

δ([q0q1], 0) = δ(q0, 0) ∪ δ(q1, 0)
= {q0q1} ∪ {ϕ}
= **{q0q1}**

δ([q0q1], 1) = δ(q0, 1) ∪ δ(q1, 1)
= {q0} ∪ {q2}
= **{q0q2}**

Following is the updated DFA table

States	0	1
q0	q0q1	q0
q0q1	q0q1	q0q2

NFA → DFA Conversion

The q0q2 is a new state and will be added to the states column. Following is the Transitions for **q0q2.**

δ([q0q2], 0) = δ(q0, 0) ∪ δ(q2, 0)
= {q0q1} ∪ {ϕ}
= **{q0q1}**

δ([q0q2], 1) = δ(q0, 1) ∪ δ(q2, 1)
= {q0} ∪ {ϕ}
= **{q0}**

So, the following is the updated table of DFA with its final states

States	0	1
→ q0	q0q1	q0
q0q1	q0q1	q0q2
*q0q2	q0q1	q0

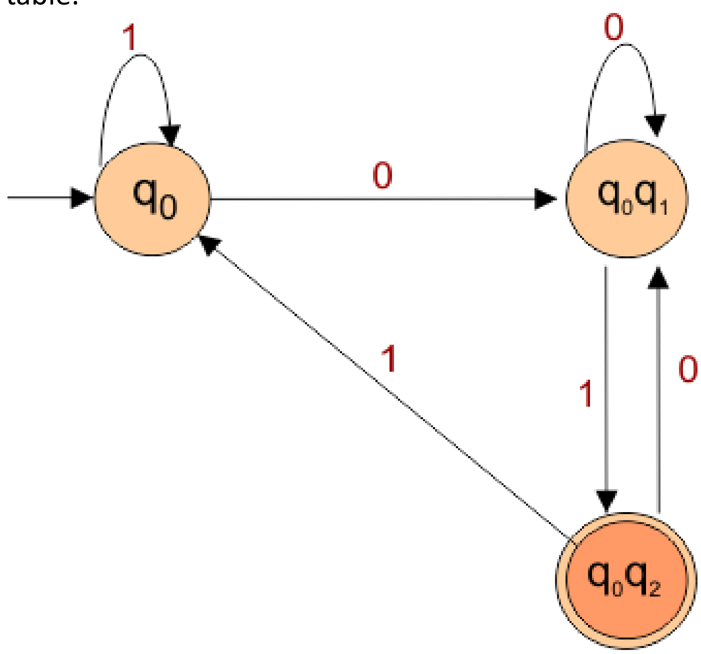
DFA Transition Table

All newly generated states are executed successfully for their transitions at this stage. So, the DFA table is ready.

Important: As **q2** was the final state in NFA Table. That's why, all those states will be the **final states** where q2 is present. Simply **mark with "*"** to represent the final state.

Step 4: Now draw DFA according to the DFA transition table

The DFA machine graph will include similar states and transitions based on the DFA transition table.



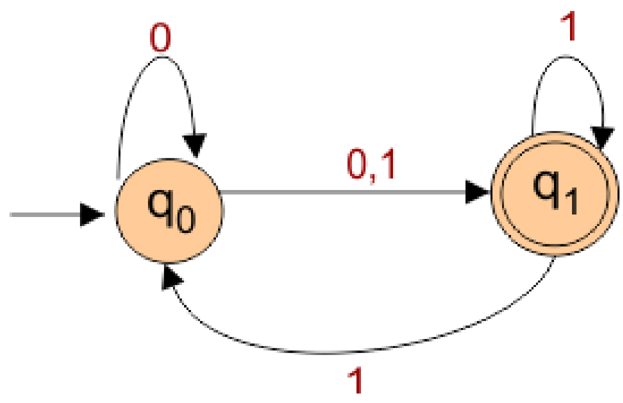
NFA → DFA Conversion

Example 4: NFA to DFA Conversions

Problem: Convert this NFA to its corresponding DFA.

Step 01: Draw NFA Graph

The following is the NFA graph that needs to be converted to a DFA.



Step 02: Draw the NFA transition Table.

The following is the NFA transition table which is derived from the given NFA.

States	0	1
→ q0	q0q1	q1
*q1	ϕ	q0q1

NFA Transition Table

Step 03: Conversion of NFA To DFA transition Table

Initially, the NFA table has three states**: q0 and q1** Any new state is added to the DFA states column.

First, Select the **first two rows** of the **NFA transition table**, which will become the first two rows of the **DFA transition table**. It is given below

States	0	1
q0	q0q1	q1

The q0q1 is a new state (as it does not exist in the NFA table) and will be added to the states column of the DFA table. Following is the Transitions for **q0q1.**

δ([q0q1], 0) = δ(q0, 0) ∪ δ(q1, 0)
= {q0q1} ∪ {ϕ}
= **{q0q1}**

δ([q0q1], 1) = δ(q0, 1) ∪ δ(q1, 1)
= {q1} ∪ {q0q1}
= **{q0q1}**

Following is the updated DFA table

States	0	1
q0	q0q1	q1
q0q1	q0q1	q0q1

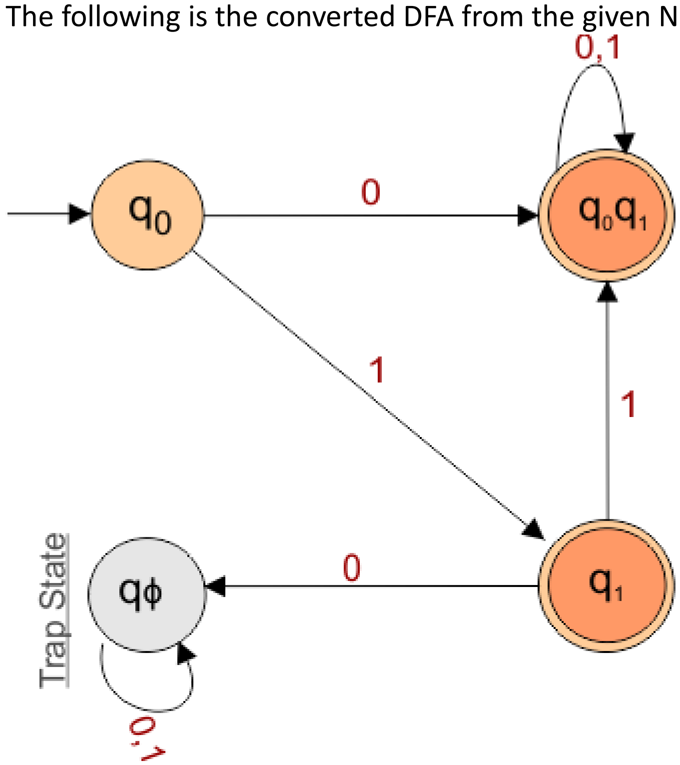
NFA → DFA Conversion

The q1 is a new state and will be added to the states column. Following is the Transitions for **q1.**
δ([q1], 0) = **{qϕ}**
δ([q1], 1) = **{q0q1}**
The following is the updated DFA table with all its final states

States	0	1
→ q0	q0q1	q1
*q0q1	q0q1	q0q1
*q1	qϕ	q0q1

DFA Transition Table

The above given is the required DFA table
Important: As **q1** was the final state in NFA Table. That's why, all those states will be the **final states** where q1 is present. Simply **mark with "*"** to represent the final state.
Step 4: Now draw DFA according to the DFA transition table
The following is the converted DFA from the given NFA.



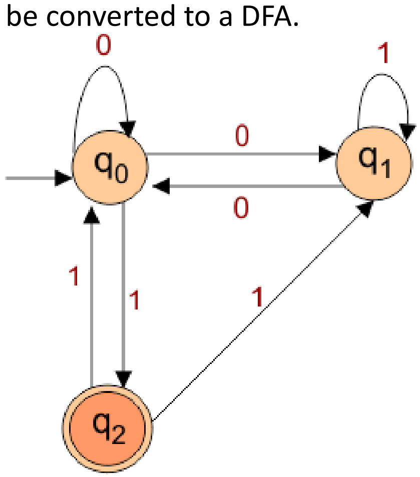
NFA → DFA Conversion

Example 5: NFA to DFA Conversions

Problem: Convert this NFA to its corresponding DFA.

Step 01: Draw NFA Graph

The following is the NFA graph that needs to be converted to a DFA.



Step 02: Draw the NFA transition Table.

The following is the NFA transition table which is derived from the given NFA.

States	0	1
→ q0	q0q1	q2
q1	q0	q1
*q2	ϕ	q0q1

NFA Transition Table

Step 03: Conversion of NFA To DFA transition Table

Initially, the NFA table has three states**: q0 and q1** Any new state is added to the DFA states column.

First, Select the **first two rows** of the **NFA transition table**, which will become the first two rows of the **DFA transition table**. It is given below

States	0	1
q0	q0q1	q2

The q0q1 is a new state (as it does not exist in the NFA table) and will be added to the states column of the DFA table. Following is the Transitions for **q0q1.**

δ([q0q1], 0) = δ(q0, 0) ∪ δ(q1, 0)
= {q0q1} ∪ {q0}
= **{q0q1}**

δ([q0q1], 1) = δ(q0, 1) ∪ δ(q1, 1)
= {q2} ∪ {q1}
= **{q1q2}**

Following is the updated DFA table

States	0	1
q0	q0q1	q2
q0q1	q0q1	q1q2

The q0q2 is a new state (as it does not exist in the NFA table) and will be added to the states column of the DFA table. Following is the Transitions for **q1q2.**

NFA → DFA Conversion

$\delta([q1q2], 0) = \delta(q1, 0) \cup \delta(q2, 0)$
$= \{q0\} \cup \{\boldsymbol{\phi}\}$
$= \mathbf{\{q0\}}$
$\delta([q1q2], 1) = \delta(q1, 1) \cup \delta(q2, 1)$
$= \{q1\} \cup \{q0q1\}$
$= \mathbf{\{q0q1\}}$

Following is the updated DFA table

States	0	1
q0	q0q1	q2
q0q1	q0q1	q1q2
q1q2	q0	q0q1

The q2 is the state in the above table which is still not expressed in the state column. Simply add it also for its transitions.

$\delta([q2], 0) = \mathbf{\{q\phi\}}$
$\delta([q2], 1) = \mathbf{\{q0q1\}}$

Following is the updated DFA table

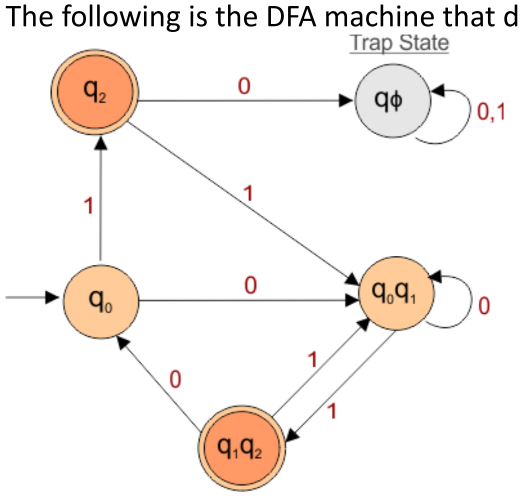
States	0	1
→ q0	q0q1	q2
q0q1	q0q1	q1q2
*q1q2	q0	q0q1
*q2	qϕ	q0q1

DFA Transition Table

The DFA table is ready.
Important: As **q2** was the final state in NFA Table. That's why, all those states will be the **final states** where q2 is present. Simply **mark with "*"** to represent the final state.

Step 4: Now draw DFA according to the DFA transition table

The following is the DFA machine that derives from the given NFA.



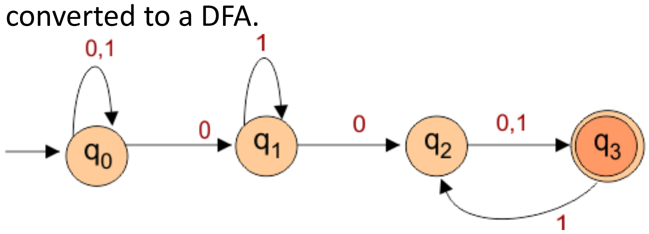
NFA → DFA Conversion

Example 6: Convert NFA to DFA

Problem: Convert this NFA to its corresponding DFA.

Step 01: Draw NFA Graph

The following is the NFA graph that needs to be converted to a DFA.



Step 02: Draw the NFA transition Table.

The following is the NFA transition table which is derived from the given NFA.

States	0	1
→ q0	q0q1	q0
q1	q2	q1
q2	q3	q3
*q3	ϕ	q2

NFA Transition Table

Step 03: Conversion of NFA To DFA transition Table

Initially, the NFA table has three states**: q0 and q1** Any new state is added to the DFA states column. **First,** Select the **first two rows** of the **NFA transition table**, which will become the first two rows of the **DFA transition table**. It is given below

States	0	1
q0	q0q1	q0

The q0q1 is a new state (as it does not exist in the NFA table) and will be added to the states column of the DFA table. Following is the Transitions for **q0q1.**

δ([q0q1], 0) = δ(q0, 0) ∪ δ(q1, 0)
= {q0q1} ∪ {q2}
= **{q0q1q2}**

δ([q0q1], 1) = δ(q0, 1) ∪ δ(q1, 1)
= {q0} ∪ {q1}
= **{q0q1}**

Following is the updated DFA table

States	0	1
q0	q0q1	q0
q0q1	q0q1q2	q0q1

The q0q1q2 is a new state (as it does not exist in the NFA table) and will be added to the states column of the DFA table. Following is the Transitions for **q1q2.**

δ([q0q1q2], 0) = δ(q0, 0) ∪ δ(q1, 0) ∪ δ(q2, 0)
= {q0q1} ∪ {q2} ∪ {q3}
= **{q0q1q2q3}**

δ([q0q1q2], 1) = δ(q0, 1) ∪ δ(q1, 1) ∪ δ(q2, 1)
= {q0} ∪ {q1} ∪ {q3}
= **{q0q1q3}**

Following is the updated DFA table

NFA → DFA Conversion

States	0	1
q0	q0q1	q0
q0q1	q0q1q2	q0q1
q0q1q2	q0q1q2q3	q0q1q3

The q0q1q2q3 is a new state

δ([q0q1q2q3], 0) = δ(q0, 0) ∪ δ(q1, 0) ∪ δ(q2, 0) ∪ δ(q3, 0)
= {q0q1} ∪ {q2} ∪ {q3} ∪ {**ϕ**}
= **{q0q1q2q3}**

δ([q0q1q2q3], 1) = δ(q0, 1) ∪ δ(q1, 1) ∪ δ(q2, 1) ∪ δ(q3, 1)
= {q0} ∪ {q1} ∪ {q3} ∪ {q2}
= **{q0q1q2q3}**

Following is the updated DFA table

States	0	1
q0	q0q1	q0
q0q1	q0q1q2	q0q1
q0q1q2	q0q1q2q3	q0q1q3
q0q1q2q3	q0q1q2q3	q0q1q2q3

The q0q1q3 is the state in the above table which is still not expressed in the state column. Simply add it also for its transitions.

δ([q0q1q3], 0) = δ(q0, 0) ∪ δ(q1, 0) ∪ δ(q3, 0)
= {q0q1} ∪ {q2} ∪ {**ϕ**}
= **{q0q1q2}**

δ([q0q1q3], 1) = δ(q0, 1) ∪ δ(q1, 1) ∪ δ(q3, 1)
= {q0} ∪ {q1} ∪ {q2}
= **{q0q1q2}**

Following is the updated DFA table

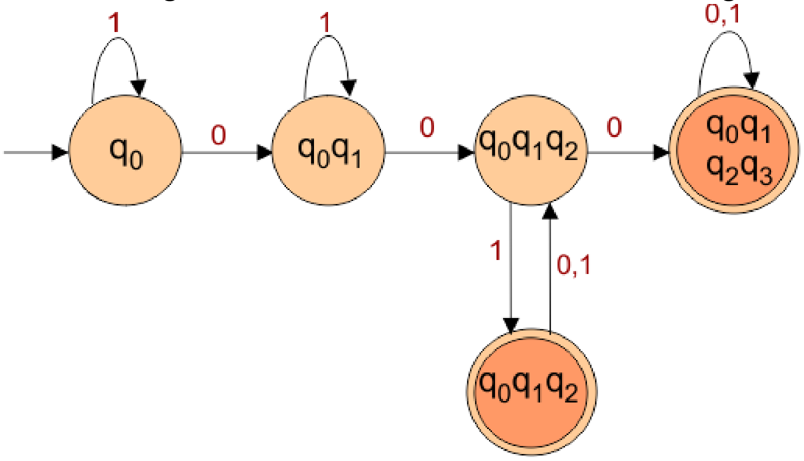
States	0	1
→ q0	q0q1	q0
q0q1	q0q1q2	q0q1
q0q1q2	q0q1q2q3	q0q1q3
*q0q1q2q3	q0q1q2q3	q0q1q2q3
*q0q1q3	q0q1q2	q0q1q2

DFA Transition Table

NFA → DFA Conversion

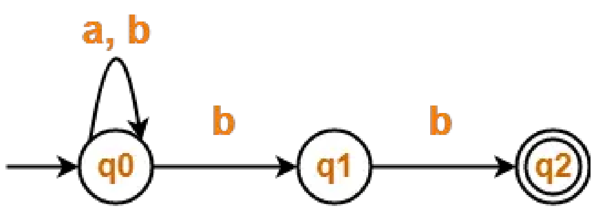
Step 4: Now draw DFA according to the DFA transition table

The following is the DFA machine that derives from the given NFA.



Note: Hence proved that,
DFA has a higher number of states as compared to NFA states. If "n" is the maximum number of states in NFA, then 2^n is the maximum number of states in its corresponding DFA.

Problem-07: Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-



Solution- Transition table for the given Non-Deterministic Finite Automata (NFA) is-

State / Alphabet	a	b
→q0	q0	q0, q1
q1	–	*q2
*q2	–	–

Step-01:
Let Q' be a new set of states of the Deterministic Finite Automata (DFA).
Let T' be a new transition table of the DFA.

Step-02: Add transitions of start state q0 to the transition table T'.

State / Alphabet	a	b
→q0	q0	{q0, q1}

NFA → DFA Conversion

Step-03:
New state present in state Q’ is {q0, q1}.
Add transitions for set of states {q0, q1} to the transition table T’.

State / Alphabet	a	b
→q0	q0	{q0, q1}
{q0, q1}	q0	{q0, q1, q2}

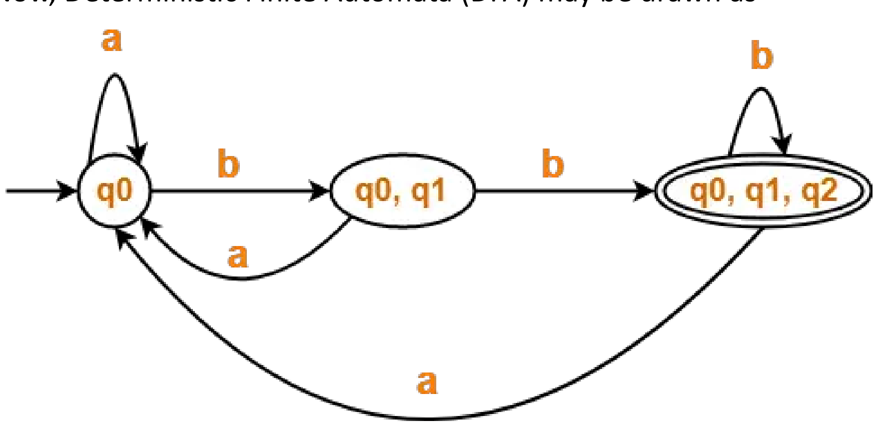
Step-04:
New state present in state Q’ is {q0, q1, q2}.
Add transitions for set of states {q0, q1, q2} to the transition table T’.

State / Alphabet	a	b
→q0	q0	{q0, q1}
{q0, q1}	q0	{q0, q1, q2}
{q0, q1, q2}	q0	{q0, q1, q2}

Step-05:
Since no new states are left to be added in the transition table T’, so we stop.
States containing q2 as its component are treated as final states of the DFA.
Finally, Transition table for Deterministic Finite Automata (DFA) is-

State / Alphabet	a	b
→q0	q0	{q0, q1}
{q0, q1}	q0	*{q0, q1, q2}
*{q0, q1, q2}	q0	*{q0, q1, q2}

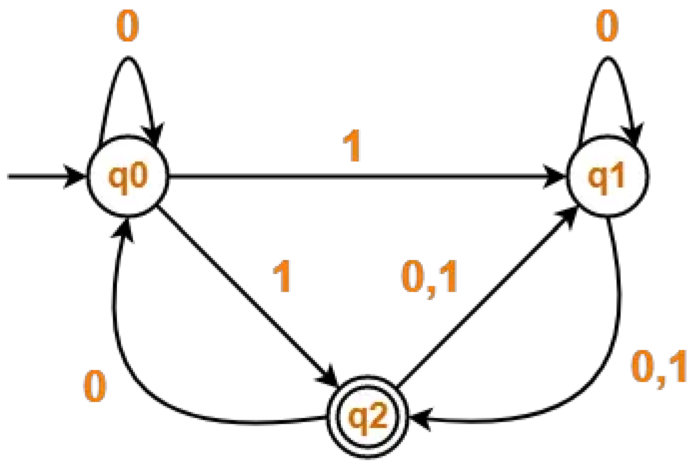
Now, Deterministic Finite Automata (DFA) may be drawn as-



Deterministic Finite Automata (DFA)

NFA → DFA Conversion

Problem-08: Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-



Solution- Transition table for the given Non-Deterministic Finite Automata (NFA) is-

State / Alphabet	0	1
→q0	q0	q1, *q2
q1	q1, *q2	*q2
*q2	q0, q1	q1

Step-01:

Let Q’ be a new set of states of the Deterministic Finite Automata (DFA).
Let T’ be a new transition table of the DFA.

Step-02:
Add transitions of start state q0 to the transition table T’.

State / Alphabet	0	1
→q0	q0	{q1, q2}

Step-03:
New state present in state Q’ is {q1, q2}.
Add transitions for set of states {q1, q2} to the transition table T’.

State / Alphabet	0	1
→q0	q0	{q1, q2}
{q1, q2}	{q0, q1, q2}	{q1, q2}

NFA → DFA Conversion

Step-04:

New state present in state Q’ is {q0, q1, q2}.
Add transitions for set of states {q0, q1, q2} to the transition table T’.

State / Alphabet	0	1
→q0	q0	{q1, q2}
{q1, q2}	{q0, q1, q2}	{q1, q2}
{q0, q1, q2}	{q0, q1, q2}	{q1, q2}

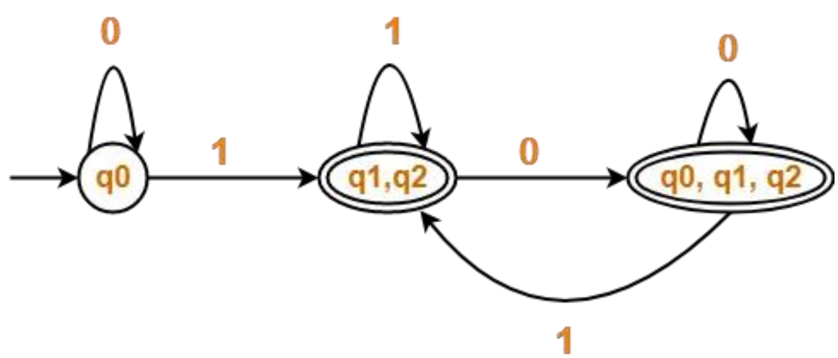
Step-05:

Since no new states are left to be added in the transition table T’, so we stop.
States containing q2 as its component are treated as final states of the DFA.

Finally, Transition table for Deterministic Finite Automata (DFA) is-

State / Alphabet	0	1
→q0	q0	*{q1, q2}
*{q1, q2}	*{q0, q1, q2}	*{q1, q2}
*{q0, q1, q2}	*{q0, q1, q2}	*{q1, q2}

Now, Deterministic Finite Automata (DFA) may be drawn as-

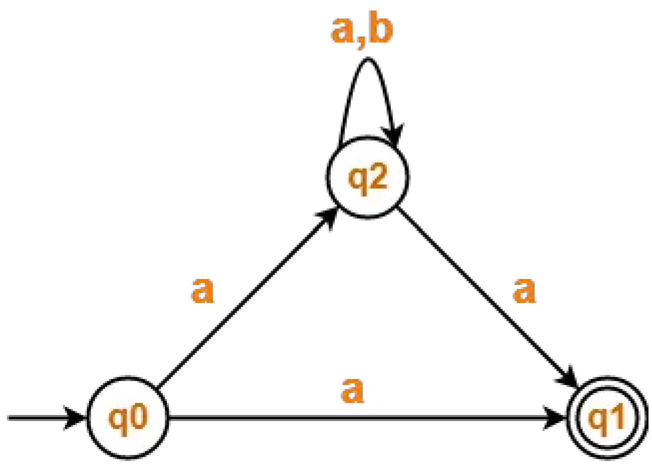


Deterministic Finite Automata (DFA)

NFA → DFA Conversion

Problem-09:

Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA)-



Solution-

Transition table for the given Non-Deterministic Finite Automata (NFA) is-

State / Alphabet	a	b
→q0	*q1, q2	–
*q1	–	–
q2	*q1, q2	q2

Step-01:

Let Q' be a new set of states of the Deterministic Finite Automata (DFA).

Let T' be a new transition table of the DFA.

Step-02:

Add transitions of start state q0 to the transition table T'.

State / Alphabet	a	b
→q0	{q1, q2}	Ø (Dead State)

Step-03:

New state present in state Q' is {q1, q2}.

Add transitions for set of states {q1, q2} to the transition table T'.

State / Alphabet	a	b
→q0	{q1, q2}	Ø
{q1, q2}	{q1, q2}	q2

NFA → DFA Conversion

Step-04:
New state present in state Q' is q2.
Add transitions for state q2 to the transition table T'.

State / Alphabet	a	b
→q0	{q1, q2}	Ø
{q1, q2}	{q1, q2}	q2
q2	{q1, q2}	q2

Step-05:
Add transitions for dead state {Ø} to the transition table T'.

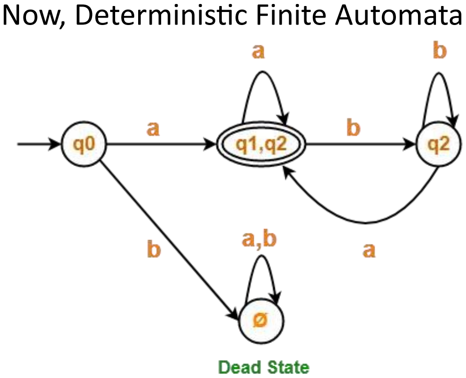
State / Alphabet	a	b
→q0	{q1, q2}	Ø
{q1, q2}	{q1, q2}	q2
q2	{q1, q2}	q2
Ø	Ø	Ø

Step-06:
Since no new states are left to be added in the transition table T', so we stop.
States containing q1 as its component are treated as final states of the DFA.

Finally, Transition table for Deterministic Finite Automata (DFA) is-

State / Alphabet	a	b
→q0	*{q1, q2}	Ø
*{q1, q2}	*{q1, q2}	q2
q2	*{q1, q2}	q2
Ø	Ø	Ø

Now, Deterministic Finite Automata (DFA) may be drawn as-



Deterministic Finite Automata (DFA)

NFA → DFA Conversion

Important Points-

It is important to note the following points when converting a given NFA into a DFA-

Note-01:

After conversion, the number of states in the resulting DFA may or may not be same as NFA.

The maximum number of states that may be present in the DFA are $2^{\text{Number of states in the NFA}}$.

Note-02:

In general, the following relationship exists between the number of states in the NFA and DFA-

$$1 <= n <= 2^m$$

Here,
n = Number of states in the DFA
m = Number of states in the NFA

Note-03:

In the resulting DFA, all those states that contain the final state(s) of NFA are treated as final states.