Theory of Computation(CS-120) Class

On

Theory of Computation

Topic: Equivalence of Deterministic and Nondeterministic Finite Automata

Dated:

Ms. Swati Goel

Department of Computer Science and Engineering,
Chitkara University, Punjab

Introduction

- For every nondeterministic finite automaton (NFA) there exists an equivalent deterministic finite automaton (DFA). It means that for every nondeterministic finite automaton accepting a language L there exist a deterministic finite automaton that accepts the same language L.

- Let, M = (Q, ∑, δ, q0, F) is an NFA which accepts the language L(M). There should be equivalent DFA denoted by M' = (Q', ∑', q0', δ', F') such that L(M) = L(M').

- The DFA equivalent of an NFA simulates the moves of the NFA in parallel. To do that the states of the DFA will be a combination of one or more states of NFA.

- Hence every state of the DFA will be a subset of set of states of the NFA and therefore the transition of NFA to DFA is normally called a **subset construction.**

Conversion from NFA to DFA

CHITKARA UNIVERSITY

Procedure for converting NFA to equivalent DFA

Suppose there is an NFA denoted by M = <Q, ∑, δ, q0, F> which accepts/recognizes a language L. **Then the equivalent DFA** M'= < Q', ∑, δ', q0', F' > which accepts the same language as given NFA; can be constructed as:

Step 1: Initially Q' = ϕ.

Step 2: Add q0 of NFA to Q'.

Step 3: For each state in Q', find the possible set of states for each input symbol using transition function of NFA. If this set of states is not in Q', add it to Q'.

Step 4: Repeat step 3 till new states are there to add in Q', if there is no further new state to add in Q', the process terminates. Final state of DFA will be all states with contain F (final states of NFA).

Equivalence of NFA and DFA

CHITKARA UNIVERSITY

Note :

1. In NFA, if the transition of start state over some input alphabet is null, then perform the transition of start state over that input alphabet to a dead state in the DFA

1. The states which are not reachable from the initial state should not be included in Q'. Thus, the set of states (Q') is not necessary equal to 2^Q.

Problems based on NFA to DFA conversion

Example 1. Consider the following NFA as shown in Figure 1.

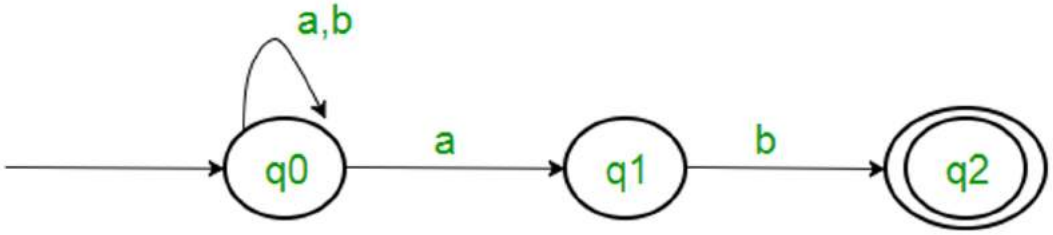


Figure 1

Solution: Following are the various parameters for NFA.

Q = { q0, q1, q2 }

∑ = (a, b)

F = { q2 }

δ (Transition Function of NFA)

State	a	b
q0	q0,q1	q0
q1		q2
q2		

Transition table for given NFA

Steps for NFA to DFA conversion

Step 1: Q' = ϕ
Step 2: Q' = {q0}
Step 3: For each state in Q', find the states for each input symbol.
Currently, state in Q' is q0, find moves from q0 on input symbol a and b using transition function of NFA and update the transition table of DFA.
δ' (Transition Function of DFA)

State	a	b
q0	{q0,q1}	q0

Now { q0, q1 } will be considered as a single state. As its entry is not in Q', add it to Q'.
So Q' = { q0, { q0, q1 } }
Now, moves from state { q0, q1 } on different input symbols are not present in transition table of DFA, we will calculate it like:
δ' ({ q0, q1 }, a) = δ (q0, a) ∪ δ (q1, a) = { q0, q1 }
δ' ({ q0, q1 }, b) = δ (q0, b) ∪ δ (q1, b) = { q0, q2 }
Now we will update the transition table of DFA.
δ' (Transition Function of DFA)

NFA to DFA conversion continued...

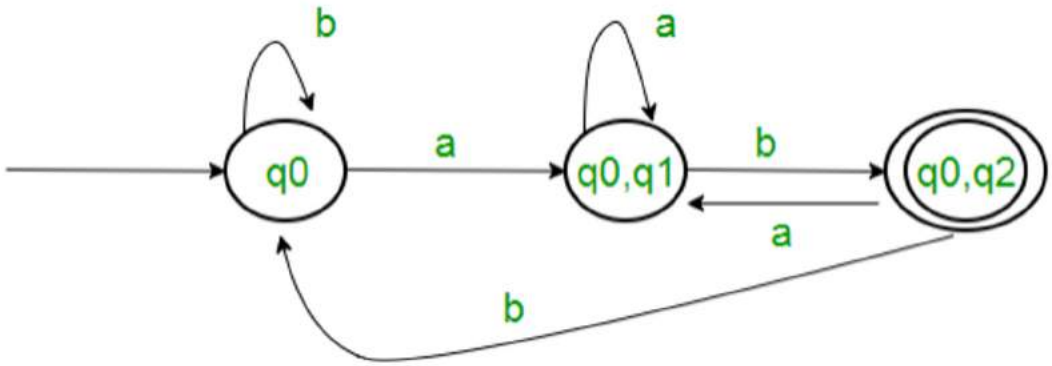
State	a	b
q0	{q0,q1}	q0
{q0,q1}	{q0,q1}	{q0,q2}

Now { q0, q2 } will be considered as a single state. As its entry is not in Q', add it to Q'.
So Q' = { q0, { q0, q1 }, { q0, q2 } }
Now, moves from state {q0, q2} on different input symbols are not present in transition table of DFA, we will calculate it like:
δ' ({ q0, q2 }, a) = δ (q0, a) $\cup$ δ (q2, a) = { q0, q1 }
δ' ({ q0, q2 }, b) = δ (q0, b) $\cup$ δ (q2, b) = { q0 }
Now we will update the transition table of DFA.
δ' (Transition Function of DFA)

State	a	b
q0	{q0,q1}	q0
{q0,q1}	{q0,q1}	{q0,q2}
{q0,q2}	{q0,q1}	q0

NFA to DFA conversion continued...

As there is no new state generated, we are done with the conversion. Final state of DFA will be state which has q2 as its component i.e., { q0, q2 }
Following are the various parameters for DFA.
Q' = { q0, { q0, q1 }, { q0, q2 } }
∑ = (a, b)
F = { { q0, q2 } } and transition function δ' as shown above. The final DFA for above NFA has been shown in Figure below.



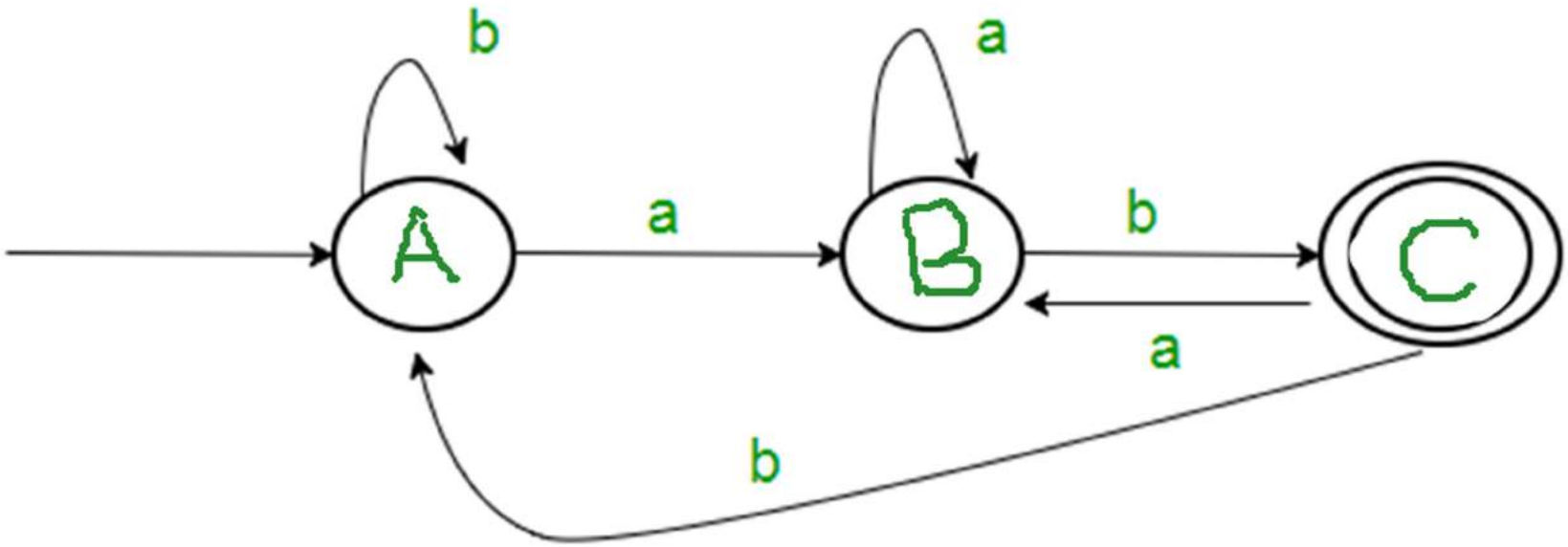
Even we can change the name of the states of DFA. Suppose:
1. A = [q0]
2. B = [q0,q1]
3. C = [q0, q2]
With these new names the DFA will be as follows

NFA to DFA conversion continued...

We can even change the name of the states of DFA. Suppose:

1. A = [q0]
2. B = [q0,q1]
3. C = [q0, q2]

With these new names the DFA will be as follows



Practice Problems

Problem 1. Convert the following NFA to its equivalent DFA.

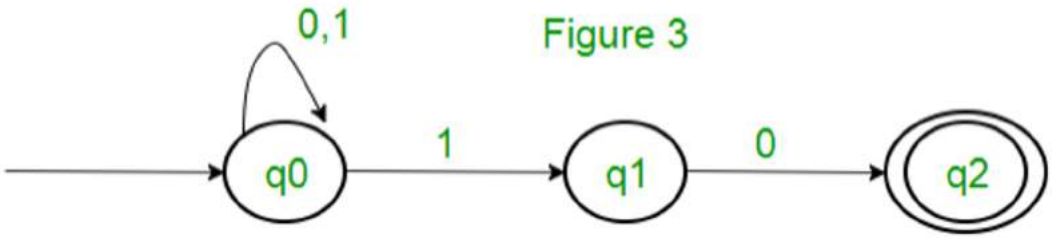
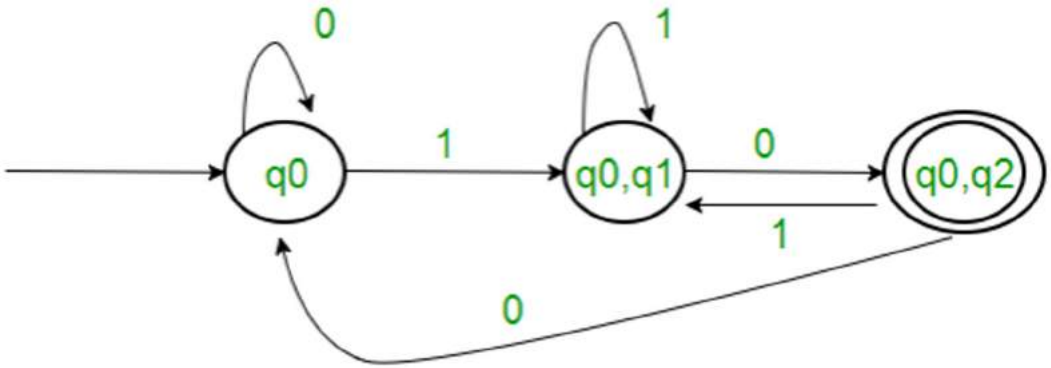


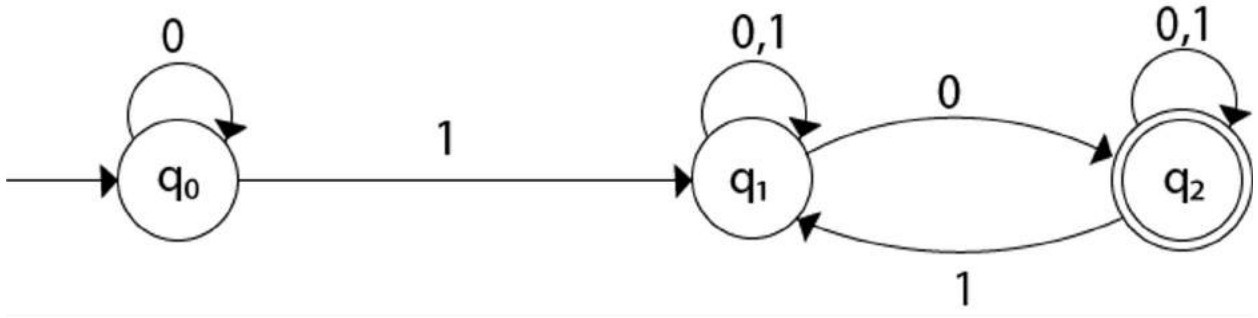
Figure 3

Solution 1:

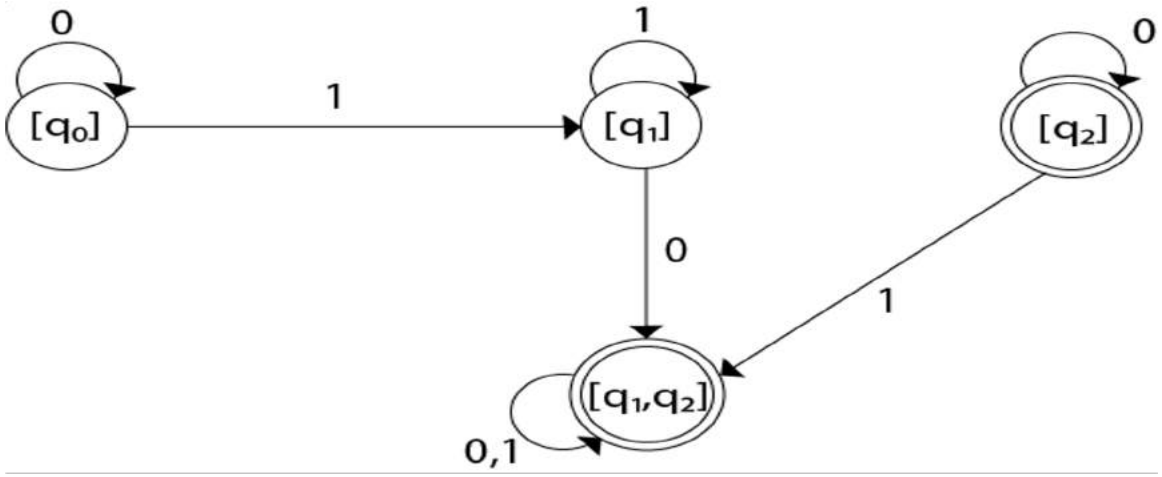


Practice Problems

Problem 2: Convert the given NFA to DFA.



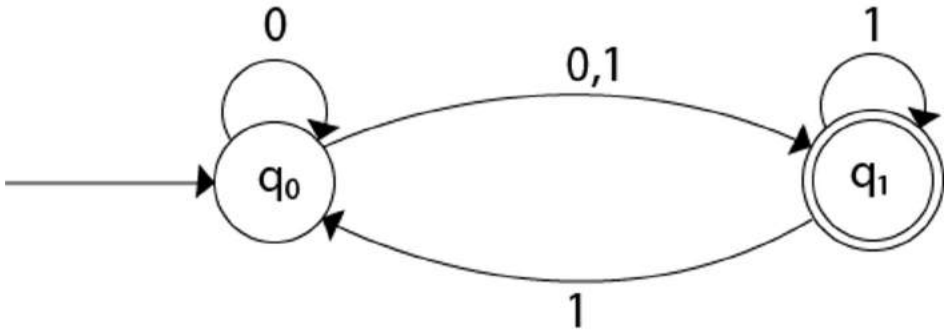
Solution 2:



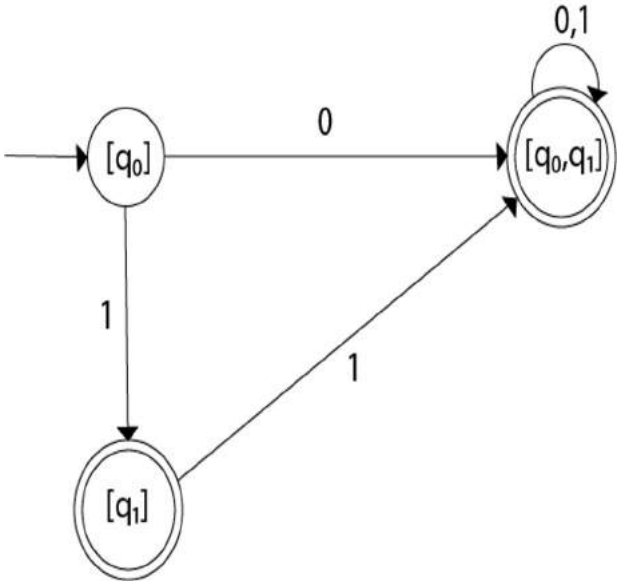
Note : The state q2 can be eliminated because q2 is an unreachable state.

Practice Problems

Problem 3: Convert the given NFA to DFA.



Solution 3:

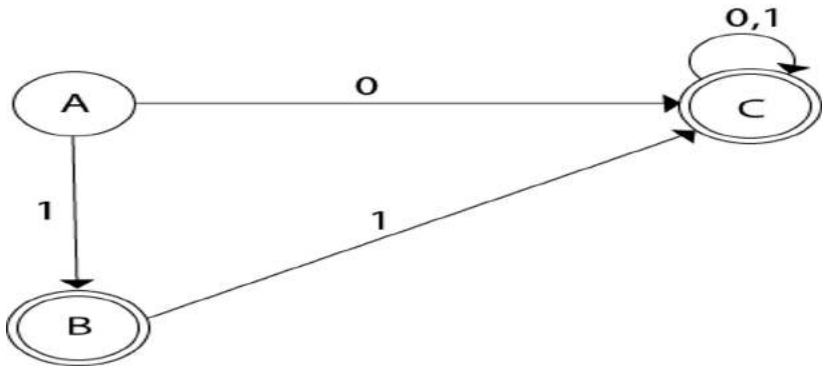


Even we can change the name of the states of DFA.

Suppose

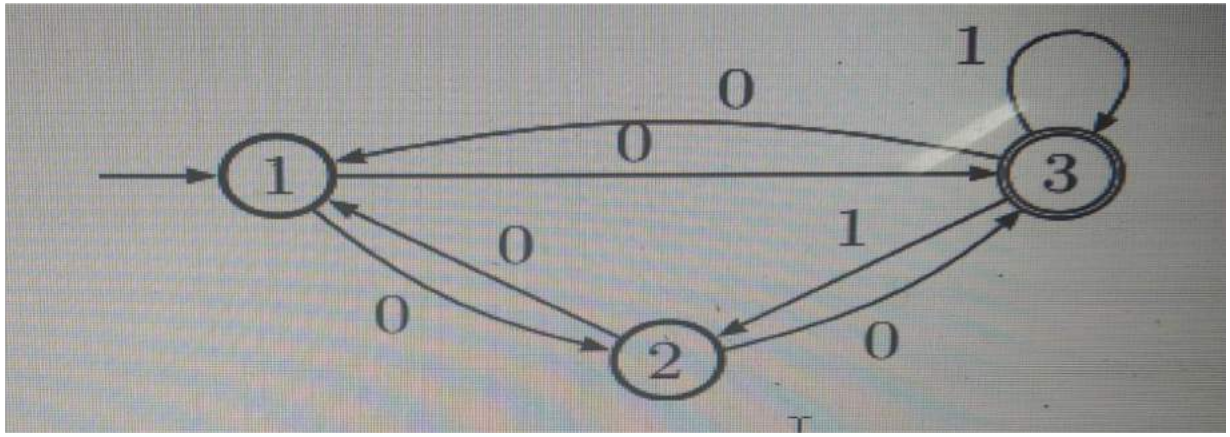
1. A = [q0]
2. B = [q1]
3. C = [q0, q1]

With these new names the DFA will be as follows:

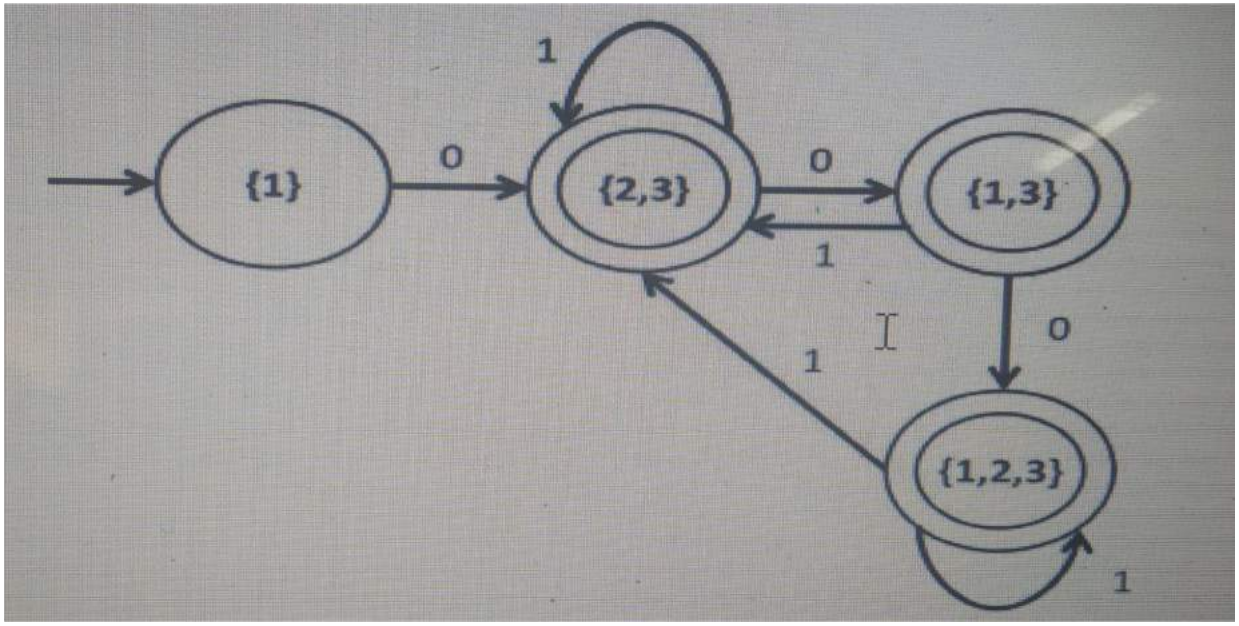


Practice Problems

Problem 4: Convert the given NFA to DFA.

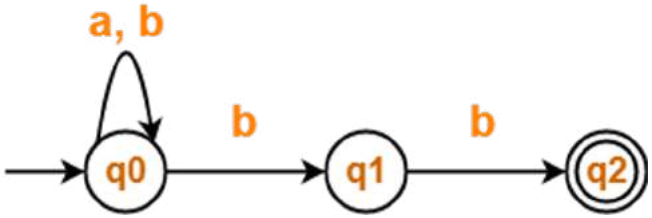


Solution 4:

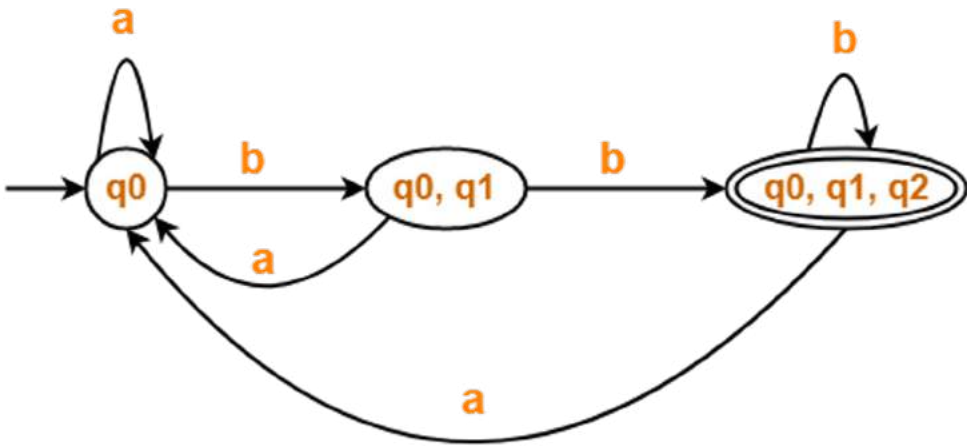


Practice Problems

Problem 5: Convert the following Non-Deterministic Finite Automata (NFA) to Deterministic Finite Automata (DFA):



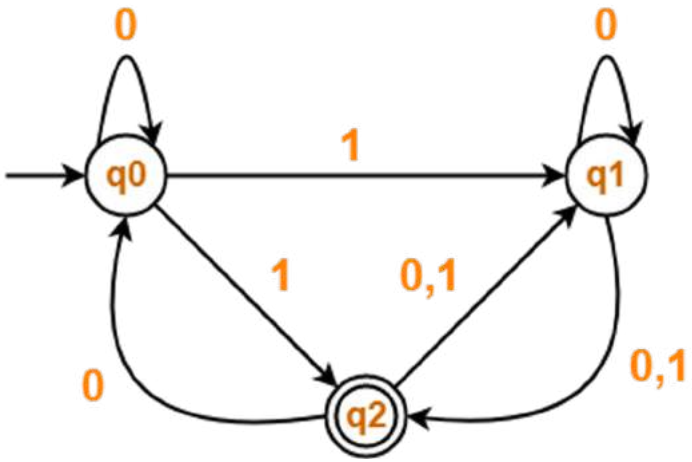
Solution 5:



Deterministic Finite Automata (DFA)

Practice Problems

Problem 6: Convert the following NFA to DFA

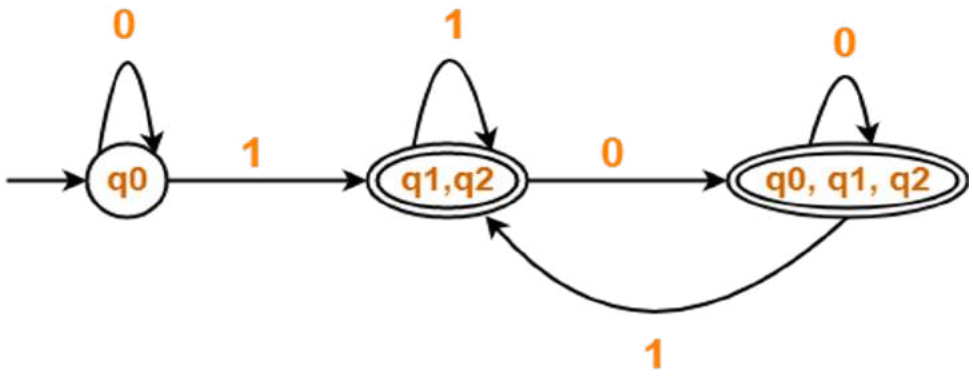


Solution 6:

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

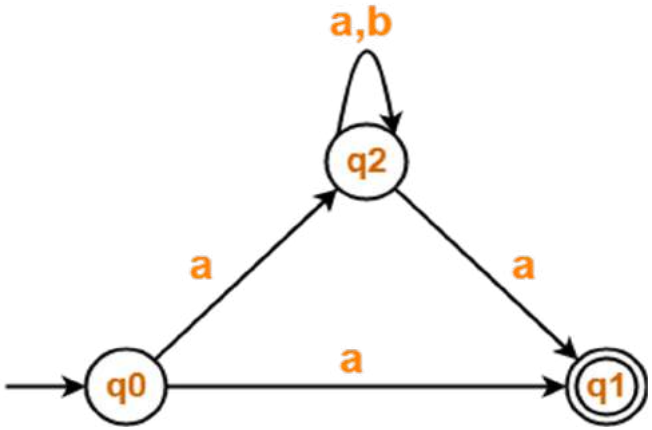
State / Alphabet	0	1
→**q0**	q0	*{q1, q2}
***{q1, q2}**	*{q0, q1, q2}	*{q1, q2}
***{q0, q1, q2}**	*{q0, q1, q2}	*{q1, q2}

Practice Problems



Deterministic Finite Automata (DFA)

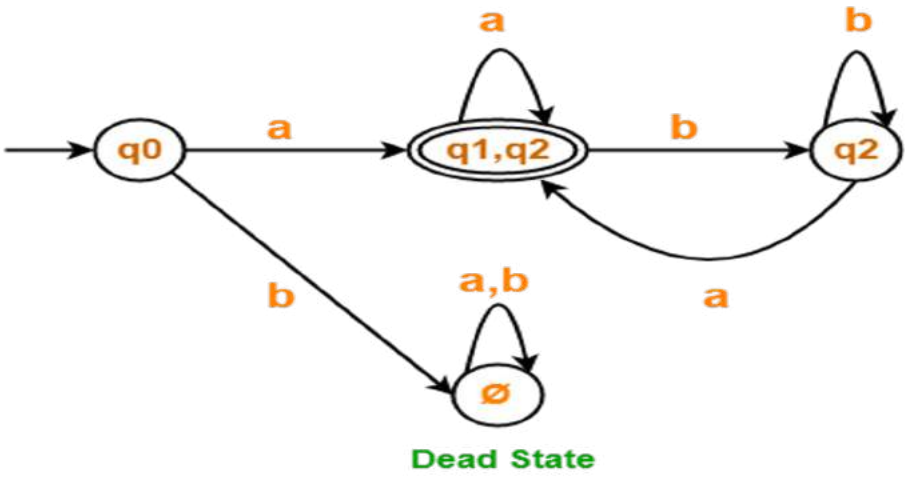
Problem 7: Convert the following NFA to DFA



Practice Problems

Solution 7: 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



Deterministic Finite Automata (DFA)