Formal Languages & Automata Theory

Lecture 5 :

Conversions and Equivalence

Presented by: Dr. Ebtsam Abdelhakam Sayed
Faculty of Computers and Information
Minia University

Topics

1. Conversion from NFA with ε to NFA without ε (ε-NFA)

2. Equivalence between two FSM's

Conversion from NFA with ε to NFA without ε (NFA)

In this method we try to remove all the ε transitions from given NFA. The method will be.

1. Find out all the ε transitions from each state from Q.

   That will be called **as ε – closure {q_i} where q_i ϵ Q**.

2. Then δ' transitions can be obtained. The δ' transitions means an ε – closure on δ moves.

3. Step-2 is repeated for each input symbol and for each state of given NFA without ε can be built.

4. Using the resultant states the transition table for equivalent NFA without ε can be built.

<u>Rule for conversion</u>

$$\delta'(q, a) = \varepsilon - \text{closure}\,(\delta\,(\hat{\delta}\,(q, \varepsilon), a)\,)$$

$$\text{Where } \hat{\delta}\,(q, \varepsilon) = \varepsilon\text{ - closure}\,(q)$$

Before solving some example based on conversion of NFA with ε to NFA A without ε above rule should be remembered.

Example 2.3: Convert the given NFA with ε to NFA without ε.

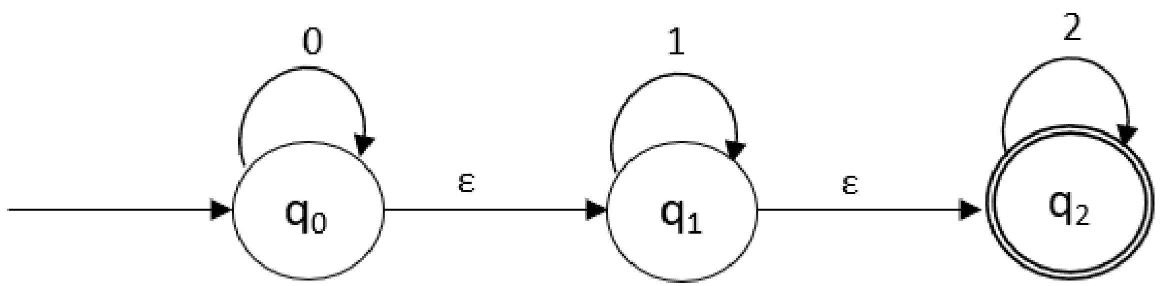


Fig. 2.5

Solution: We will first obtain ε reachable states from current state.

Hence:

ε - closure $(q_0) = \{q_0, q_1, q_2\}$

ε – closure $(q_1) = \{q_1, q_2\}$

ε – closure $(q_2) = \{q_2\}$

As ε – closure (q_0) means with null input (no input symbol) we can reach to q_0, q_1, or q_2. In a similar manner for q_1 and q_2, ε – closures are obtained now we will obtain

Now we will summarize all the computed δ^{-} transitions -

$\delta^{-}(q_0, 0) = (q_0, q_1, q_2)$ $\delta^{-}(q_0, 1) = \{q_1, q_2\}$ $\delta^{-}(q_0, 2) = \{q_2\}$

$\delta^{-}(q_1, 0) = \Phi$ $\delta^{-}(q_1, 1) = \{q_1, q_2\}$ $\delta^{-}(q_1, 2) = \{q_2\}$

$\delta^{-}(q_2, 0) = \Phi$ $\delta^{-}(q_2, 1) = \Phi$ $\delta^{-}(q_2, 2) = \{q_2\}$

From this we can write the transition table as:

State \ Input	0	1	2
$\rightarrow q_0$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	Φ	$\{q_1, q_2\}$	$\{q_2\}$
q_2	Φ	Φ	$\{q_2\}$

The NFA will be:

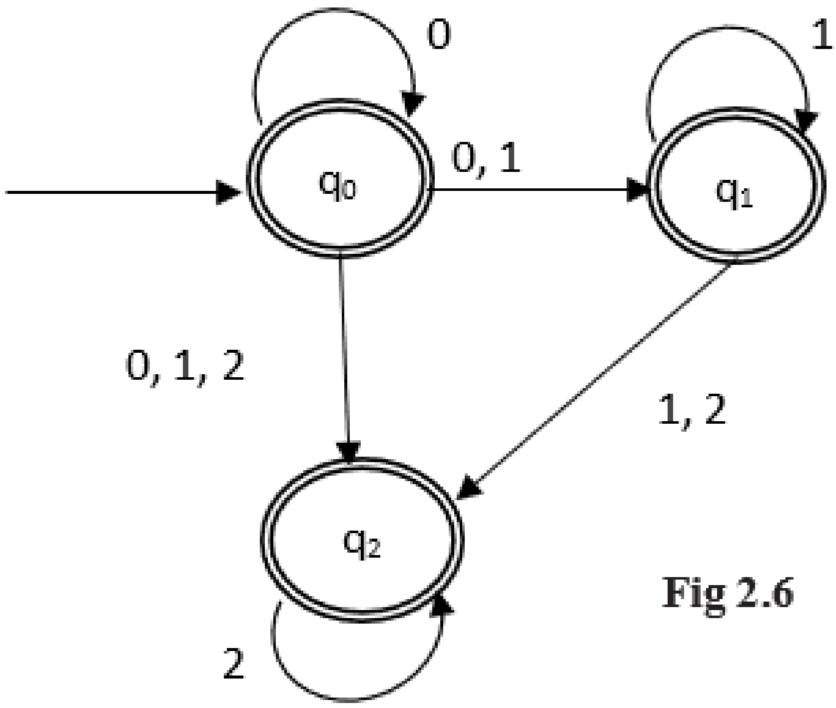


Fig 2.6

Here q_0, q_1 and q_2 is a final state because ε – closure (q_0), ε – closure (q_1) and ε – closure (q_2) contains final state q_2.

Example 2. 5: Convert the following NFA with ε to NFA without ε.

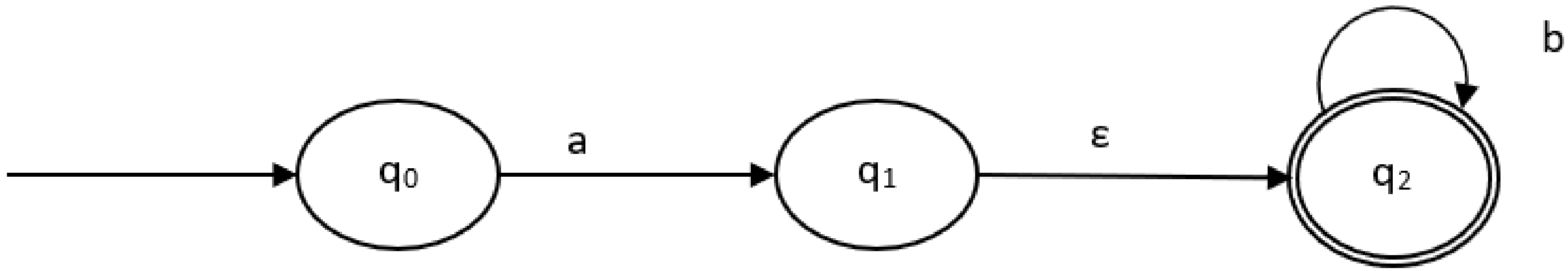


Fig. 2.8

Solution: we will first obtain ε – closure of q_0, q_1 and q_2 as follows.

$$\varepsilon - \text{closure}(q_0) = \{q_0\}$$
$$\varepsilon - \text{closure}(q_1) = \{q_1, q_2\}$$
$$\varepsilon - \text{closure}(q_2) = \{q_2\}$$

New the δ^- transition on each input symbol is obtained as

The transition table can be (Left for reader)

States q_1 and q_2 become the final as ε – closure of q_1 and q_2 contains the final state q_2. The NFA can be shown by following transition diagram.

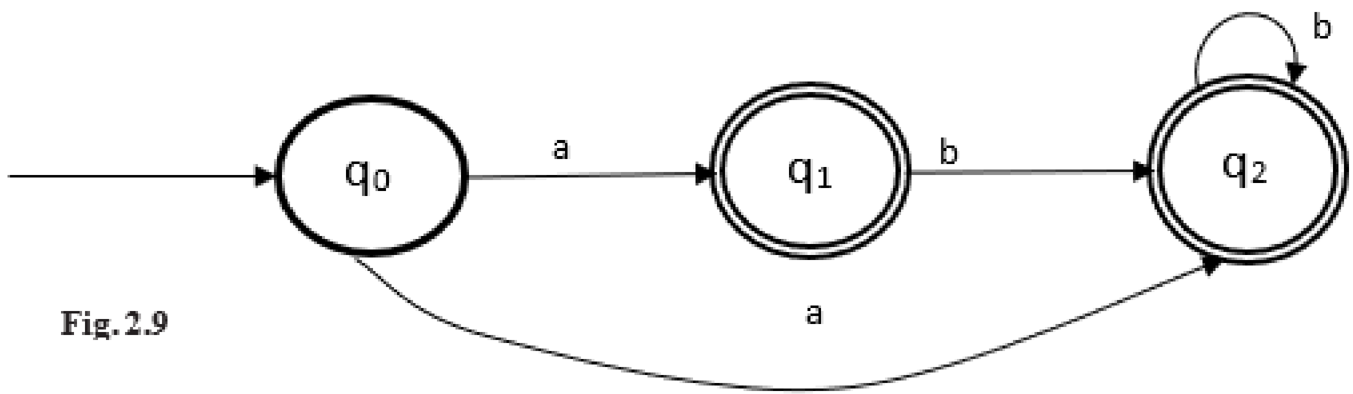
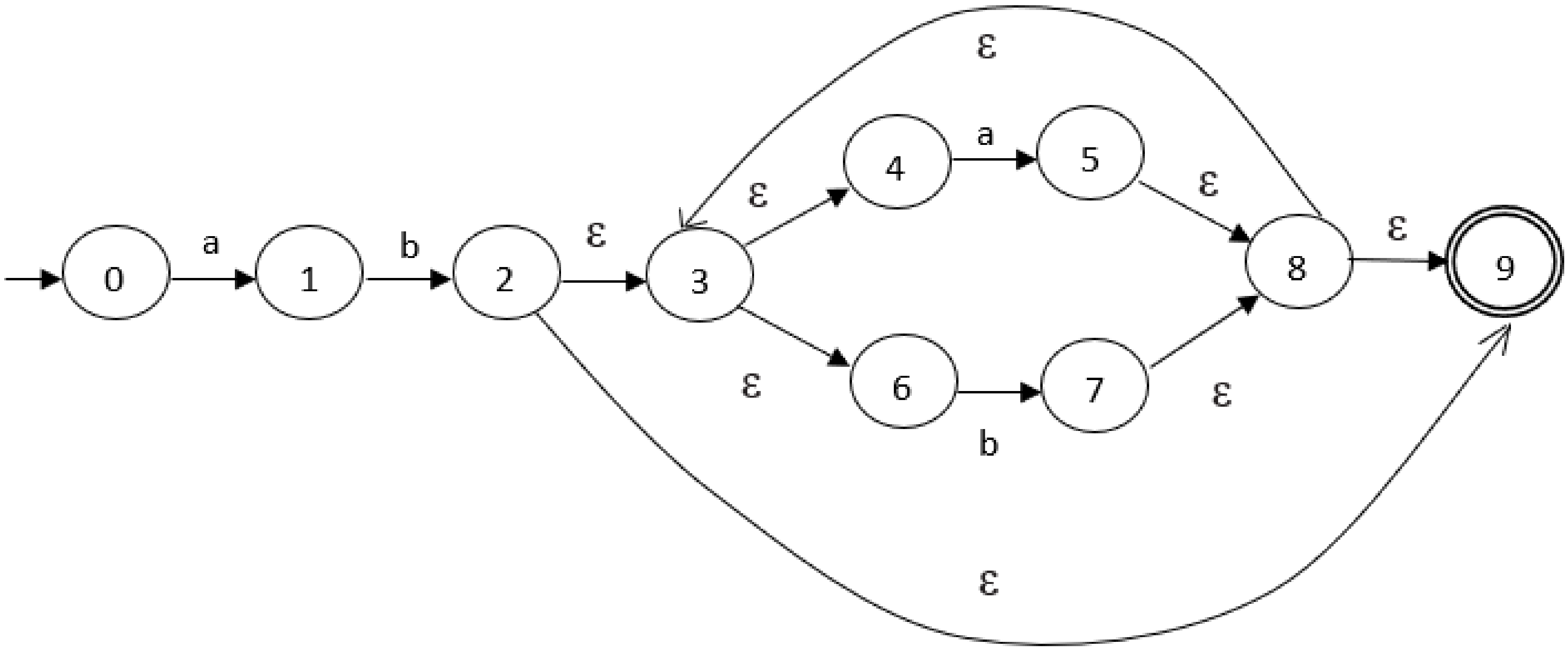


Fig. 2.9

Think and then solve

Example 2.4: Convert the following NFA with ε to NFA without ε.



Step 1: Calculate the ε – closure

Solution: we will first obtain ε-closure for every state.

ε – closure (0)	=	[0]
ε – closure (1)	=	[1]
ε – closure (2)	=	[2, 3, 4, 6, 9]
ε – closure (3)	=	[3, 4, 6]
ε – closure (4)	=	[4]
ε – closure (5)	=	[5, 8, 3, 4,6, 9] = [3, 4, 5, 6, 8, 9] sorted
ε – closure (6)	=	[6]
ε – closure (7)	=	[3,4, 6, 7, 8, 9]
ε – closure (8)	=	[3,4, 6, 8, 9]
ε – closure (9)	=	[9]

Step 3: Construct the transition table

Now we will build the transition table using above calculated δ^{-} transitions.

State \ input	a	b
{0}	{1}	Φ
{1}	Φ	{2, 3, 4, 6, 9}
{2}	{3,4 5, 6, 8, 9}	{3, 4, 6, 7, 8, 9}
{3}	{3, 4, 5, 6,8, 9}	{3, 4, 6, 7, 8, 9}
{4}	{3, 4, 5, 6, 8, 9}	Φ
{5}	{3, 4, 5, 6, 8, 9}	{3, 4, 6, 7, 8, 9}
{6}	Φ	{3, 4, 6, 7, 8, 9}
{7}	{3, 4, 5, 6, 8, 9}	{3, 4, 6, 7, 8, 9}
{8}	{3, 4, 5, 6, 8, 9}	{3, 4, 6, 7, 8, 9}
{9}	Φ	Φ

Equivalence between two FSM's

- The two finite automata are said to be equivalent **if both the automata accept the same set of strings (language) over an input set Σ.**
- **When two FAS are equivalent** then there is some string x over Σ on acceptance of that string if one FA reaches to final state other FA also reaches to final state.

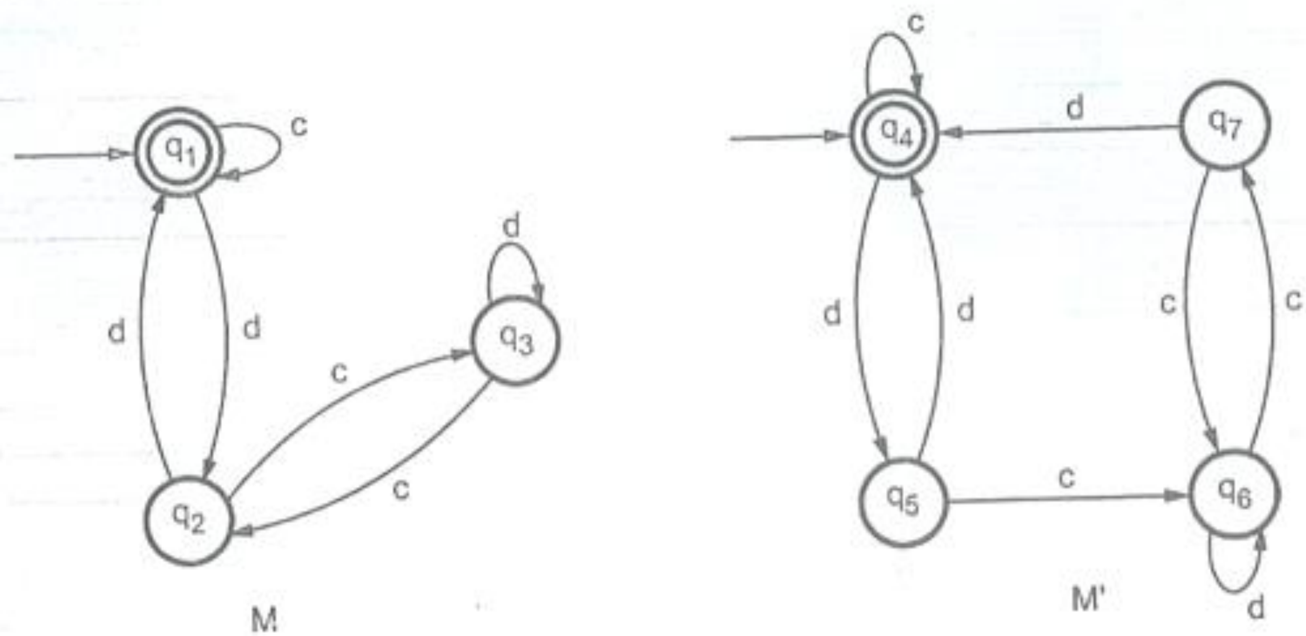
Method for Comparing two FAS.

Let M and M' be two FAS and Σ is a set of input strings.

1. We will construct a transition table have pair wise entries (q, q') where ε M and q' ϵ M' for each symbol.
2. If we get in a pair as one final state and other non-final state then we terminate construction of transition table declaring that two FAS are not equivalent.
3. The construction of transition table gats terminated when there is no new pair appearing in the transition table.

Let us take one example to understand the technique of comparing two FA.

Example 2.20: Consider the DFAs given below are they equivalent



Solution:

We will first build the transition table for each input c and d from first machine M

on receiving input c in state q_1 we reach to state q_1 only from second machine M',

for state q_4 on receiving c we reach to state q_4.

Thus for state (q_1, q_4) for input c we get next state as (q_1, q_4).

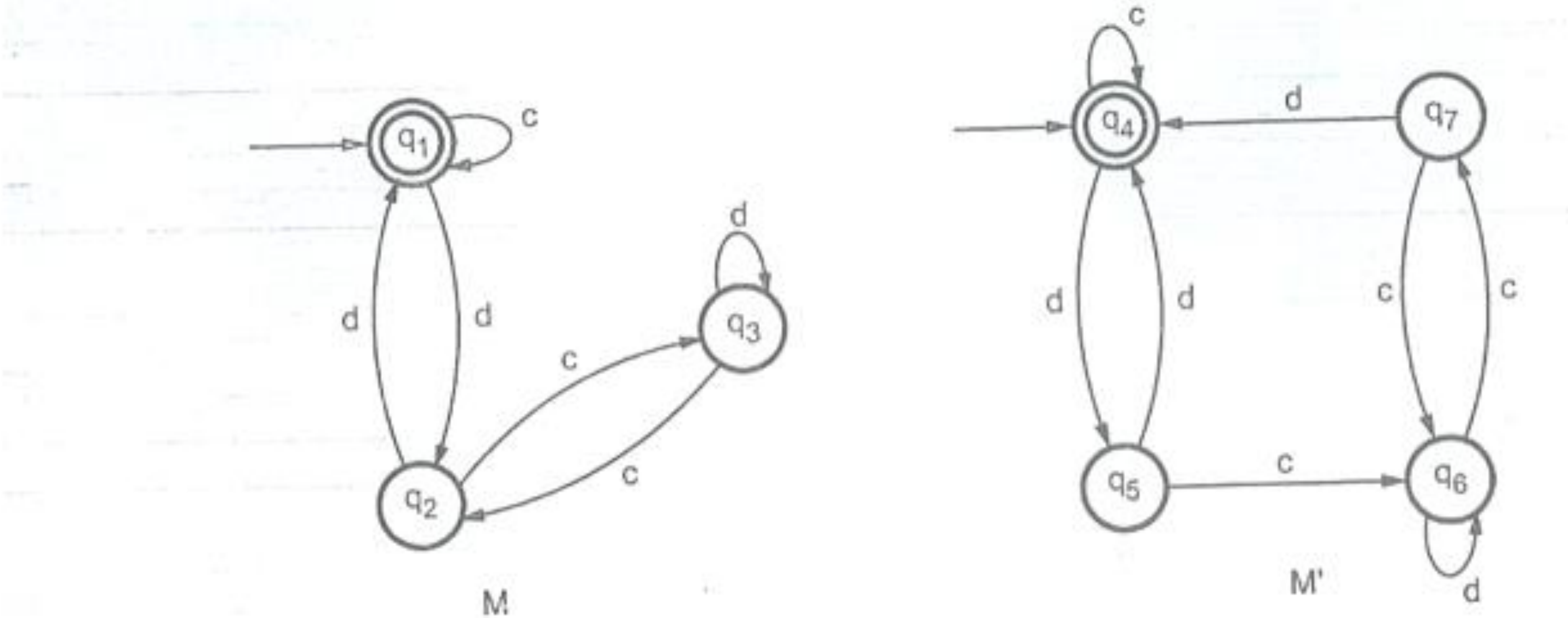
Similarly, for input d in state (q_1, q_4) we get next state as (q_2, q_5).

Both q1 and q4 are final state obtained in pair (q1, q4).

Both q2 and q5 are non-final states obtained in pair (q2, q5), we will obtain

transition for (q2, q5) for input c and d the complete table is as given below.

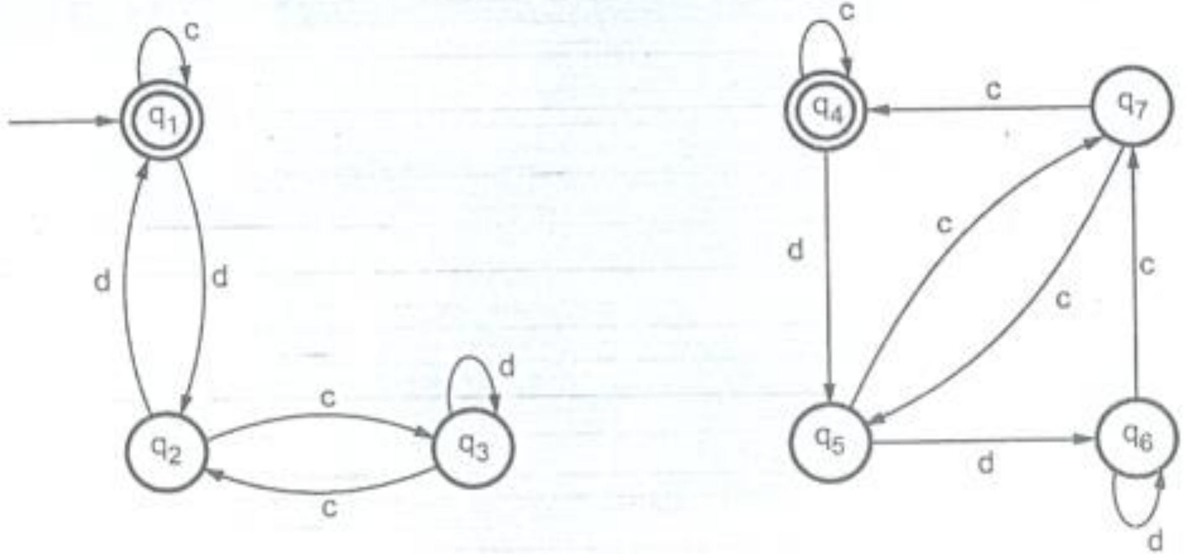
Example 2.20: Consider the DFAs given below are they equivalent

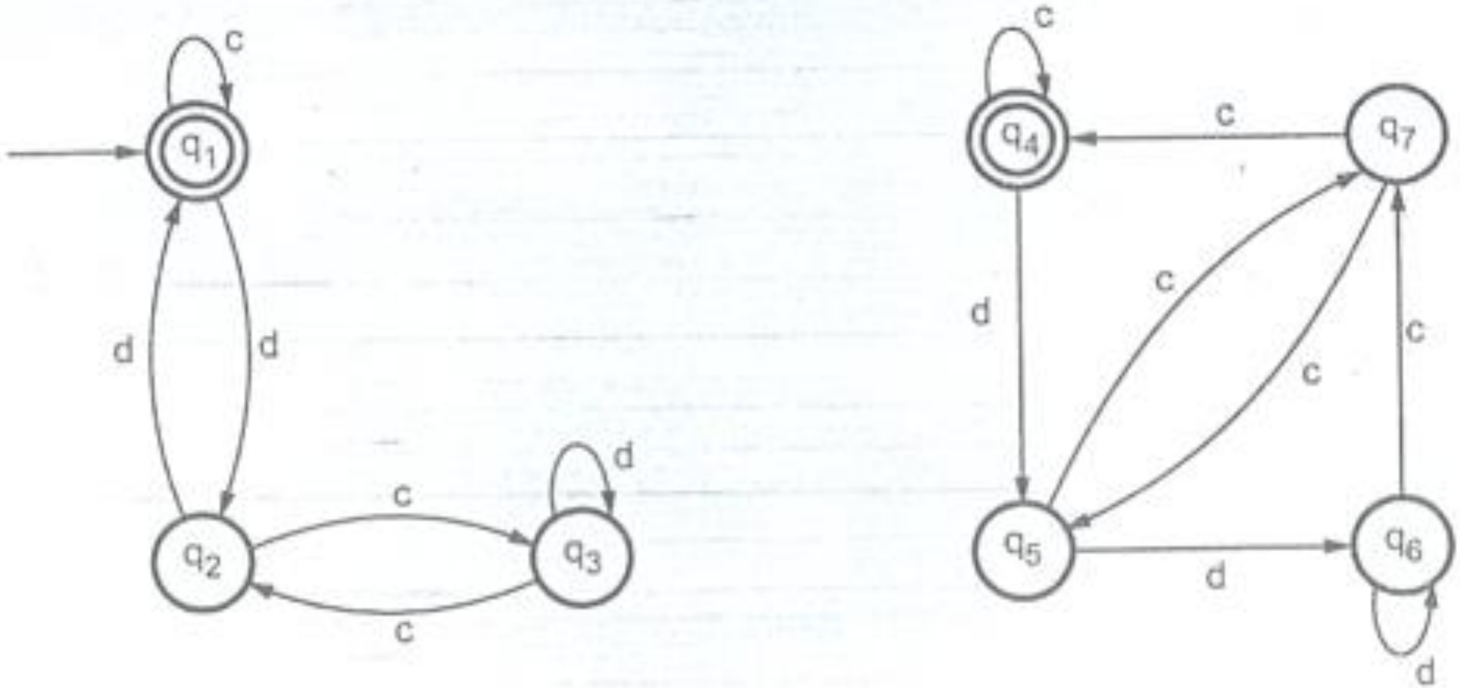


	c	d
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_6)	(q_1, q_4)
(q_3, q_6)	(q_2, q_7)	(q_3, q_6)
(q_2, q_7)	(q_3, q_6)	(q_1, q_4)

From the above table note that we do not get one final state and other non-final state in a pair. Hence we declare that two DFAs are equivalent.

Example 2.21: Following are two FAs check whether they are equivalent or not.





Solution: We will design the transition table for each input symbol c and as follows.

	c	d
(q_1, q_4)	(q_1, q_4)	(q_2, q_5)
(q_2, q_5)	(q_3, q_7)	(q_1, q_6)

We will terminate the construction of transition table because we get a pair (q_1, q_6) in which q_1 is a final and q_6 is a non-final state. As per equivalence rule final and non-final state cannot form a pair. Hence the given FAs are not equivalent.

Any Questions?