

Author: **Vivek Kulkarni**
(vivek_kulkarni@yahoo.com)

Chapter-2: Finite State Machines

Solutions for Review Questions

Q.1 Construct Mealy and Moore machines for the following:

For the input from Σ^* , where $\Sigma = \{0, 1, 2\}$ print the residue-modulo-5 of the input treated as a ternary (base 3 with digits 0, 1, and 2) number.

Solution:

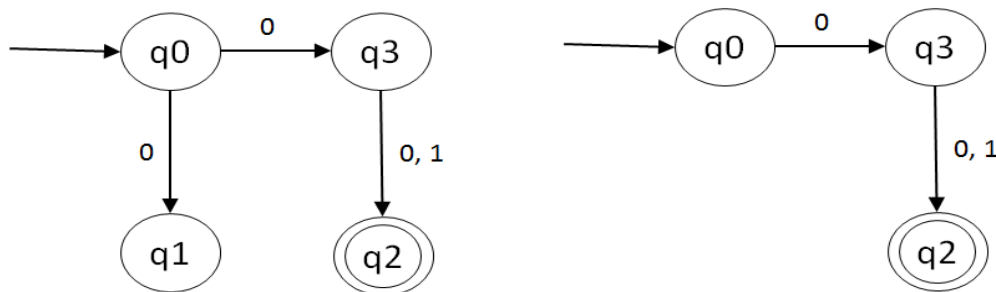
Refer to the example 2.21 from the book.

Q.2 Discuss the relative powers of NFA and DFA.

Solution:

NFA (Non-deterministic Finite Automata) and DFA (Deterministic Finite Automata) are equivalent to each other. Each NFA can be converted to its equivalent DFA, which means, NFA and DFA are of the same power in terms of acceptance of language and state transitions depending upon given input over language alphabets.

We can show equivalence relation in NFA and DFA by drawing NFA and its equivalent DFA as follows.



Above NFA shows two transitions for an input symbol 0 from state q_0 . This NFA only accepts strings “00” and “01”, which means language accepted by the NFA can be given as,

$$L1 = \{00, 01\}$$

Hence state q_1 can be removed from the equivalent DFA as it is not reaching final state q_2 on any symbol. Equivalent DFA accepts same language,

$$L2 = \{00, 01\}$$

As $L1 = L2$, NFA and DFA drawn above are equivalent and accept the same language. Therefore their power is same.

Q.3 Write the machine function and the state transition function for a binary adder. Support your answer with a transition diagram.

Solution:

Refer to the example 2.1 from the book.

Q.4 Prove the following:

“Corresponding to every transition graph, there need not exist an FSM, but the converse is always true”.

Solution:

Refer to the section 2.3.6, theorem 2.1.

Q.5 Define and give suitable examples for a transition graph.

Solution:

Transition Graph (or Transition Diagram) is a directed graph whose vertices corresponds to the states of the Finite State Machine and directed edges corresponds to the transitions from one state to another state on the acceptance of input symbol which is written above the directed edge. Refer section 2.3.1 for details. Refer to the solution of question Q.3 above for the example of transition graph.

Q.6 Construct a Mealy machine that accepts the strings from $(0 + 1)^*$ and produces the output as indicated below:

End of string	Output
101	x
110	y
Otherwise	z

Solution:

Refer to the example 2.20 from the book.

Q.7 Explain whether a language of palindromes is accepted by an FSM. Justify.

Solution:

Refer to the section 2.14.

Q.8 Describe the following:

- (a) State equivalence
- (b) FSM equivalence

Solution:

- (a) State equivalence – Refer to the section 2.6.2.1.
- (b) FSM equivalence – Refer to the section 2.12.

Q.9 Write a short note on Mealy and Moore machines.

Solution:

Refer to the section 2.10.

Q.10 Explain with an example, the process of converting a Mealy machine to its corresponding Moore machine.

Solution:

Refer to the section 2.11.2.

Q.11 Write a short note on the properties and limitations of FSM.

Solution:

Refer to the section 2.14.

Q.12 Compare Moore and Mealy machines.

Solution:

Refer to the section 2.10.

Q.13 Design an FSM to check divisibility by three, where $\Sigma = \{0, 1, \dots, 9\}$.

Solution:

Refer to the example 2.2 from the book.

Q.14 What are finite automata? Construct the minimum state automata equivalent to following state transition diagram:

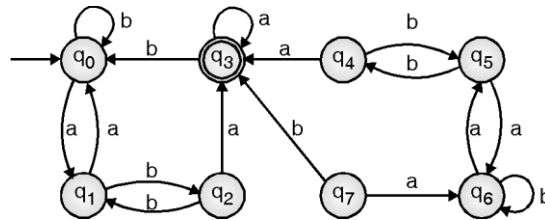
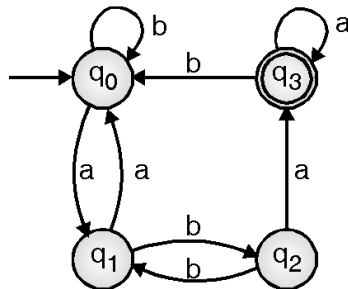


Figure 2.51: Example automata

Solution:

We can see from the figure that the states q_4, q_5, q_6 and q_7 are not reachable from the initial state, as there is no path from the initial state that reaches to these states. We can easily remove these 4 states and their transitions as they do not take part in any string acceptance.

Hence, we are left with the reduced FA as shown in the diagram below,



Let us now draw the STF form the above TG. The STF is,

$Q \setminus \Sigma$	a	b
q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
$*q_3$	q_3	q_0

As we can see, there are no more states that we can reduce and hence, it is the answer.

Q.15 Construct NFA without ϵ -transitions for the following NFA with ϵ -transitions.

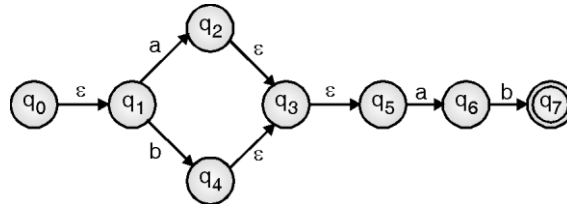


Figure 2.52: Example NFA with ϵ -transitions

Solution:

Let us first find the ϵ -closure of all the states. Let us assume that q_0 is the initial state of the NFA given.

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

$$\epsilon\text{-closure}(q_1) = \{q_1\}$$

$$\epsilon\text{-closure}(q_2) = \{q_2, q_3, q_5\}$$

$$\epsilon\text{-closure}(q_4) = \{q_4, q_3, q_5\}$$

$$\epsilon\text{-closure}(q_3) = \{q_3, q_5\}$$

$$\epsilon\text{-closure}(q_5) = \{q_5\}$$

$$\epsilon\text{-closure}(q_6) = \{q_6\}$$

$$\epsilon\text{-closure}(q_7) = \{q_7\}$$

q_7 is the only final state even for the NFA without ϵ -transitions.

Now let us find transition function δ' for the resultant NFA. This can be obtained by using rule:

$$\delta'(q, a) = \epsilon\text{-closure}(\delta(\delta^{\wedge}(q, \epsilon), a))$$

Therefore,

$$\begin{aligned} \delta'(q_0, a) &= \epsilon\text{-closure}(\delta(\delta^{\wedge}(q_0, \epsilon), a)) \\ &= \epsilon\text{-closure}(\delta(\{q_0, q_1\}, a)) \\ &= \epsilon\text{-closure}(\delta(q_0, a) \cup \delta(q_1, a)) \\ &= \epsilon\text{-closure}(\phi \cup \{q_2\}) \\ &= \epsilon\text{-closure}(q_2) \\ &= \{q_2, q_3, q_5\} \end{aligned}$$

We can obtain all the other transitions. The transition table for the resultant NFA is,

$Q \backslash \Sigma$	a	b
q_0	$\{q_2, q_3, q_5\}$	$\{q_4, q_3, q_5\}$
q_1	$\{q_2, q_3, q_5\}$	$\{q_4, q_3, q_5\}$
q_2	$\{q_6\}$	ϕ
q_3	$\{q_6\}$	ϕ
q_4	$\{q_6\}$	ϕ
q_5	$\{q_6\}$	ϕ
q_6	ϕ	$\{q_7\}$
* q_7	ϕ	ϕ

Q.16 Design an FSM that reads strings made up of letters in the word 'CHARIOT' and recognizes those strings that contain the word 'CAT' as a substring.

Solution:

Refer to the example 2.5 from the book.

Q.17 Construct a Moore machine equivalent to the Mealy machine represented by the following TG:

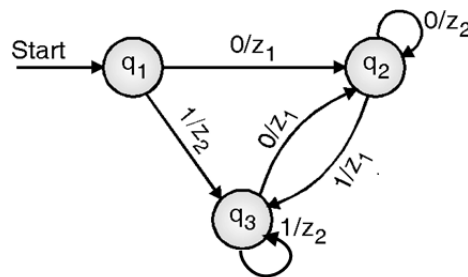


Figure 2.53: Example Mealy machine

Solution:

Let us draw state function and machine function for the given Mealy machine.

$Q \setminus \Sigma$	0	1
q_1	q_2	q_3
q_2	q_2	q_3
q_3	q_2	q_3

$Q \setminus \Sigma$	0	1
q_1	z_1	z_2
q_2	z_2	z_1
q_3	z_1	z_2

$\delta : Q \times \Sigma \rightarrow Q$ (State function)

$\lambda : Q \times \Sigma \rightarrow \Delta$ (Machine function)

According to the rule of conversion, the resultant Moore machine consists of states:

$$Q' = Q \times \Delta = \{[q_1, z_1], [q_1, z_2], [q_2, z_1], [q_2, z_2], [q_3, z_1], [q_3, z_2]\}$$

Also δ' and λ' can be found as follows:

$$\begin{aligned} \text{a) } \delta'([q_1, z_1], 0) &= [\delta(q_1, 0), \lambda(q_1, 0)] \\ &= [q_2, z_1] \end{aligned}$$

$$\begin{aligned} \delta'([q_1, z_1], 1) &= [\delta(q_1, 1), \lambda(q_1, 1)] \\ &= [q_3, z_2] \end{aligned}$$

$$\lambda'([q_1, z_1]) = z_1$$

$$\begin{aligned} \text{b) } \delta'([q_1, z_2], 0) &= [\delta(q_1, 0), \lambda(q_1, 0)] \\ &= [q_2, z_1] \end{aligned}$$

$$\begin{aligned} \delta'([q_1, z_2], 1) &= [\delta(q_1, 1), \lambda(q_1, 1)] \\ &= [q_3, z_2] \end{aligned}$$

$$\lambda'([q_1, z_2]) = z_2$$

$$\begin{aligned} \text{c) } \delta'([q_2, z_1], 0) &= [\delta(q_2, 0), \lambda(q_2, 0)] \\ &= [q_2, z_2] \end{aligned}$$

$$\begin{aligned} \delta'([q_2, z_1], 1) &= [\delta(q_2, 1), \lambda(q_2, 1)] \\ &= [q_3, z_1] \end{aligned}$$

$$\lambda'([q_2, z_1]) = z_1$$

$$\begin{aligned} \text{d) } \delta'([q_2, z_2], 0) &= [\delta(q_2, 0), \lambda(q_2, 0)] \\ &= [q_2, z_2] \end{aligned}$$

$$\begin{aligned} \delta'([q_2, z_2], 1) &= [\delta(q_2, 1), \lambda(q_2, 1)] \\ &= [q_3, z_1] \end{aligned}$$

$$\lambda'([q_2, z_2]) = z_2$$

$$\begin{aligned}
 \text{e) } \delta'([q_3, z_1], 0) &= [\delta(q_3, 0), \lambda(q_3, 0)] \\
 &= [q_2, z_1] \\
 \delta'([q_3, z_1], 1) &= [\delta(q_3, 1), \lambda(q_3, 1)] \\
 &= [q_3, z_2] \\
 \lambda'([q_3, z_1]) &= z_1 \\
 \text{f) } \delta'([q_3, z_2], 0) &= [\delta(q_3, 0), \lambda(q_3, 0)] \\
 &= [q_2, z_1] \\
 \delta'([q_3, z_2], 1) &= [\delta(q_3, 1), \lambda(q_3, 1)] \\
 &= [q_3, z_2] \\
 \lambda'([q_3, z_2]) &= z_2
 \end{aligned}$$

Q.18 Consider the DFA as shown in the Fig. 2.54. Obtain the minimum state DFA.

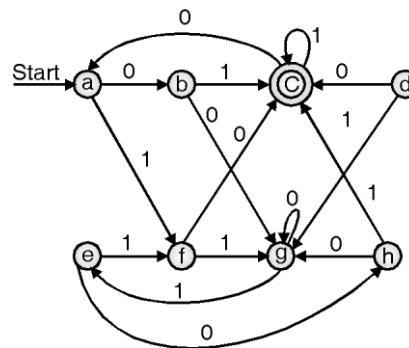


Figure 2.54: Example DFA

Solution:

Refer to the section 2.13.

Q.19 Consider the Moore machine described by the transition table given below. Construct the corresponding Mealy Machine.

Current state	Next state		Output
	a = 0	a = 1	
→ q ₁	q ₁	q ₂	0
q ₂	q ₁	q ₃	0
q ₃	q ₁	q ₃	1

Solution:

Refer to the example 2.23 from the book.

Q.20 Construct a DFA equivalent to the NFA: $(\{p, q, r, s\}, \{0, 1\}, \delta, p, \{q, s\})$, where ‘ δ ’ is given by:

Σ	0	1
Q		
p	q,r	q
q	r	q,r
r	s	p
s	-	p

Solution:

Refer to the example 2.10 from the book.

Q.21 Write and explain all the steps required for the conversion of an NFA to a DFA using a suitable example.

Solution:

Refer to the section 2.6.

Q.22 Convert the following Mealy machine to a Moore machine.

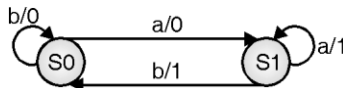


Figure 2.55: Example Mealy machine

Solution:

Let us draw state function and machine function for the given Mealy machine.

Q \ Σ	a	b
S0	S1	S0
S1	S1	S0

Q \ Σ	a	b
S0	0	0
S1	1	1

$\delta : Q \times \Sigma \rightarrow Q$ (State function)

$\lambda : Q \times \Sigma \rightarrow \Delta$ (Machine function)

According to the rule of conversion, the resultant Moore machine consists of states:

$Q' = Q \times \Delta = \{[S0, 0], [S0, 1], [S1, 0], [S1, 1]\}$

Also δ' and λ' can be found as follows:

$$\begin{aligned}
 \text{a) } \delta'([S0, 0], a) &= [\delta(S0, a), \lambda(S0, a)] \\
 &= [S1, 0] \\
 \delta'([S0, 0], b) &= [\delta(S0, b), \lambda(S0, b)] \\
 &= [S0, 0] \\
 \lambda'([S0, 0]) &= 0 \\
 \text{b) } \delta'([S0, 1], a) &= [\delta(S0, a), \lambda(S0, a)] \\
 &= [S1, 0] \\
 \delta'([S0, 1], b) &= [\delta(S0, b), \lambda(S0, b)] \\
 &= [S0, 0] \\
 \lambda'([S0, 1]) &= 1 \\
 \text{c) } \delta'([S1, 0], a) &= [\delta(S1, a), \lambda(S1, a)] \\
 &= [S1, 1] \\
 \delta'([S1, 0], b) &= [\delta(S1, b), \lambda(S1, b)] \\
 &= [S0, 1] \\
 \lambda'([S1, 0]) &= 0 \\
 \text{d) } \delta'([S1, 1], a) &= [\delta(S1, a), \lambda(S1, a)] \\
 &= [S1, 1] \\
 \delta'([S1, 1], b) &= [\delta(S1, b), \lambda(S1, b)] \\
 &= [S0, 1] \\
 \lambda'([S1, 1]) &= 1
 \end{aligned}$$

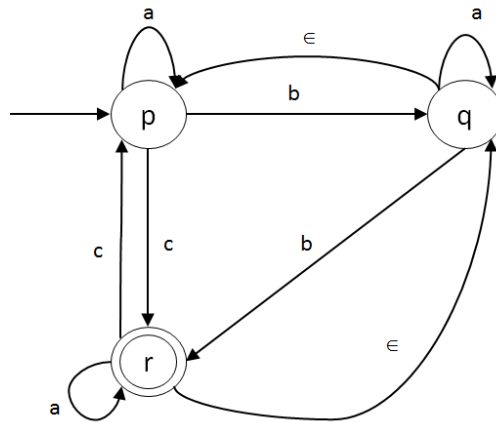
Q.23 Consider the following NFA with ϵ -transitions. Assume 'p' to be the initial state and 'r' as the final state.

	ϵ	a	b	c
p	ϕ	{p}	{q}	{r}
q	{p}	{q}	{r}	ϕ
r	{q}	{r}	ϕ	{p}

- 1) Compute the ϵ -closure of each state
- 2) List all the strings of length three or less accepted by the automata
- 3) Convert the automaton to its equivalent DFA

Solution:

Let us first draw the TG for the given NFA with ϵ -transitions.

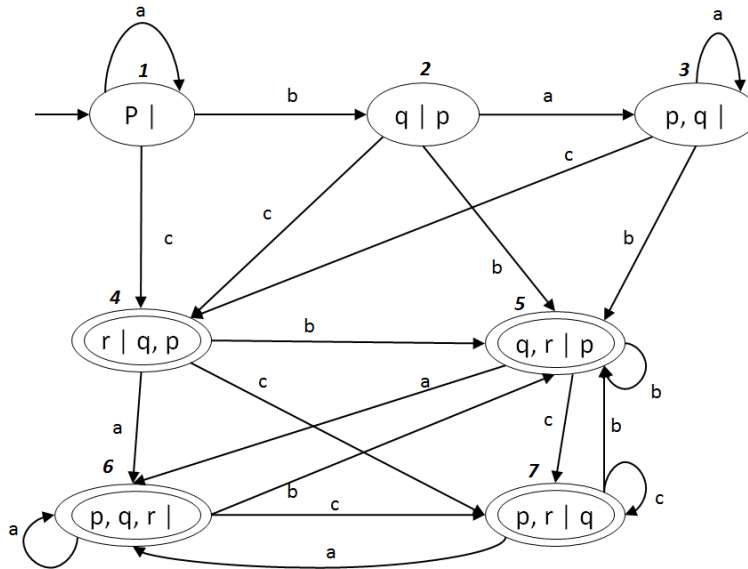


- 1) ϵ -closure (p) = {p}
 ϵ -closure (q) = {p, q}
 ϵ -closure (r) = {p, q, r}

- 2) All the strings of length three or less accepted by the automata
 = {c, ac, bb, bc, ca, cb, cc, abb, abc, bab, bac, aca, acb, acc, caa, cbb, ccc, cab, cba, cac, cca, cbc, ccb}

- 3) The equivalent DFA can be obtained using a direct method as below. Each state has the primary as well as the secondary label. Secondary label consists of all the reachable states from the given state. We also have relabelled the states from 1 to 7, out of which 4, 5, 6 and 7 are final states.

Let us also refer to the STF drawn and see whether it can be reduced.

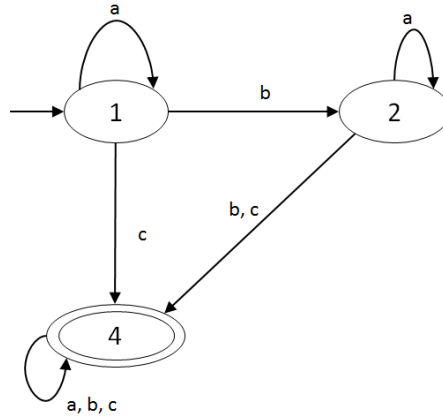


	a	b	c
1	1	2	4
2	3	5	4
3	3	5	4
* 4	6	5	7
* 5	6	5	7
* 6	6	5	7
* 7	6	5	7

One can see that states 4, 5, 6 and 7 are equivalent. Hence, we can keep only 4 and remove the others by replacing all their occurrence by 4. Similarly, 2 and 3 are also equivalent; hence, 3 can be removed in lieu of state 2. Therefore, the reduced STF is,

	a	b	c
1	1	2	4
2	2	3	4
* 4	4	4	4

The reduced state DFA is,



Q.24 Construct a DFA for the NFA, whose state transition function is given below. Assume 'p' to be the initial state and $F = \{q, r\}$.

	0	1
p	{p, q}	ϕ
q	r	s
f	s	ϕ
s	s	s

Solution:

Initial state for the resultant DFA is also p and set of states $Q' = 2^Q$.

Hence, $Q' = \{p, q, r, s, pq, pr, ps, qr, qs, rs, pqr, pqs, prs, qrs, pqrs\}$

For the given NFA, set of Final states is given as $\{q, r\}$.

Hence, set of final states for the resultant DFA is, $F' = \{q, r, pq, pr, qr, qs, rs, pqr, pqs, prs, qrs, pqrs\}$

When two or more symbols represent one state, transition for the resultant DFA can be obtained as follows,

$$\delta(pq, 0) = [\delta(p, 0) \cup \delta(q, 0)] = [\{p, q\} \cup \{r\}] = pqr$$

Thus, the STF for the resultant DFA can be written as below,

$Q \setminus \Sigma$		0	1
(1)	p	pq	ϕ
(2)	*q	r	s
(3)	*r	s	ϕ
(4)	s	s	s
(5)	*pq	pqr	s
(6)	*pr	pqs	ϕ
(7)	ps	pqs	s
(8)	*qr	rs	s
(9)	*qs	rs	s
(10)	*rs	s	s
(11)	*pqr	pqrs	s
(12)	*pqs	pqrs	s
(13)	*prs	pqs	s
(14)	*qrs	rs	s
(15)	*pqrs	pqrs	s

We also have relabelled the states from (1) to (15).

We can see that states, (11), (12) and (15) are equivalent. Also, (8), (9) and (14) are equivalent. If we remove states (12) and (15) and replace all occurrences of them by (11) we can get the reduced state DFA. Similarly, we can remove (9) and (14) and keep (8) instead. The reduced table would look like,

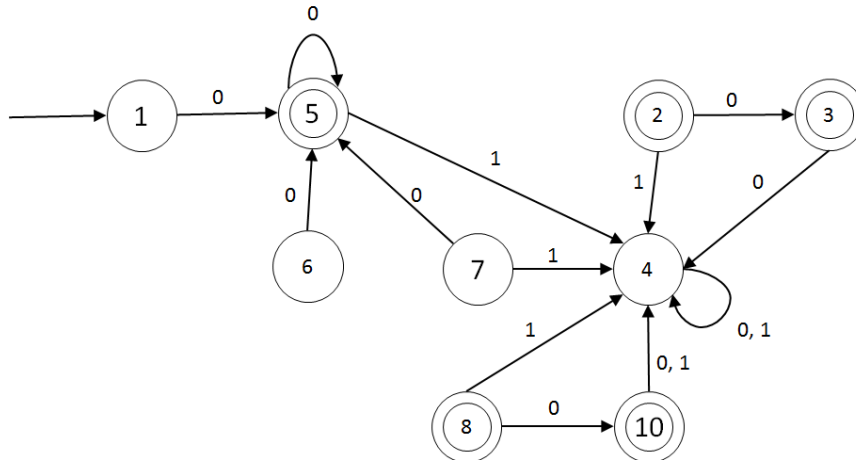
$Q \setminus \Sigma$	0	1
(1) p	5	-
(2) *q	3	4
(3) *r	4	-
(4) s	4	4
(5) *pq	11	4
(6) *pr	11	-
(7) ps	11	4
(8) *qr	10	4
(10) *rs	4	4
(11) *pqr	11	4
(13) *prs	11	4

From the reduced state table we can see that states (5), (11) and (13) are equivalent. Hence, (11) and (13) can be removed and their occurrences can be replaced by (5) to obtain the reduced state DFA.

$Q \setminus \Sigma$	0	1
1	5	-
* 2	3	4
* 3	4	-
4	4	4
* 5	5	4
* 6	5	-

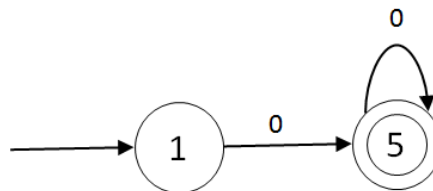
7	5	4
* 8	10	4
* 10	4	4

The TG for the DFA can be drawn as,



One can see that states 2, 3, 6, 7, 8 and 10 are no reachable from the initial state. Hence, they do not take part in the language recognition and hence can be removed being redundant. Also, state 4 though is reachable from the initial state does not involve in the acceptance and it does not lead to any final state. It is a trap state. Hence, needs to be removed as well.

Removing all the unwanted states the final TG for the resultant DFA is,



Q.25 Explain Moore and Mealy machines using suitable examples. How do we construct the equivalent Mealy machine for a given Moore machine? Give a suitable example.

Solution:

Refer to the section 2.11.1.

Q.26 Compare Mealy and Moore machines. Design a Mealy machine to replace each occurrence of sub-string 'abb' by 'aba', where $\Sigma = \{a, b\}$.

Solution:

Refer to the section 2.11 for Mealy and Moore machine comparison.

We need to design a Mealy machine which will replace substring 'abb' by 'aba'.

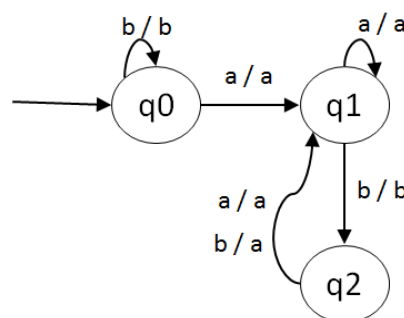
- Let us assume that initial state of the machine is q_0 . As none of the substrings start with symbol 'b', all such 'b's will be read and ignored by the machine in state q_0 .
- Transition to state q_1 is made after reading the input symbol 'a' where subsequent input will be checked. As we have to replace substring, only matching pattern should be replaced and other patterns should be kept intact as they are.
- State q_1 will check for the next input after previous input symbol 'a' that is read and will transit to state q_2 if 'b' is the input symbol read. Otherwise no transition is made.
- In state q_2 it checks if input is 'b' (thus satisfying the required pattern 'abb') and transits to state q_1 with replacing 'b' by 'a' and continue the process for each such pattern.

State function and the machine function for the required mealy machine are shown below.

$Q \setminus \Sigma$	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_1

$Q \setminus \Sigma$	a	b
q_0	a	b
q_1	a	b
q_2	a	a

The transition diagram for the mealy machine is as shown below,



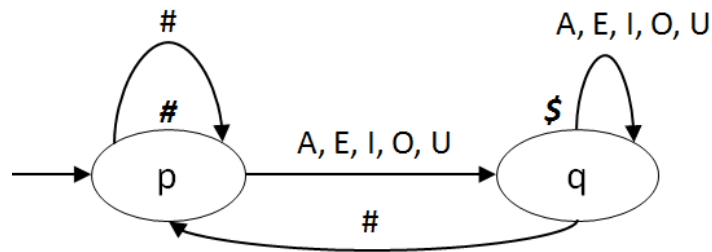
Q.27 Design a Moore machine that changes all the vowels to ‘\$’.

Solution:

This is a very simple sequence detector problem. Let $\Sigma = \{A, B, C, \dots, Z\}$.

It is required that the vowels from any word, i.e., $\{A, E, I, O, U\}$ needs to be replaced by ‘\$’.

Let us label all the remaining symbols except the vowels as ‘#’ for the simplicity. Using this convention Moore machine can be drawn as,



As we can see from the TG that state p is associated with the output ‘#’ which indicates no change in the symbol if it is any other character than the vowels, while; the state q is associated with the output ‘\$’ that changes the vowels to ‘\$’.

Q.28 Construct a DFA equivalent to the NFA: $(\{p, q, r, s\}, \{0, 1\}, \delta_N, p, \{q, s\})$, where δ_N is as given in the following table:

Q \ Σ	0	1
$\rightarrow p$	{q, r}	{q}
*q	{r}	{q, r}
r	{s}	{p}
*s	-	{p}

Solution:

Refer to the example 2.10 from the book.

Q.29 Write short notes on:

- (1) Deterministic finite automata
- (2) Moore and Mealy machines
- (3) Moore's algorithm for FSM equivalence
- (4) Relative powers of NFA and DFA
- (5) Limitations of FSM

Solution:

- (1) Deterministic finite automata – Refer to the section 2.4.
- (2) Moore and Mealy machines – Refer to the section 2.10.
- (3) Moore's algorithm for FSM equivalence – Refer to the section 2.12.1.
- (4) Relative powers of NFA and DFA – Refer to the section 2.6.
- (5) Limitations of FSM – Refer to the section 2.14.

Q.30 Give formal definitions for the following:

- (1) Deterministic finite automata
- (2) NFA with ϵ -transitions
- (3) Moore machine
- (4) Acceptance of a string by FA

Solution:

- (1) Deterministic finite automata – Refer to the section 2.4.
- (2) NFA with ϵ -transitions – Refer to the section 2.7.
- (3) Moore machine – Refer to the section 2.10.1.
- (4) Acceptance of a string by FA – Refer to the section 2.3.3.

Q.31 Translate the following Mealy machine into its equivalent Moore machine.

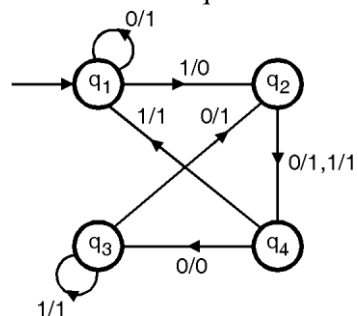


Figure 2.56: Example Mealy machine

Solution:

Let us draw state function and machine function for the given Mealy machine.

Q \ Σ	0	1
q ₁	q ₁	q ₂
q ₂	q ₄	q ₄
q ₃	q ₂	q ₃
q ₄	q ₃	q ₁

$\delta : Q \times \Sigma \rightarrow Q$ (State function)

Q \ Σ	0	1
q ₁	1	0
q ₂	1	1
q ₃	1	1
q ₄	0	1

$\lambda : Q \times \Sigma \rightarrow \Delta$ (Machine function)

According to the rule of conversion, the resultant Moore machine consists of states,

$$Q' = Q \times \Delta = \{ [q_1, 0], [q_1, 1], [q_2, 0], [q_2, 1], [q_3, 0], [q_3, 1], [q_4, 0], [q_4, 1] \}$$

Also δ' and λ' can be found as follows:

$$\text{a) } \delta'([q_1, 0], 0) = [\delta(q_1, 0), \lambda(q_1, 0)]$$

$$= [q_1, 1]$$

$$\delta'([q_1, 0], 1) = [\delta(q_1, 1), \lambda(q_1, 1)]$$

$$= [q_2, 0]$$

$$\lambda'([q_1, 0]) = 0$$

$$\text{b) } \delta'([q_1, 1], 0) = [\delta(q_1, 1), \lambda(q_1, 1)]$$

$$= [q_2, 0]$$

$$\delta'([q_1, 1], 1) = [\delta(q_1, 1), \lambda(q_1, 1)]$$

$$= [q_2, 0]$$

$$\lambda'([q_1, 1]) = 1$$

$$\text{c) } \delta'([q_2, 0], 0) = [\delta(q_2, 0), \lambda(q_2, 0)]$$

$$= [q_4, 1]$$

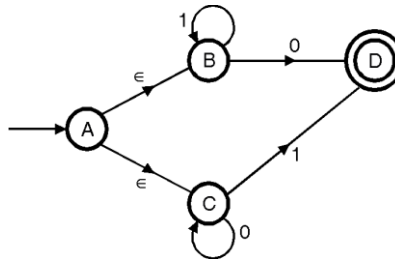
$$\delta'([q_2, 0], 1) = [\delta(q_2, 1), \lambda(q_2, 1)]$$

$$= [q_4, 1]$$

$$\lambda'([q_2, 0]) = 0$$

$$\begin{aligned}
\text{d) } \delta'([q_2, 1], 0) &= [\delta(q_2, 0), \lambda(q_2, 0)] \\
&= [q_4, 1] \\
\delta'([q_2, 1], 1) &= [\delta(q_2, 1), \lambda(q_2, 1)] \\
&= [q_4, 1] \\
\lambda'([q_2, 1]) &= 1 \\
\text{e) } \delta'([q_3, 0], 0) &= [\delta(q_3, 0), \lambda(q_3, 0)] \\
&= [q_2, 1] \\
\delta'([q_3, 0], 1) &= [\delta(q_3, 1), \lambda(q_3, 1)] \\
&= [q_3, 1] \\
\lambda'([q_3, 0]) &= 0 \\
\text{f) } \delta'([q_3, 1], 0) &= [\delta(q_3, 0), \lambda(q_3, 0)] \\
&= [q_3, 1] \\
\delta'([q_3, 1], 1) &= [\delta(q_3, 1), \lambda(q_3, 1)] \\
&= [q_3, 1] \\
\lambda'([q_3, 1]) &= 1 \\
\text{g) } \delta'([q_4, 0], 0) &= [\delta(q_4, 0), \lambda(q_4, 0)] \\
&= [q_3, 0] \\
\delta'([q_4, 0], 1) &= [\delta(q_4, 1), \lambda(q_4, 1)] \\
&= [q_4, 1] \\
\lambda'([q_4, 0]) &= 0 \\
\text{h) } \delta'([q_4, 1], 0) &= [\delta(q_4, 0), \lambda(q_4, 0)] \\
&= [q_3, 0] \\
\delta'([q_4, 1], 1) &= [\delta(q_4, 1), \lambda(q_4, 1)] \\
&= [q_4, 1] \\
\lambda'([q_4, 1]) &= 1
\end{aligned}$$

Q.32 Convert the following NFA into NFA without ϵ -moves.

Figure 2.57: Example NFA with ϵ -moves**Solution:**

Let's calculate ϵ -closure of the states.

$$\epsilon\text{-Closure}(A) = \{B, C\}$$

$$\epsilon\text{-Closure}(B) = \{B\}$$

$$\epsilon\text{-Closure}(C) = \{C\}$$

$$\epsilon\text{-Closure}(D) = \{D\}$$

As D is the only final state for given NFA without ϵ -transitions, F' : set of final states for resultant NFA without ϵ -moves is given as,

$$F' = \{D\}$$

Now let us find transition function δ' for the resultant NFA. This can be obtained by using rule:

$$\delta'(q, a) = \epsilon\text{-closure}(\delta(\delta^*(q, \epsilon), a))$$

$$\begin{aligned} \delta'(A, 0) &= \epsilon\text{-closure}(\delta(\delta^*(A, \epsilon), 0)) \\ &= \epsilon\text{-closure}(\delta(\{B, C\}, 0)) \\ &= \epsilon\text{-closure}(\delta(B, 0) \cup \delta(C, 0)) \\ &= \epsilon\text{-closure}(\{D\}, \{c\}) \\ &= \{C, D\} \end{aligned}$$

$$\begin{aligned} \delta'(A, 1) &= \epsilon\text{-closure}(\delta(\delta^*(A, \epsilon), 1)) \\ &= \epsilon\text{-closure}(\delta(\{B, C\}, 1)) \\ &= \epsilon\text{-closure}(\delta(B, 1) \cup \delta(C, 1)) \\ &= \epsilon\text{-closure}(\{B\}, \{D\}) \\ &= \{B, D\} \end{aligned}$$

$$\delta'(B, 0) = \epsilon\text{-closure}(\delta(\delta^*(B, \epsilon), 0))$$

$$= \epsilon\text{-closure}(\delta(B, 0))$$

$$= \epsilon\text{-closure}(\{D\})$$

$$= \{D\}$$

$$\delta'(B, 1) = \epsilon\text{-closure}(\delta(\delta^{\wedge}(B, \epsilon), 1))$$

$$= \epsilon\text{-closure}(\delta(B, 1))$$

$$= \epsilon\text{-closure}(\{B\})$$

$$= \{B\}$$

$$\delta'(C, 0) = \epsilon\text{-closure}(\delta(\delta^{\wedge}(C, \epsilon), 0))$$

$$= \epsilon\text{-closure}(\delta(C, 0))$$

$$= \epsilon\text{-closure}(\{C\})$$

$$= \{C\}$$

$$\delta'(C, 1) = \epsilon\text{-closure}(\delta(\delta^{\wedge}(C, \epsilon), 1))$$

$$= \epsilon\text{-closure}(\delta(C, 1))$$

$$= \epsilon\text{-closure}(\{D\})$$

$$= \{D\}$$

$$\delta'(D, 0) = \epsilon\text{-closure}(\delta(\delta^{\wedge}(D, \epsilon), 0))$$

$$= \epsilon\text{-closure}(\delta(D, 0))$$

$$= \epsilon\text{-closure}(\phi)$$

$$= \phi$$

$$\delta'(D, 1) = \epsilon\text{-closure}(\delta(\delta^{\wedge}(D, \epsilon), 1))$$

$$= \epsilon\text{-closure}(\delta(D, 1))$$

$$= \epsilon\text{-closure}(\phi)$$

$$= \phi$$

From the above transitions, we can construct the transition table for the equivalent NFA without ϵ -moves as below,

$Q \setminus \Sigma$	0	1
A	{C,D}	{B,D}
B	{D}	{B}
C	{C}	{D}
D	ϕ	ϕ

The transition graph for the NFA without ϵ -moves is,

