TOC(CD) ASSIGNMENT

M. Mounika
320126551039 CSD

1) DFA (Deterministic Finite Automata):

It is a machine combination of 5 tuples

$m = \{Q, \Sigma, \delta, q_0, F\}$

where, Q = finite non-empty set of states

Σ = input alphabet

δ = Transition function / mapping function

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

q_0 = initial state ; $q_0 \in Q$

F = Final state ; $F \subseteq Q$

→ In DFA, there is only one path for specific input from the current state to next state

→ It can contain multiple final states.

→ DFA does not accept the null move i.e., the DFA cannot change without input character.

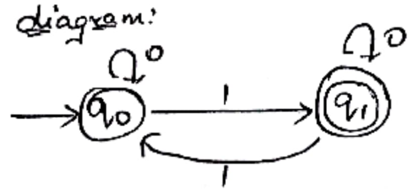
→ It is used in lexical analysis in compiler

Ex: Design a DFA for odd number of 1's over $\Sigma = \{0,1\}$

Condition for odd no. of 1's is

$L = \{1, 111, 11111, \ldots\}$

Transition diagram:



Transition table:

state	Input 0	Input 1
q_0	q_0	q_1
q_1	q_1	q_0

2) Non-Deterministic Finite Data (NFA):

→ NFA is easy to construct compared to DFA.

→ It also have 5 tuples same as DFA.

$$m = \{Q, \Sigma, \delta, q_0, F\}$$

Q = Finite set of states

Σ = Finite set of inputs

δ = Transition function

q_0 = Initial state

F = Final state

→ Every NFA is not a DFA but every DFA is a NFA.

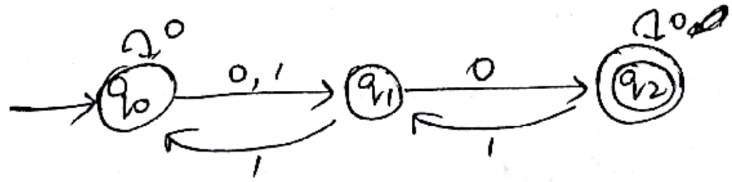
→ The finite automata are NFA when there exist multiple paths for specific input from current state to next state.

→ NFA can use empty string transition

→ Dead state is not required.

→ Backtracking is not always possible in NFA.

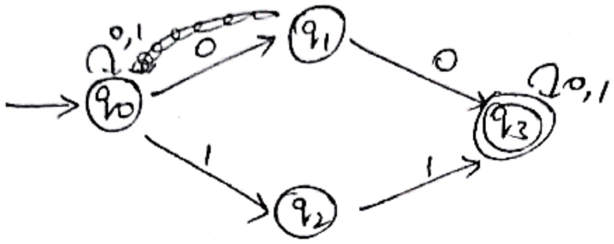
Ex:



3) Conversion from NFA - DFA:

Suppose there is an NFA which recognizes a language L. Then the DFA can be constructed for language L as:

NFA :



Transition table for NFA:

State	Input 0	Input 1
$\rightarrow q_0$	$[q_0, q_1]$	$[q_0, q_2]$
q_1	q_3	ϕ
q_2	ϕ	q_3
q_3	q_3	q_3

DFA Conversion table: $[\delta' = \delta((q_0, q_1), 0)]$

The DFA table cannot have multiple states. So, make $[q_0, q_1]$ as a single state.

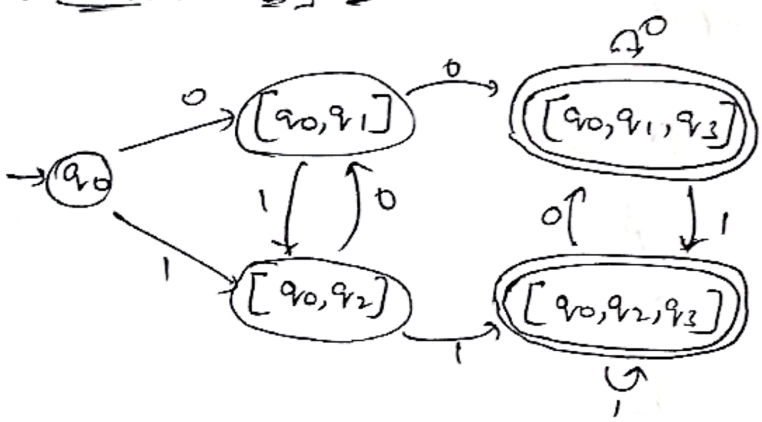
→ Convert the given NFA to DFA by considering two states as a single state.

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

→ The final DFA all states contains the final states of NFA are treated as final states.

State	Input 0	1
$\rightarrow q_0$	$[q_0, q_1]$	$[q_0, q_2]$
$[q_0, q_1]$	$[q_0, q_1, q_3]$	$[q_0, q_2]$
$[q_0, q_2]$	$[q_0, q_1]$	$[q_0, q_2, q_3]$
$[q_0, q_1, q_3]$	$[q_0, q_1, q_3]$	$[q_0, q_2, q_3]$
$[q_0, q_2, q_3]$	$[q_0, q_1, q_3]$	$[q_0, q_2, q_3]$

DFA Transition diagram:



4) Moore Machine:

Finite automata with output

- Moore machine
- Mealy machine

Moore machine is a finite state machine in which the next state is decided by current state and current input symbol. The output symbol at a given time depends only on present state of a machine.

A moore machine can be described by a 6 tuple:

$$m = \{Q, \Sigma, O, \delta, \lambda, q_0\}$$

Q = Finite set of states

Σ = Finite set of symbols called input alphabet

O = Finite set of symbols called output alphabet.

δ = Input transition function

$$\delta : Q \times \Sigma \rightarrow Q$$

λ = Output transition function where

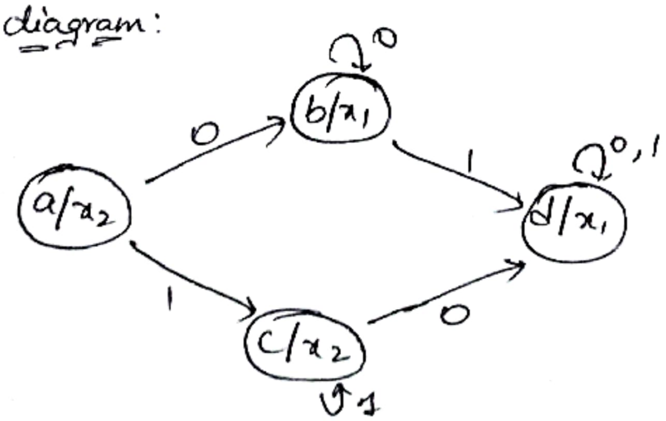
$$\lambda : Q \rightarrow O$$

q_0 = Initial state of machine

Ex:-

State	Next state Input 0	1	Output
a	b	c	x_2
b	b	d	x_1
c	c	d	x_2
d	d	d	x_1

State diagram:



5) Mealy Machine:

A mealy machine is a machine in which output symbol depends upon the present input symbol and present state of machine. In mealy machine, the output is represented with each input symbol for each state separated.

It is a combination of 6 tuple

$$M = \{Q, \Sigma, O, \delta, \lambda, q_0\}$$

Q = Finite set of states

Σ = Finite set of input alphabets

O = Finite set of output alphabets

δ = input ~~sb~~ transition function

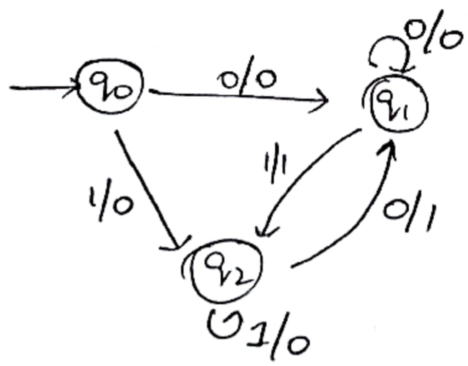
$$\delta : Q \times \Sigma \to Q$$

λ = Output transition function

$$\lambda : Q \times \Sigma \to O$$

q_0 = Initial state

Ex:



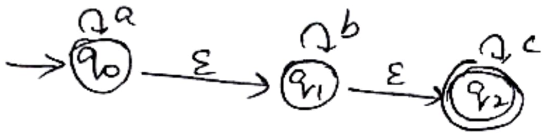
6) NFA-ε to NFA:

Epsilon NFA is the NFA which contains epsilon moves / null moves. To remove the epsilon / null moves we convert into NFA.

Conversion:

Rule 1: ε-closure = { states of NFA-ε }

Rule 2: $\delta'(q,a) = \varepsilon\text{-closure}\,(\delta(\hat{\delta}(q,\varepsilon),a))$



Transition table of NFA-ε:

State	Input a	b	c	ε
$\rightarrow q_0$	q_0	ϕ	ϕ	q_1
q_1	ϕ	q_1	ϕ	q_2
q_2	ϕ	ϕ	q_2	ϕ

Step 1: $\varepsilon\text{-closure}\,() = \{q_0, q_1, q_2\}$

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

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

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

Step 2: $\delta'(q,a) = \varepsilon\text{-closure}\,(\delta(\hat{\delta}(q,\varepsilon), a))$

$= \varepsilon\text{-closure}\,(\delta(\varepsilon\text{-closure}\,(q_0), a))$

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

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

$= \varepsilon\text{-closure}\,(q_0 \cup \phi \cup \phi)$

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

$\delta'(q_0,a) = \{q_0, q_1, q_2\}$

$\delta'(q_0,b) = \varepsilon\text{-closure}\,(q_1)$

$= \{q_1, q_2\}$

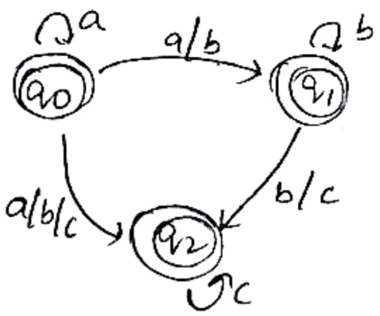
$\delta'(q_0,c) = \varepsilon\text{-closure}\,(q_2)$

$= \{q_2\}$

$\delta'(q_1,a) = \phi$ $\quad$ $\delta'(q_2,a) = \phi$

$\delta'(q_1,b) = \{q_1,q_2\}$ $\quad$ $\delta'(q_2,b) = \phi$

$\delta'(q_1,c) = \{q_2\}$ $\quad$ $\delta'(q_2,c) = q_2$



Transition table of NFA:

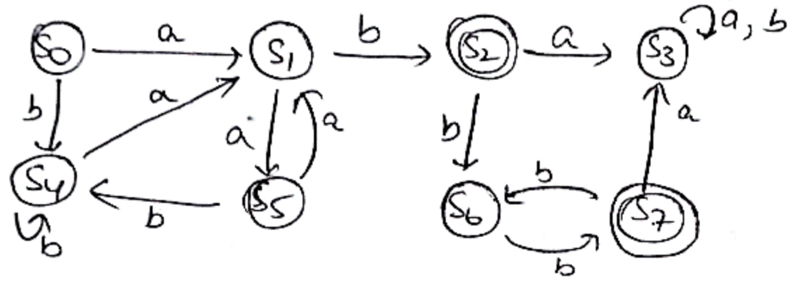
State	a	b	c
$\rightarrow q_0$	$[q_0, q_1, q_2]$	$[q_1, q_2]$	$[q_2]$
q_1	ϕ	$[q_1, q_2]$	$[q_2]$
q_2	ϕ	ϕ	q_2

7) Minimization of FA:

Minimization of DFA means reducing the number of states from given FA. Thus, we get the finite state machine with redundant states after minimizing the FSM.

- → Remove unreachable states
- → Remove dead states.
- → Remove indistinguishable states

Ex:



i) Transition table:

State	Input a	Input b
→ S_0	S_1	S_4
S_1	S_5	S_2
S_2	S_3	S_6
S_3	S_3	S_3
S_4	S_1	S_4
S_5	S_1	S_4
S_6	S_3	S_7
S_7	S_3	S_6

ii) Level (0) $\pi_0 = \{ \text{final}, \text{non-final} \}$

$Q_1^0 = \{S_2, S_7\}$, $Q_2^0 = \{S_0, S_1, S_3, S_4, S_5, S_6\}$

$\pi_0 = \{[S_2, S_7], [S_0, S_1, S_3, S_4, S_5, S_6]\}$

iii) Level 1 (π_1):

from $Q_1^0 = [S_2, S_7]$ No partition , So,

$Q_1' = [S_2, S_7]$

from $Q_2^0 = [S_0, S_1, S_3, S_4, S_5, S_6]$ There is partition

under column a : no change.

col b : $Q_2' = [S_1, S_6]$

$Q_3' = [S_0, S_3, S_4, S_5]$

	a	b
S_0	S_1	S_4
S_1	S_5	S_2
S_3	S_3	S_3
S_4	S_1	S_4
S_5	S_1	S_4
S_6	S_3	S_7

$\therefore \pi_1 = \{ [S_2, S_7], [S_1, S_6], [S_0, S_3, S_4, S_5] \}$

level 2 (π_2):

$Q_2' = [S_2, S_7]$, $Q_2^2 = [S_1, S_6]$

$Q_3^2 = [S_0, S_4, S_5]$, $Q_4^2 = [S_3]$

$\pi_2 = \{ [S_2, S_7], [S_1, S_6], [S_0, S_4, S_5], [S_3] \}$

	a	b
S_0	S_1	S_4
S_3	S_3	S_3
S_4	S_1	S_4
S_5	S_1	S_4

level 3 [π_3]:

$Q_1^3 = [S_2, S_7]$, $Q_3^2 = S_1$, $Q_3^3 = S_6$

$Q_4^3 = [S_0, S_4, S_5]$, $Q_5^3 = [S_3]$

S_1	S_5	S_2
S_6	S_3	S_7

$\therefore \pi_3 \{ [S_2, S_7], S_1, S_6, [S_0, S_4, S_5], S_3 \}$

Diagram: