MC-304 Theory of Computation

Assignment - I

Anish Sachdeva
DTU/2K16/MC/13

Q1) Differentiate between finite automata and Transition system.

Ans) A finite automata is a 5-tuple machine $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is finite set of states

Σ is an input alphabet

$\delta : Q \times \Sigma \rightarrow Q$ is the transition function

$q_0 \in Q$ is the initial state, and

F is the set of final states $F \subseteq Q$

M, the finite automata accepts a string w if M when starting in the state q_0 will end any such state $q_f \in F$ (the final states)
so, the set of all strings recognized by the machine M is called the language recognized by the machine (automata) or M recognizes language L.

A transition system is a 3-tuple $(\Gamma, \rightarrow, T)$ where

Γ is a set of configurations (this need to be finite)

$\rightarrow \subseteq \Gamma \times \Gamma$ is a transition relation and

$T \subseteq \Gamma$ is a set of terminal configurations

A variation of this notion is that a labelled transition system.
This is a 4-tuple $(\Gamma, \rightarrow, T, A)$

Γ: the set of transition states (need not be finite)

T: set of the terminal transitions, $T \subseteq \Gamma$

$\rightarrow \subseteq \Gamma \times A \times \Gamma$: relation or labelled relations in the transition system

A: The set of labels (finite)

A finite automaton can be seen as a labelled transition system whose configurations are it's states, whose label set is the input label (or ~~other~~ ~~character~~ alphabet), whose terminal configurations are it's final states and whose transition relation corresponds to its transition function.

i) In a transition function, the set of states isn't necessarily finite, or even countable

ii) ~~In~~ The set of transitions is not necessarily finite, or even countable

iii) No "start" state or "final" states are given.

Q2) Differentiate between DFA and NFA

Ans) A DFA (Deterministic Finite State Automata) is a Machine represented by 5-tuple :-

$M = M(Q, \Sigma, \delta, q_0, F)$

Q: the set of states present in the automata (finite)

Σ : The ~~set~~ alphabet accepted by the automata

δ : The transition function in the automata denoted by.

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

F: The set of all final states (accept states) where $F \subseteq Q$ such that any automata that ends at F will be accepted.

The set of all strings w such that $L = \{ w / M \text{ accepts } w \}$ is called the language (L) of the finite state automata.

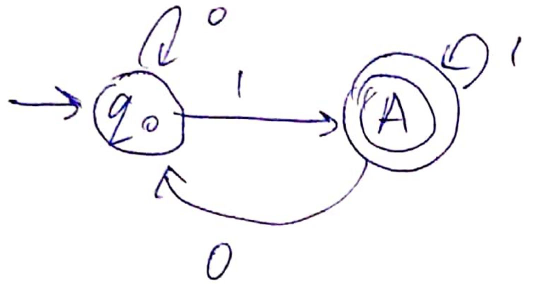
Scanned with CamScanner

Example:

The finite state automata that accepts all strings ~~starting~~ ending with 1:

$L = \{ w \mid w \text{ ends with } 1 \}$

$L = \{ 1, 01, 001, 0001, 1111, 10101 \ldots \}$



This is the finite state automata that accepts all strings that end with 1.

NDFA (Non-deterministic Finite State automata)

A non-deterministic finite state automata is defined by a 5-tuple $M(Q, \Sigma, \delta, q_0, F)$ where

Q: States in the ~~finite state~~ non-deterministic automata

Σ: Alphabet in the automata

δ: This is the transition function defined as:

$\delta: P(Q) \times \Sigma \rightarrow P(Q)$ where P denotes the power set

q_0: q_0 is the initial state or starting state

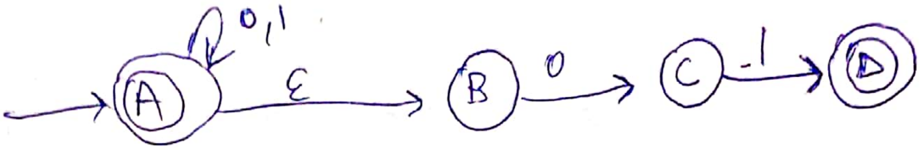
$F \subseteq Q$: The final accept states of the automata.

An example of a non-deterministic automata that accepts all strings that end with {01}s.

$L = \{ 01, 001, 000001, 11101, 1010,1001 \ldots \}$

This is the language that this automata will accept.

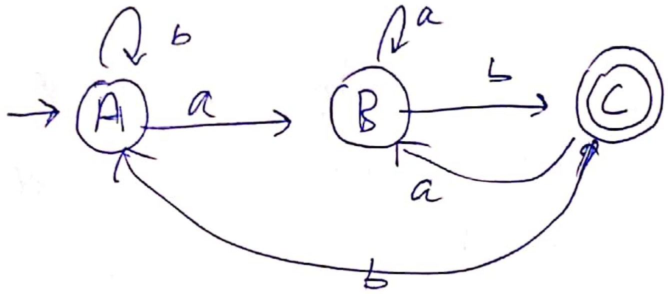
Scanned with CamScanner



This will recognize the language of all strings ending with {01}.

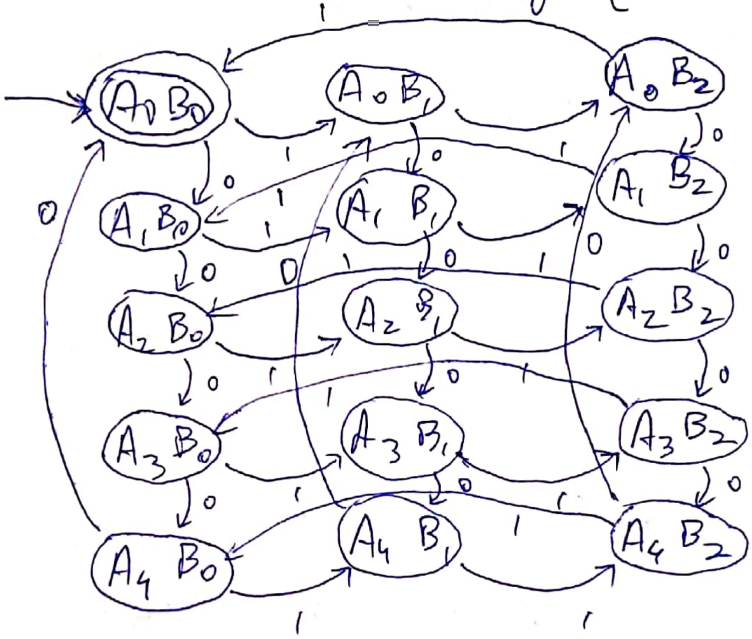
Q3) Give DFA accepting language over {a,b} that have the set of all strings that end with ab.

Ans)



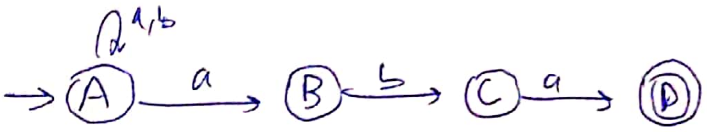
Q4) Give DFA accepting language over {0,1} that have the set of all strings such that the number of 0's is divisible by 5 and the number of 1's is divisible by 3.

Ans) Let there exist state $\{A_i\}$ where $\{A_i \mid n_0(w) \bmod 5 = i\}$ and let there exist states $\{B_j\}$ such that $\{B_j \mid n_1(w) \bmod 3 = j\}$ so our DFA will consist of $\{A_i B_j \mid 0 \le i \le 4,\ 0 \le j \le 2\}$



Q3) Construct an NDFA accepting strings over {a,b} ending in aba. Use it to construct a DFA accepting the same set of strings.

Ans) The NFA that accepts all strings that end with 'aba' can be represented as:-



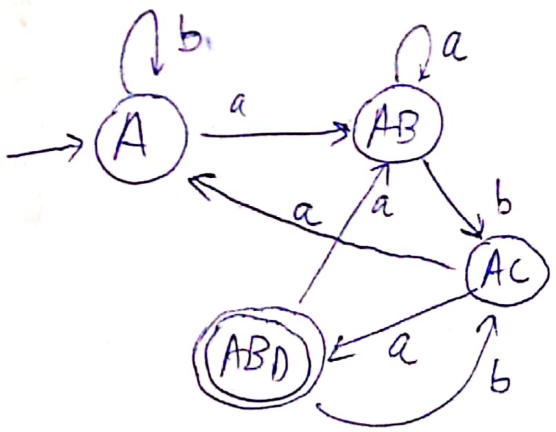
Now, we can represent the transition function δ of this NDFA as:-

$\delta : P(Q) \times \Sigma \to P(Q)$

δ	a	b
A	{A,B}	A
B	φ	C
C	D	φ
D	φ	φ

We will create a new DFA from the NFA using a new transition function δ' such as.

δ'	a	b
q_0: A	{AB}	A
B	φ	C
C	D	φ
(D)	φ	φ
AB	AB	AC
AC	ABD	A
(ABD)	AB	AC



Q6) Construct a Mealy machine equivalent to the Moore machine defined by the table below:

Present state	Next State $a=0$	Next State $a=1$	output
$\rightarrow q_0$	q_1	q_2	1
q_1	q_3	q_2	0
q_2	q_2	q_1	1
q_3	q_0	q_3	1

A moore machine can be constructed into an equivalent mealy machine using the following method.
Let there be a moore machine

$M = (Q, \Sigma, \Gamma, \delta, q_0, F, \lambda)$ where Γ is the output alphabet and λ is the transition function $\lambda : Q \rightarrow \Gamma$.
We will construct new mealy machine with new output transition function λ' such that $\lambda' : Q \times \Sigma \rightarrow \Gamma$

$$\lambda'(q, \sigma) = \lambda(\delta(q, \sigma)) \quad (q \in Q,\ \sigma \in \Sigma,\ \delta \in \text{Moore})$$

λ'	$a=0$	$a=1$
q_0	0	1
q_1	1	1
q_2	1	0
q_3	1	1

$\Rightarrow$ So, equivalent Mealy machine will be

Mealy $= (Q, \Sigma, \Gamma, q_0, F, \lambda')$

Q7) Construct a moore machine equivalent to a mealy machine defined by:-

Present state	Next State: a=0 State	a=0 Output	a=1 State	a=1 Output
$\rightarrow q_1$	q_1	1	q_2	0
q_2	q_4	1	q_4	1
q_3	q_2	1	q_3	1
q_4	q_3	0	q_1	1

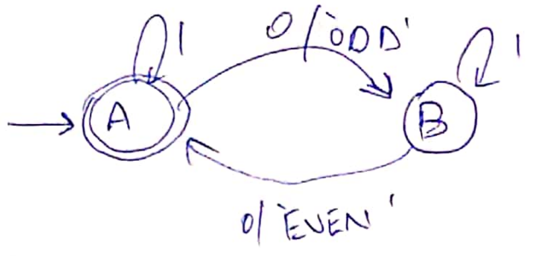
Making equivalent moore machine automata/transition table:-

Present State	Next State: a=0	a=1	output
q_1	q_{11}	q_{20}	1
q_{20}	q_{41}	q_{41}	0
q_{21}	q_4	q_4	1
q_{30}	q_{21}	q_{31}	0
q_{31}	q_{21}	q_{31}	1
q_4	q_{30}	q_1	1

This is the corresponding transition state diagram of the moore machine.

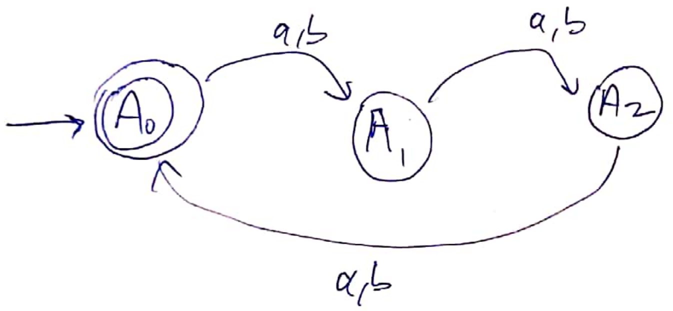
Scanned with CamScanner

Q8): Construct a mealy machine which takes input 0,1 and : output EVEN, ODD according as the total number of 0's encountered is even or odd.

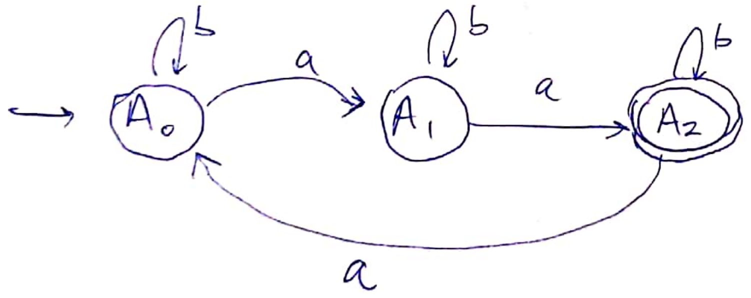


Q9) Find DFA's for the following languages on $\Sigma = \{a, b\}$ where $n_a(w)$ gives number of character a in a string w.

a) $L = \{w \mid |w| \bmod 3 = 0\}$



b) $L = \{W \mid n_a(W) \bmod 3 > 1\}$

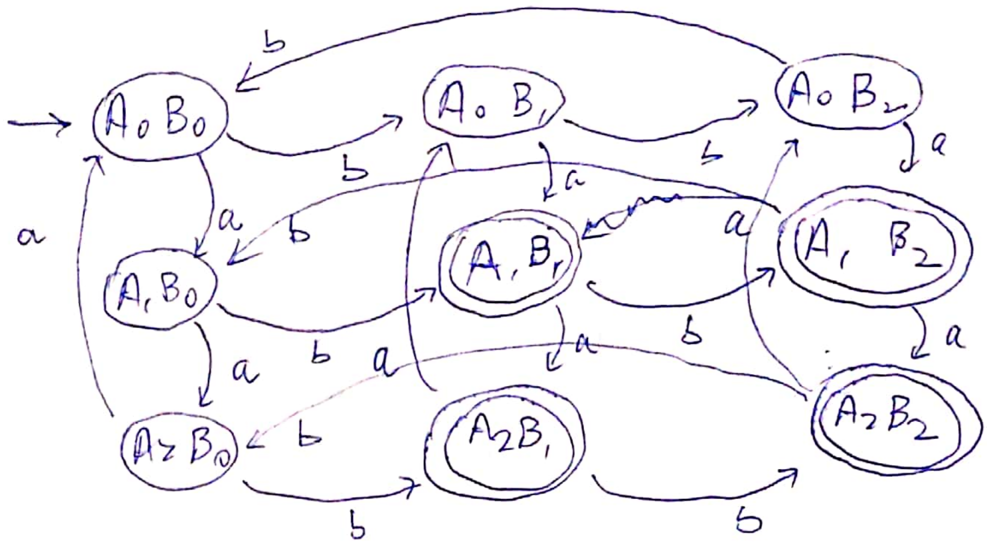


c) $L = \{W \mid n_a(W)\, n_b(W) \bmod 3 > 0\}$

This can also be written as:-

$L = \{W \mid (n_a(W) n_b(W)) \bmod 3 \,!= 0\}$

Let the state {A_i} represent $n_a(w) \bmod 3 = i$ and {B_j} represent $n_b(w) \bmod 3 = j$. We will create DFA where every thing except for {A_0B_0} is inverted.



d) $L = \{ w \mid |w| \bmod 3 = 0,\ |w| \neq 6 \}$