Automata Theory ①

Date 11-1-202
uploaded 13-1-2

Non deterministic Automata :→

Non-Determinism means a choice of moves for an automaton. Rather Than Prescribing a unique move in each situation, we allow a set of Possible moves. formally, we achieve this by defining the transition function so that its range is a set of Possible states.

Definition :— A nondeterministic finite automata or NFA is defined by the quadruple $M = \{Q, \Sigma, \delta, q_0, F)$,

where Q, Σ, q_0, F are defined as for deterministic finite automata, but

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$$

Note that There are Three Major differences b/w this definition and The definition of a dfa.

(1) In a non deterministic automata, The range of δ is in The Powerset 2^Q, so that its value is not a single element of Q, but a subset of it. This subset defines The set of all Possible states that can be reached by The Transition. If for instance, The Current state is q_1, the symbol a is read, and

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

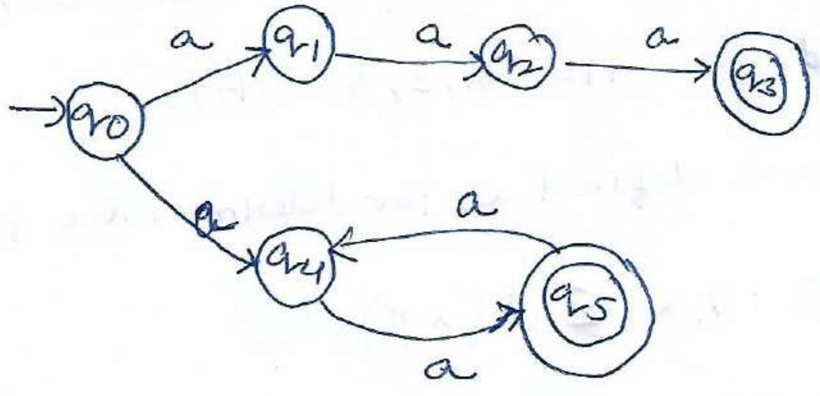
Then either q_0 or q_2 could be The next state of The nfa.

(2) Also we allow λ as The second argument of δ. This Means that nfa can make a Transition without consuming an input symbol.

(3) Finally, in an NFA, The Set $\delta(q_i, a)$ may be empty, Meaning that there is no transition defined for this specified situation.

A string is accepted by an NFA if there is some sequence of possible moves that will put the machine in a final state at the end of the string. A string is rejected (That is, not ~~possible~~ accepted) only if there is no possible sequence of moves by which a final state can be reached.

Ex:- Consider the Transition graph in figure, it describes a nondeterministic automata since there are two transition labeled a out of q_0.



Ex:- Transition system for a nondeterministic automation

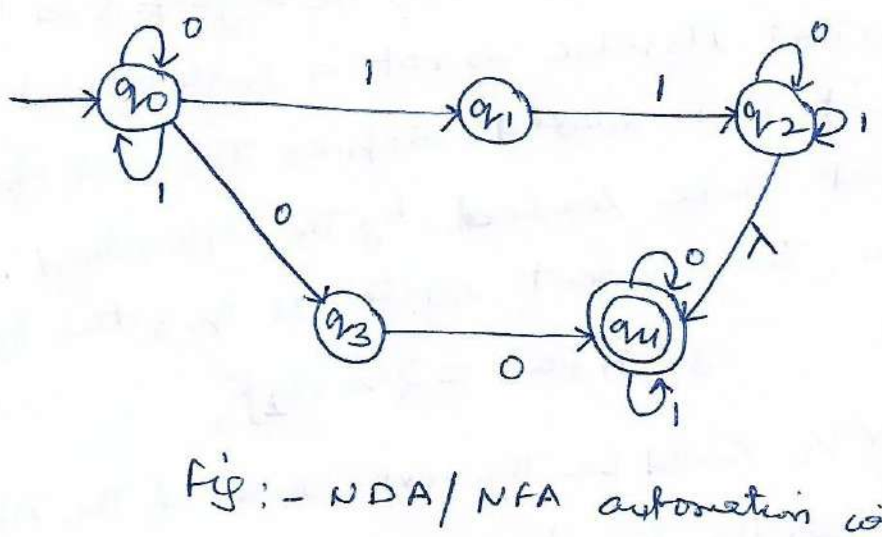
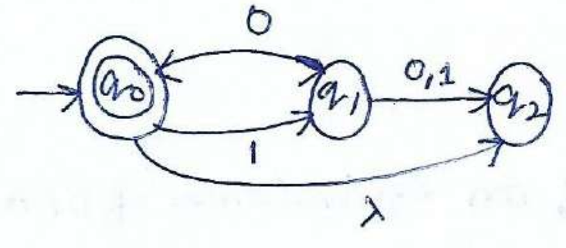


Fig:- NDA/NFA automation with empty move

Example:- A nondeterministic automaton is shown in figure. It is nondeterministic not only because several edges with the same label originate from one vertex, but also because it has λ- Transition. Some transitions, such as $\delta(q_2, 0)$, are unspecified in the graph. This is to be interpreted as a Transition to the empty set, that is, $\delta(q_2, 0) = \phi$. The automaton accepts strings λ, 1010, and 101010, but not 110 and 10100. Note that for 10 there are two alternative walks, one leading to q_0, the other to q_2. Even though q_2 is not a final state, the string is accepted because one walk leads to a final state.



Recap:-

Finite state Machine (FSM)

- FA with O/P
  - Mealy machine
    - output with input
  - Moore Machine
    - output is associated with states
- FA without O/P
  - λ-NFA
    - $Q \times \Sigma \cup \{\lambda\} \to 2^Q$
  - NFA
    - $Q \times \Sigma \to 2^Q$
  - DFA
    - $Q \times \Sigma \to Q$

(9) The equivalence of DFA and NDFA :- we naturally

Try to find the relation b/w DFA and NDFA. Intuitively, we now feel that:

(i) A DFA can simulate the behaviour of NDFA by increasing the number of states. (In other words, a DFA $(Q, \Sigma, \delta, q_0, F)$ can be viewed as an NDFA $(Q, \Sigma, \delta', q_0, F)$ by defining $\delta'(q, a) = \{\delta(q, a)\}$.)

(ii) Any NDFA is a more general machine without being more powerfull.

* we now give a Theorem ~~of~~ on equivalence of DFA and NDFA.

Theorem :-

For every NDFA, There exists a DFA which simulates The behaviour of NDFA. Alternatively, if L is The set accepted by NDFA, Then There exists DFA which accepts L.

Proof :- let $M = (Q, \Sigma, \delta, q_0, F)$ be an NDFA accepting L. We construct a DFA M' as:

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

where :- (i) $Q' = 2^Q$ (any state in Q' is denoted by $[q_1, q_2, \ldots q_j]$, where $q_1, q_2, \ldots q_j \in Q$

(ii) $q_0' = [q_0]$ and

(iii) F' is The set all subsets of Q containing an element of F.

(5) (IV) $\delta'([q_1, q_2, \cdots q_i], a) = \delta(q_1, a) \cup \delta(q_2, a)$

$\cup\, \delta(q_3, a) \cup \cdots\cdots \cup \delta(q_i, a).$

Equivalently,

$\delta'([q_1, q_2 \cdots q_i], a) = [P_1, \cdots P_j]$

If and only if $\delta(\{q_1, \cdots q_i\}, a) = \{P_1, P_2 \cdots P_j\}$

Example:— Construct a deterministic automaton equivalent t

$M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_0\})$

where δ is defined by its state table (a)

(a)

State/Σ	0	1
$\rightarrow (q_0)$	q_0	q_1
q_1	q_1	q_0, q_1

Solution:— For the deterministic automaton M_1,

(i) The states are subset of $\{q_0, q_1\}$, i.e, ϕ, $[q_0]$, $[q_1]$, $[q_0, q_1]$;

(ii) $[q_0]$ is the initial state;

(iii) $[q_0]$ and $[q_0, q_1]$ are the final states as these are the only states containing q_0; and

(iv) δ is defined by the state table given by Table (a).

⑥

Table:– State Table of M_1 for example

State/Σ	0	1
ϕ	ϕ	ϕ
$[q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1]$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_0, q_1]$

The states q_0 and q_1 appear in the rows corresponding to q_0 and q_1 and the column corresponding to 0.

~~sof~~ so, $\delta([q_0, q_1], 0) = [q_0, q_1]$.

When M has n states, the corresponding finite automaton has 2^n states. However, we need not construct δ for all these 2^n states, but only for those states that are reachable from $[q_0]$.

This is because our interest is only in constructing M1 accepting T(M). So, we start the construction of δ for $[q_0]$. We continue by considering only the states appearing earlier under the I/P columns and constructing δ for such states. We halt when no more new states appear under the input columns

Example:- find a deterministic acceptor equivalent to

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

where δ is given by table given below

State/Σ	a	b
$\rightarrow q_0$	q_0, q_1	q_2
q_1	q_0	q_1
q_2	.	q_0, q_1

solution:- The deterministic automaton M_1 equivalent to M is defined as follows:

$$M_1 = (2^Q, \{a, b\}, \delta, [q_0], F')$$

where

$$F = \{[q_2], [q_0, q_2], [q_1, q_2], [q_0, q_1, q_2]\}$$

we start the construction by considering $[q_0]$ first, we get $[q_2]$ and $[q_0, q_1]$. Then we construct δ for $[q_2]$ and $[q_0, q_1]$. $[q_1, q_2]$ is a new state appearing under the input columns. After constructing δ for $[q_1, q_2]$, we donot get any new states and so we terminate the construction of δ. The state table is given by following.

Table:- State table of M_1 for above ex.

State/Σ	a	b
$[q_0]$	$[q_0, q_1]$	$[q_2]$
$[q_2]$	ϕ	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_0]$	$[q_0, q_1]$

(7) Ex:- Construct a deterministic finite automaton equivalent to

$$M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_3\})$$

where δ is given by table following

Table:- State table

state/Σ	a	b
$\rightarrow q_0$	q_0, q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3		q_2

Soln:- let $Q = \{q_0, q_1, q_2, q_3\}$ then the DFA M_1 equivalent to M is given by $M_1 = (2^Q, \{a, b\}, \delta, [q_0], F)$

where F consists of:

$[q_3]$, $[q_0, q_3]$, $[q_1, q_3]$, $[q_2, q_3]$, $[q_0, q_1, q_3]$, $[q_1, q_2, q_3]$ and $[q_0, q_1, q_2, q_3]$

and where δ is defined by given state table for M_1

Table :- state table for M_1

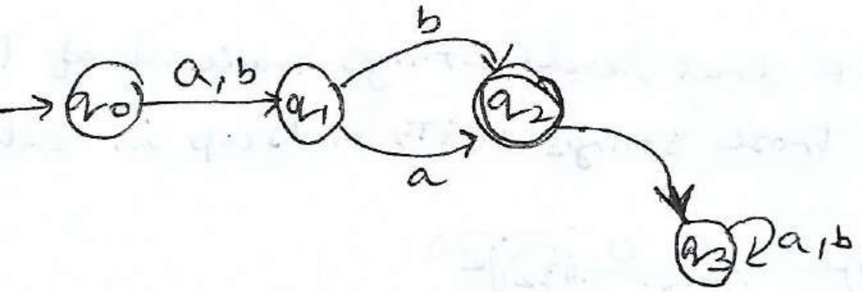
State/Σ	a	b
$[q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_3]$
$[q_0, q_1, q_3]$	$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$	$[q_0, q_1, q_2, q_3]$

Some question on DFA

Q. Construct a DFA, that accepts set of all string over $\Sigma = \{a, b\}$ of length 2

Sol^n:-

$L = \{aa, ab, ba, bb\}$

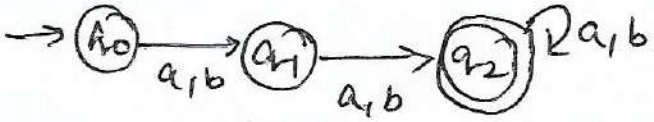


for if abb or aba case occurs

Q. Construct a DFA, that accept set of all strings over $\Sigma = \{a, b\}$ where length is atleast 2.

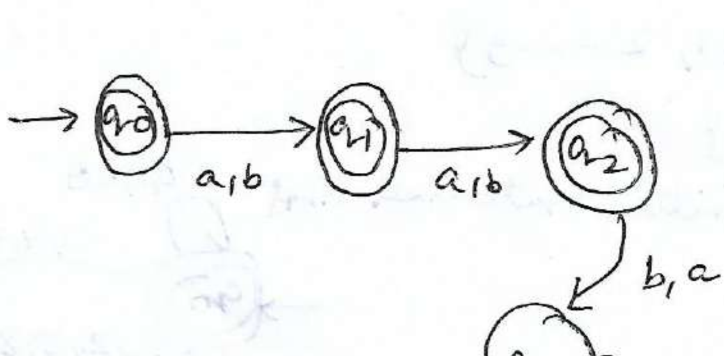
$|w| \geq 2$

Sol^n:- $L = \{aa, ab, ba, bb, aaa, aab, \ldots\}$



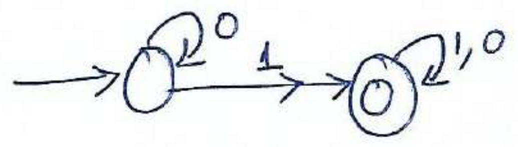
Q. $\Sigma = \{a, b\}$, $|w| \leq 2$

Sol^n:- $L = \{\epsilon, a, b, aa, ab, ba, bb\}$



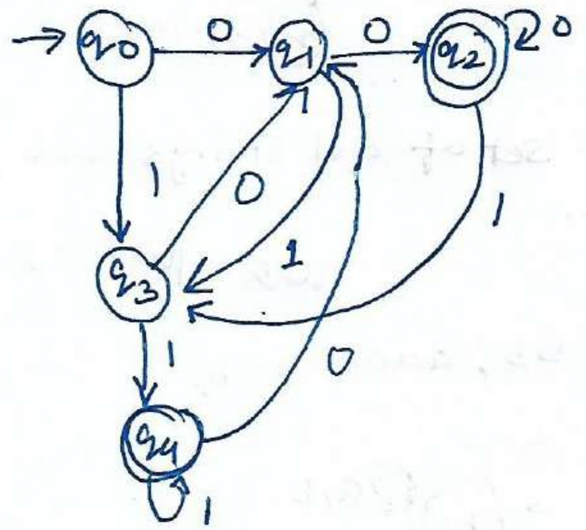
* if ϵ is the Part of the lang then always makes initial state as final state

Regular expression:→



where $\Sigma = \{0,1\} \Rightarrow 0^*1(0+1)^*$

* Design a DFA that reads strings made up of {0,1} and accept only those strings which ends up in either 00 or 11.



Here the FA has two different final states q_2 and q_4. q_2 state accepts string ending with 11.

* Construct a DFA that accepts the set of natural numbers x which are divisible by 3.

soln:- let $M = \{Q, \Sigma, q_0, \delta, F\}$ be a DFA with

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1, 2 --- 9\}$

$F = \{q_0\}$

i.e. here q_0 is initial state and final state also.