Disadvantages of DFA

(*) constructing some of the DFA's is difficult
(*) DFA can not guess about its input
(*) DFA is not very powerful
(*) DFA is in only one state at any point of time ⇒ there may be need of being in many states

Why NFA ? (Non-deterministic Finite Automata)

- Computers are completely deterministic machines
- state of a computer can be predicted from its input
- however, disadvantages of DFA's can be over-come by having NFA's.

Reasons

(*) Construction of NFA is very simple
(*) NFA has ability to guess something about its input
(*) power powerful (vast applications)
(*) power to be in several states at any instant
(*) NFA is efficient in describing complicated languages.

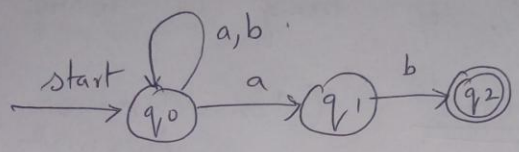
NFA is a 5-tuple

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

- Q: non-empty set of states
- Σ: Alphabet (input)
- q_0: start state
- F: set of final states

δ is the transition function. $(Q \times \Sigma)$ and is a mapping from $Q \times \Sigma$ to 2^Q.
Machine can exist in one or more states.

Ex:



$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$q_0 \leftarrow$ start state
$F = \{q_2\}$
δ

δ	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	ϕ	$\{q_2\}$
q_2	ϕ	ϕ

TT.

Language accepted by a NFA

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFA.
- A string w is accepted by the machine M, if it takes the machine from initial configuration (q_0, w) to the final configuration of the form $(\{p, q, r, \ldots t\}, \varepsilon)$ denoted by

$(q_0, w) \vdash (\{p, q, r, \ldots, t\}, \varepsilon)$

$$L(M) = \{ w \mid w \in \Sigma \text{ and } (q_0, w) \vdash (\{p, q, r \cdots t\}, \varepsilon) \}$$

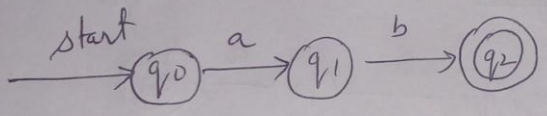
one of these is the final state

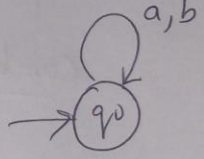
Ex Obtain an NFA to accept strings of a's and b's ending with ab.

Sol. minimum string = ab.

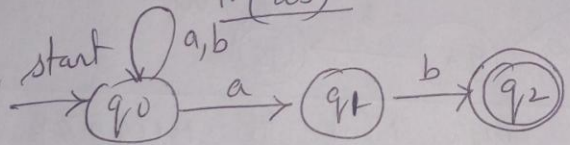
maximum " = $(ab)(ab)^*$

Where * indicates zero or multiple occurences.





1. (ab)



2. $(ab)^*$

any number of a's and b's

3. $(ab)(ab)^*$

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

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

$\Sigma = \{a, b\}$

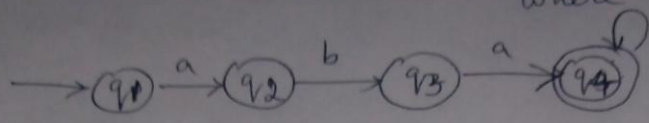
$q_0 \leftarrow$ start state

$F = \{q_2\}$

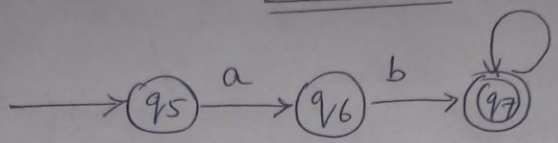
δ	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0
q_1	ϕ	q_2
q_2	ϕ	ϕ

Ex Obtain an NFA to accept the language (4)

$L = \{ w \mid w \in abab^n \text{ or } aba^n \text{ where } n \geq 0 \}$.

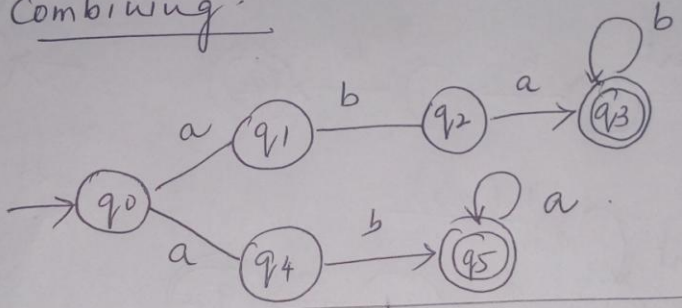


$abab^n \quad n \geq 0$

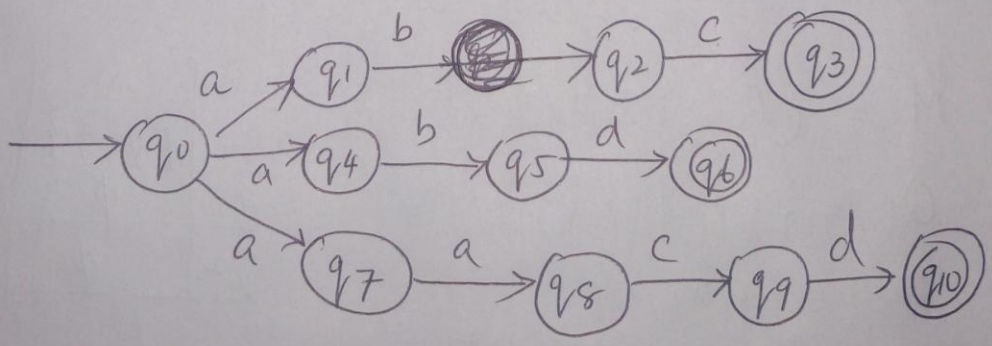


$aba^n \quad n \geq 0$

Combining:



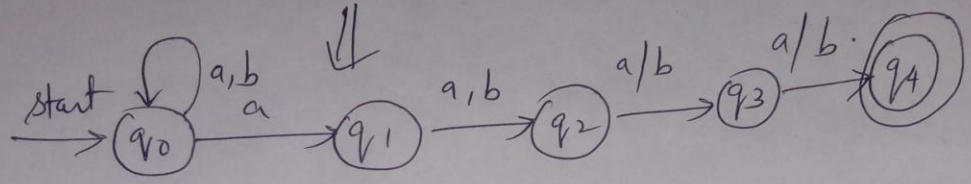
Ex Design an NFA to recognize abc, abd, aacd.



Ex Obtain an NFA to accept strings of a's and b's such that 4th symbol from the right side is a

(a/b) a (a/b) (a/b) (a/b)

5th 4th 3rd 2nd 1st ←



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

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{a, b\}$

q_0 ← start state

f ← q_4

δ:

S	a	b
→ q_0	$\{q_0, q_1\}$	q_0
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_4	q_4
* q_4	ϕ	ϕ