
KTU
NOTES

Any Language that has Alphabets is a formal Language.
Semantic Analysis : Language that has no meaning.

Alphabet, strings & Languages.

- **Alphabets:**
  Represented by "Σ"

  $\Sigma = \{0, 1\}$

  Sigma Σ $\qquad$ $\Sigma = \{a, b, \ldots, z\}$
  $\{A, B, \ldots, Z\}$

  - Finite set
  - Non-empty set of symbols.

- **Strings:**
  sequence of symbols defined over Σ

  $\{a, b\}$ ^ a b ab ba aba bab - all are strings defined over

  Length of string 'ω' over $\Sigma = \{0, 1\}$
  $\omega = 0001$

  Length : $|\omega| = 4$

  $\omega = 0$, $|\omega| = 1$

  Null string:
  Represented by ε

$\Sigma = \{a, b\}$
$\Sigma^0 = \varepsilon$
$\Sigma^1 = \{a, b\}$
$\Sigma^2 = \{aa, bb, ba, bb\}$

Kleene closure : Σ^*

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots\ldots$

- contain all strings that can be defined over an alphabet including null string.

Positive closure : $\Sigma^+ = \Sigma^* - \{\varepsilon\}$

contain all strings that can be defined over an alphabet excluding null string.

Prefix of a string

- removing zero or more trailing symbols

$w = aba$

a, ab, aba

Suffix of a string

- removing zero or more leading symbols.

$w = aba$

a, ba, aba

Substring:

Eg:- $w = aba$

b, ab substring

aa is not a substring.

Language :

A Language L over Σ and L is a subset of Σ^*.

$L = \{a, aa, ab, aa, \ldots\ldots aaa, \ldots\}$

infinite Language.

$\Downarrow$

~~String starting with a.~~

Empty Language:

Language contain epsilon as member

$\phi \neq \{\varepsilon\}$

Concatenation of strings.

Let w_1 and w_2 be 2 strings, a concatenated string w is defined as

$w = w_1 . w_2$

$w_1 w_2 \neq w_2 w_1$

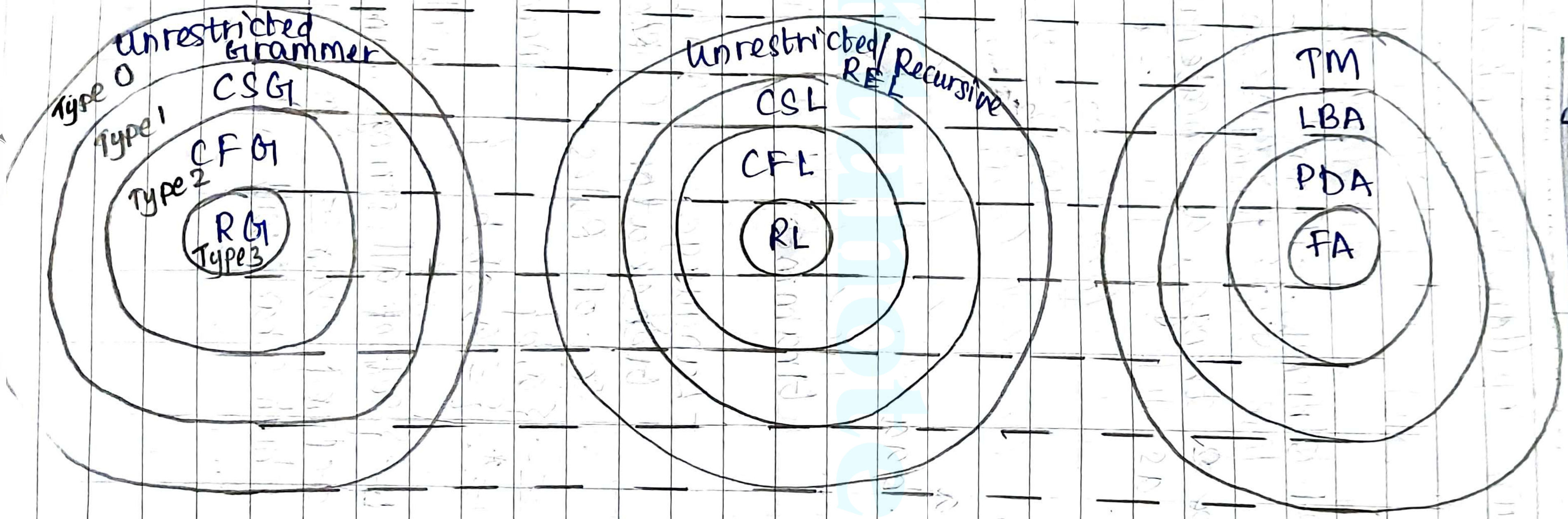
eg: $w_1 = aab$

$w_2 = baaa$

$w_1 . w_2 = aab.baa$

Language	Grammer	Automata
- Subset of Σ^*	- Language Generator - Set of Rules $\Sigma = \{a, b\}$ $\Sigma^* = \{\varepsilon, a, ab, b, ..\}$ L = { set of all strings defined over Σ which strarts with 'a' } Rule	- Language Recognizer/Acceptor - Membership - will give two answers eg: yes / no.

Chomsky classification of Languages:



RG - Regular Grammer
CFG - Context free Grammer
CSG - Context Sensitive Grammer

RL - Regular Language
CFL - Context Free Language
CSL - Context Sensitive Language.
REL - Recursive Enumerable Language

FA - Finite Automata
PDA - Pushdown Automata
LBA - Linear Bound Automata
TM - Turing machine

Finite Automata (FA)

- FA with output
  - Moore Machine
  - Mealey Machine
- FA without output
  - Deterministic FA (DFA or DFSM)
    DFSM - Deterministic Finite state machine
  - Non-Deterministic FA (NFA or NFSM)

Symbols in FA

 - States

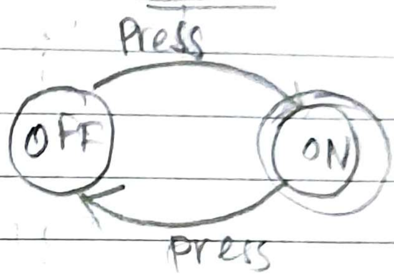
 - Initial state

 - final state / Accepting state.

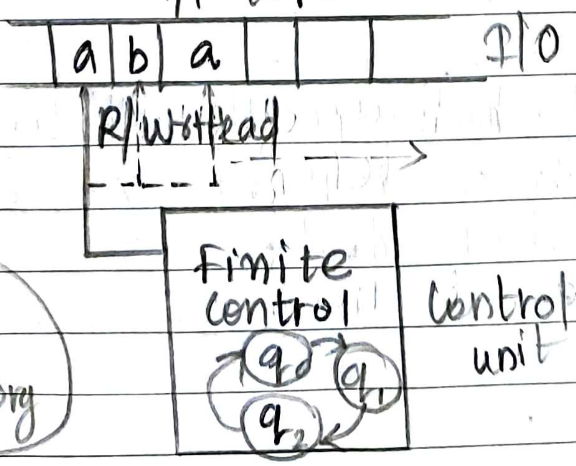
 - Transition from q_0 to q_1 on a.

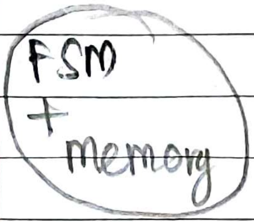
Press - Input signal

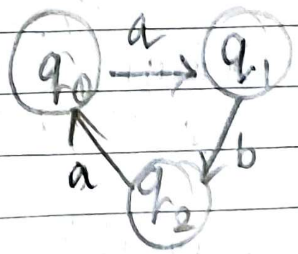
Electric switch



FSM - Finite state machine.







I/O
Control unit
Memory.

Deterministic FA

Transition $\begin{cases} f(x) = y \\ \text{OFF, press} \longrightarrow \text{ON} \\ \delta(Q \times \Sigma) \longrightarrow Q \end{cases}$

FA :

5 tuples.

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

$Q = \{ON, OFF\}$
$\Sigma = \{press\}$
$q_0 = OFF$
$F = \{ON\}$

Q = Finite set of states
Σ - Alphabet
δ = Transition function.
q_0 = Initial state and a member of Q.
$q_0 \in Q$.

F = Set of finite state.
$F \subseteq Q$.

	Press
→OFF	ON
*ON	OFF

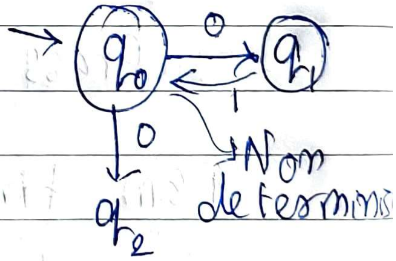
$(OFF, press) \longrightarrow (ON)$

$\delta(Q, \Sigma) \longrightarrow Q$

$\delta(ON, Press) \longrightarrow OFF$

$q_0 \xrightarrow{1} q_1$
transition.

Transition.
$\delta(Q \times \Sigma) \longrightarrow Q$



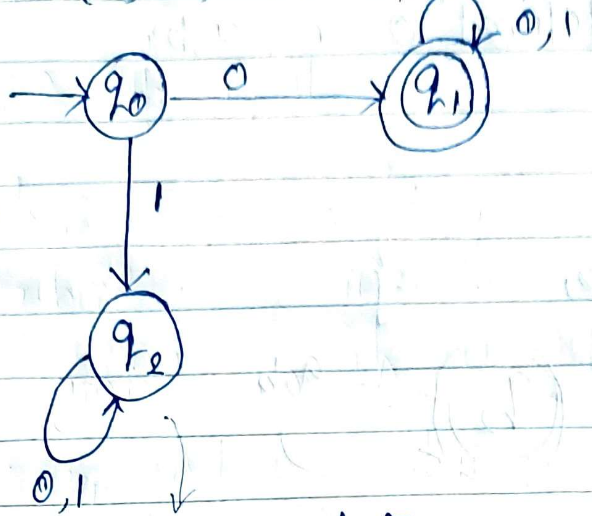
The machine is not a valid DFA

Q. Design a DFA which accepts all the strings starting with 0 defined over {0, 1}

(Alphabet → The set from which we take inputs)

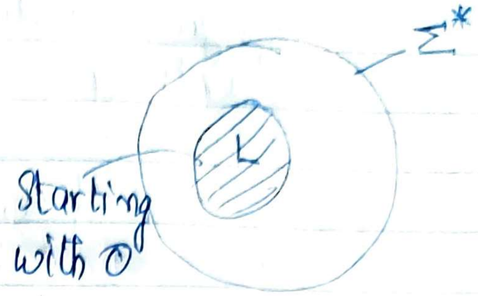
$\Sigma = \{0, 1\}$

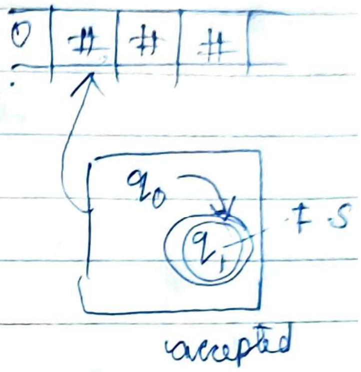
$L = \{0, 01, 00, 000, 010, \ldots\}$



proper DFA

non accepting state
Also known as TRAP state/Dead state.





0 ——
1 —— Not accepted.

$\delta(\text{state}, \text{symbol}) \rightarrow \text{state}$

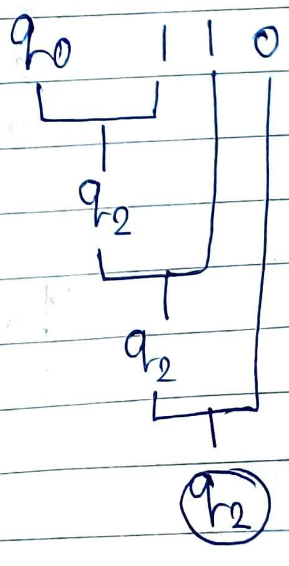
$\delta(q_0, 0) \rightarrow q_1$
$\delta(q_0, 1) \rightarrow q_2$
$\delta(q_1, 1) \rightarrow q_1$
$\delta(q_1, 0) \rightarrow q_1$
$\delta(q_2, 0) \rightarrow q_2$
$\delta(q_2, 1) \rightarrow q_2$

q_0 110 × → Can't accept.

should not have multiple transition for same input
⇒ ∴ DFA

$q_0 = q_0$
$F = \{q_1\}$

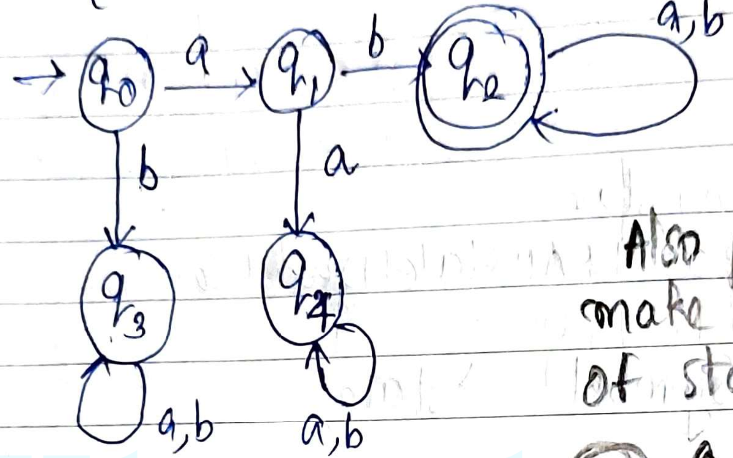
	0	1
$\rightarrow q_0$	q_1	q_2
$* q_1$	q_1	q_1
q_2	q_2	q_2



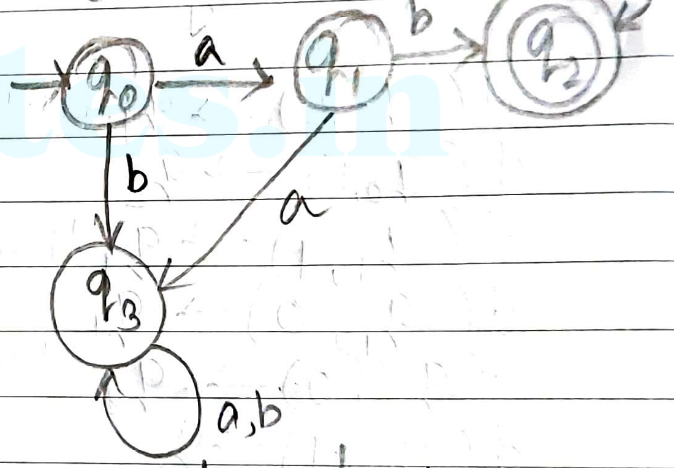
Design a DFA that accepts all the strings starting with ab defined over {a,b}

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

$L = \{ab, aba, abb, \ldots\}$



Also possible to make minimum number of states.



$\delta(q_0, a) \rightarrow q_1$
$\delta(q_0, b) \rightarrow q_3$
$\delta(q_1, a) \rightarrow q_4$
$\delta(q_1, b) \rightarrow q_2$
$\delta(q_2, a) \rightarrow q_2$
$\delta(q_2, b) \rightarrow q_2$
$\delta(q_3, a) \rightarrow q_3$
$\delta(q_3, b) \rightarrow q_3$
$\delta(q_4, a) \rightarrow q_4$
$\delta(q_4, b) \rightarrow q_4$

	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_4	q_2
$* q_2$	q_2	q_2
q_3	q_3	q_3
q_4	q_4	q_4

$q_0 = q_0$
$F = \{q_2\}$

Alphabet (Σ)
Strings

Language

generated by → Grammar (Language Generator)

Recognized by → Automata (Language Acceptor)

- Regular Language
- Context free Language
- Context sensitive Language
- Recursive Language
- Recursive Enumerable Language

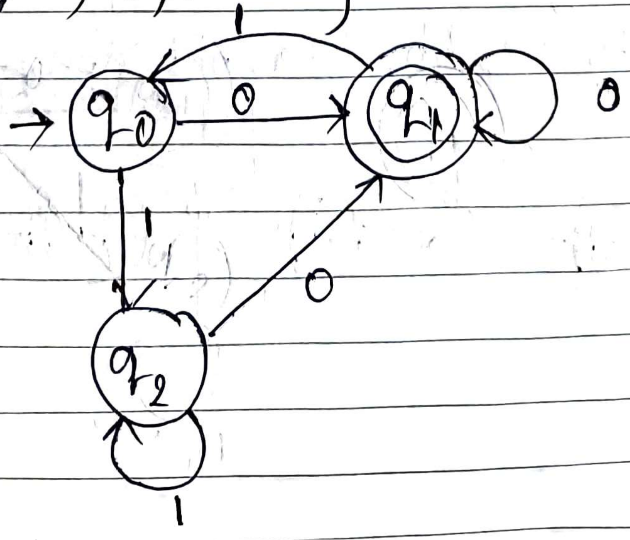
Machine correspond to Regular Language is finite Automata.

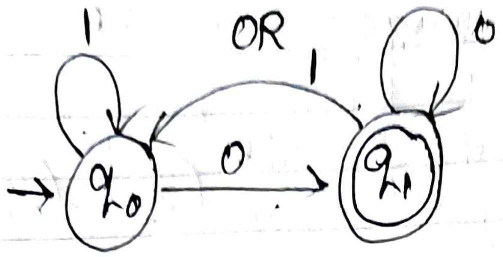
- In DFA, there should be transition for all i/p symbols from all states.
- In DFA there should be exactly one transition for a particular i/p symbol from a particular state. Determinism
- Design a DFA that accepts all the strings ending with zero defined over {0, 1}

$\Sigma = \{0, 1\}$

$(0|1)^* 0$

$L = \{0, 00, 10, \ldots\}$





no dead state because here string is infinite.

$\delta(q_0, 0) \rightarrow q_1$
$\delta(q_0, 1) \rightarrow q_0$
$\delta(q_1, 0) \rightarrow q_1$
$\delta(q_1, 1) \rightarrow q_0$

	0	1
$\rightarrow q_0$	q_1	q_0
$* q_1$	q_1	q_0

- Design a DFA which accepts a Language L defined over {a, b} which has strings of length atmost 3.

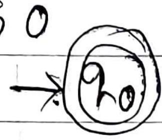
a) - atmost 3
b) - atleast 3
c) - exactly 3

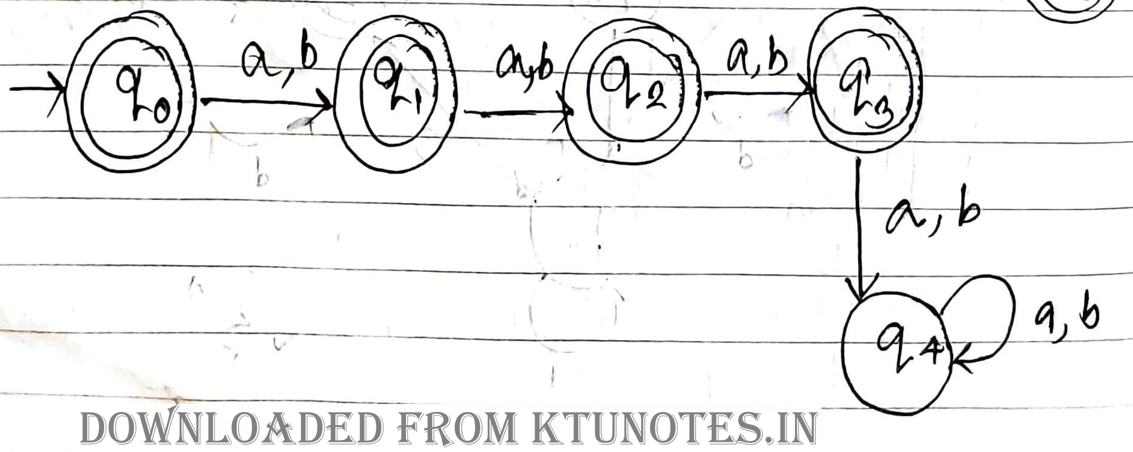
a)

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

$|w| = 0, 1, 2, 3$
$0 \rightarrow \varepsilon$

$L = \{\varepsilon, a, b, aa, bb, aaa, bbb, aba, \ldots\}$ length 0 $\Rightarrow$





$\delta(q_0, a) \rightarrow q_1$ | $\delta(q_1, a) \rightarrow q_2$ | $\delta(q_2, a) \rightarrow q_3$

$\delta(q_0, b) \rightarrow q_1$ | $\delta(q_1, b) \rightarrow q_2$ | $\delta(q_2, b) \rightarrow q_3$

$\delta(q_3, a) \rightarrow q_4$ | $\delta(q_4, a) \rightarrow q_4$

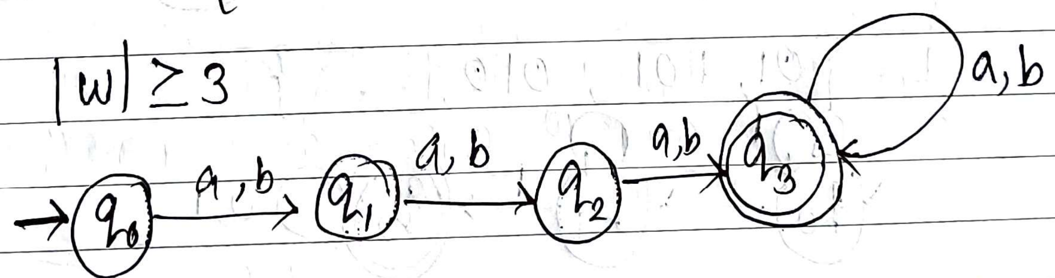
$\delta(q_3, b) \rightarrow q_4$ | $\delta(q_4, b) \rightarrow q_4$

	a	b
$\rightarrow *q_0$	q_1	q_1
$*q_1$	q_2	q_2
$*q_2$	q_3	q_3
$*q_3$	q_4	q_4
q_4	q_4	q_4

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

$L = \{aaa, aba, bbb, bba, \ldots\}$

$|w| \geq 3$



$\delta(q_0, a) \rightarrow q_1$; $\delta(q_1, a) \rightarrow q_2$; $\delta(q_2, a) \rightarrow q_3$

$\delta(q_0, b) \rightarrow q_1$; $\delta(q_1, b) \rightarrow q_2$; $\delta(q_2, b) \rightarrow q_3$

$\delta(q_3, a) = q_3$

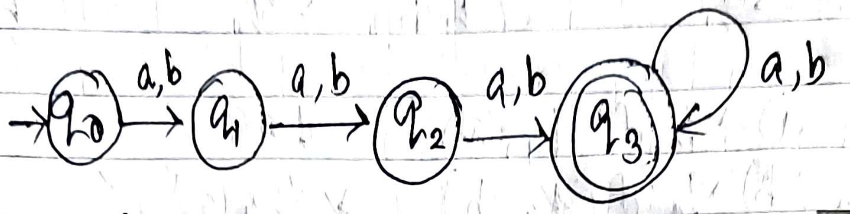
$\delta(q_3, b) = q_3$

	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
$*q_3$	q_3	q_3

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

$|w|$

$L = \{aaa, aab, abb, aba, bab, bbb, baa, \ldots\}$



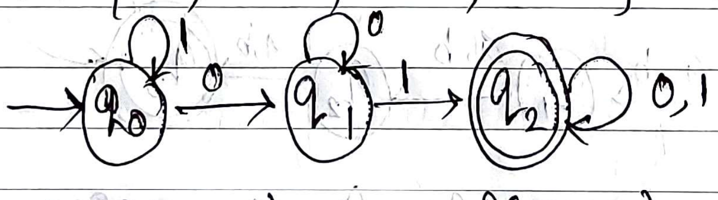
$\delta(q_0, a) \rightarrow q_1$
$\delta(q_0, b) \rightarrow q_1$
$\delta(q_1, a) \rightarrow q_2$
$\delta(q_1, b) \rightarrow q_2$
$\delta(q_2, a) \rightarrow q_3$
$\delta(q_2, b) \rightarrow q_3$
$\delta(q_3, a) \rightarrow q_3$
$\delta(q_3, b) \rightarrow q_3$

	a	b
q_0	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_3	q_3

- Design a DFA for the language L defined over Σ {0,1} which contains substring 01.

$\Sigma = \{0, 1\}$

$L = \{01, 101, 010, \ldots\}$



$\delta(q_0, 0) \rightarrow q_1$
$\delta(q_0, 1) \rightarrow q_0$
$\delta(q_1, 0) \rightarrow q_1$
$\delta(q_1, 1) \rightarrow q_2$

$\delta(q_2, 0) \rightarrow q_2$
$\delta(q_2, 1) \rightarrow q_2$

	0	1
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_2	q_2

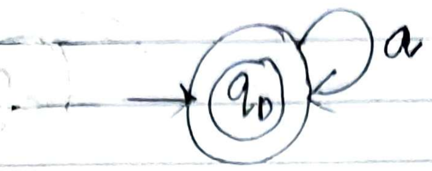
- Design a DFA for the Language $L = \{a^n \mid n \geq 0\}$ defined over $\{a\}$

When we have ε in language directly make initial & final state

$\{ a^0, a^1, a^2 \ldots \}$

$\{ \varepsilon, a, aa, aaa, \ldots \}$

$L = \{ a, aa, aaa \ldots \}$ no ε here



Non-Deterministic Finite Automata (NFA)

- formally defined as a quintuple (5 tuples)

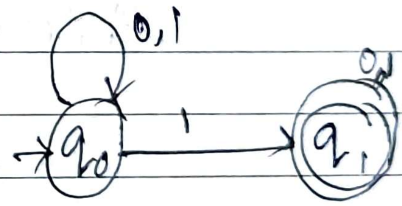
Q - Finite non-empty set of states.

Σ - finite non empty set of input symbol or alphabet

δ - Transition Function: $\delta : Q \times \Sigma \rightarrow P(Q)$

q_0 - Initial state

F - set of final states.



$\delta(q_0, 1) = ?$

$\delta(Q \times \Sigma) \rightarrow 2^Q$

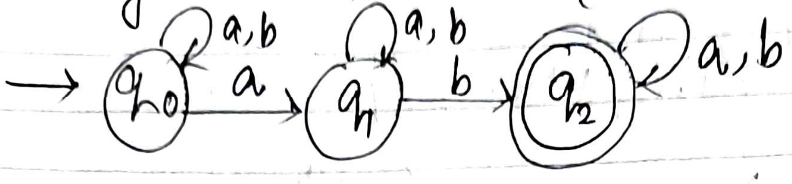
Non deterministic

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

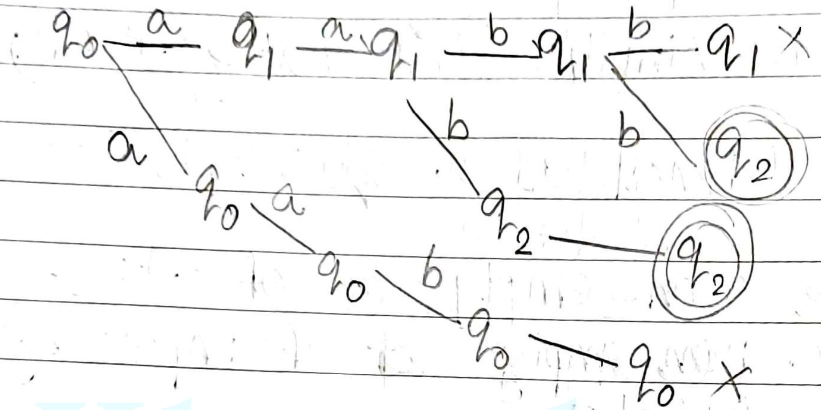
$Q = \{q_0, q_1\}$

$P(Q) = \{\{q_0\}, \{q_1\}, \{\}, \{q_0, q_1\}\}$

Design an NFA for the language consisting of all the strings that has ab as a substring defined over $\{a,b\}$



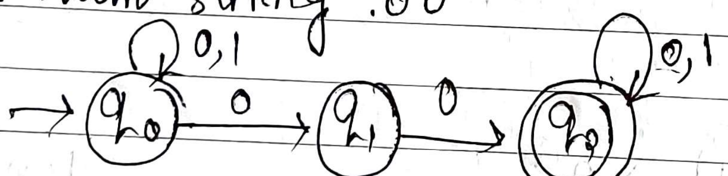
aabb



- Design an NFA for the language containing substring 00.

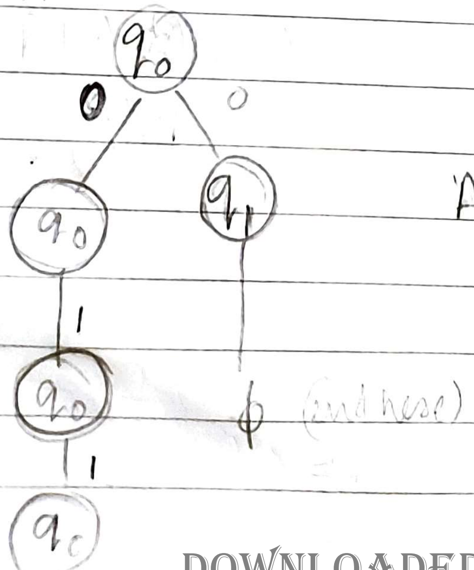
$(0|1)^* 00 (0|1)^*$

minimum string : 00



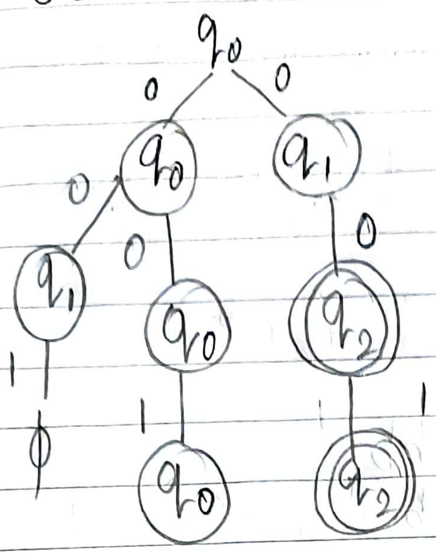
	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	ϕ
$* q_2$	$\{q_2\}$	$\{q_2\}$

011



Any of the leaf node

001



Extended Transition Function :

Defined as $\hat{\delta}$ (Delta Cap or Delta Hat)

$\hat{\delta}(q_0, w) = P$

where $w \in \Sigma^*$

$P \in Q$

formally

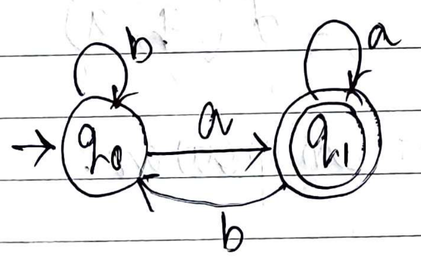
Let $w = xa$

where a is the last symbol and x is the string prefix to 'a'.

then

$$\hat{\delta}(q_0, xa) = \delta(\hat{\delta}(q_0, x), a)$$

eg:



aabb

$$\hat{\delta}(q_0, aabb) = \delta(\hat{\delta}(q_0, aab), b)$$

$$= \delta(\delta(\hat{\delta}(q_0, aa), b, b))$$

$$= \delta(\delta(\delta(\hat{\delta}(q_0, a), a, b, b)))$$

$$= \delta(\delta(\delta(\delta(q_0, a), a, b, b)))$$

$$= \delta(\delta(\delta(q_1, a), b, b))$$

$= \delta(\delta(q_1, b), b)$

$= \delta(q_0, b)$

$= \underline{\underline{q_0}}$

Another method

$q_0\, aabb$

$\vdash a q_1 a bb$

$\vdash aa q_1 bb$

$\vdash aab q_0 b$

$\vdash aabb q_0$

$$a = \varepsilon . a$$

$$\hat{\delta}(q, \varepsilon a) = \delta(\hat{\delta}(q, \varepsilon), a)$$

$$= \delta(q, a)$$

$$\therefore \hat{\delta}(q, \varepsilon a) = \hat{\delta}(q, a)$$

$$= \delta(q, a)$$

DFA & NFA are equivalent in power.

DFA $\cong$ NFA

DFA = NFA

NFA = DFA

Every DFA is By default an NFA.

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

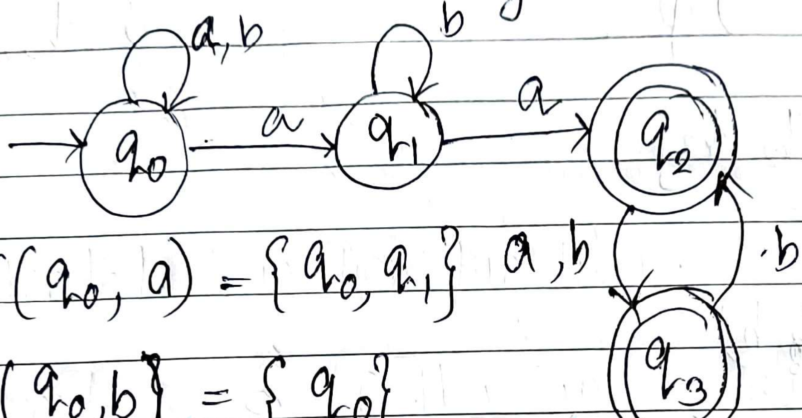
$\delta(Q \times \Sigma) \to 2^Q$

$\therefore \; Q \subseteq 2^Q$

DFA $\subseteq$ NFA

For every NFA we draw, there is an equivalent DFA.

- Convert the following NFA to DFA.



Method used is called Subset Construction

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

$\delta(q_0, b) = \{q_0\}$

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

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

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

Transition for new state:

$\delta(q_0 q_1 q_2, a)$

$\rightarrow \delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a)$

$\rightarrow \{q_0 q_1\} \cup \{q_2\} \cup \{q_3\}$

$\rightarrow \{q_0 q_1 q_2 q_3\}$

	a	b
$\rightarrow \{q_0\}$	$\{q_0 q_1\}$	$\{q_0\}$
$\{q_0 q_1\}$	$\{q_0 q_1 q_2\}$	$\{q_0 q_1\}$
$*\{q_0 q_1 q_2\}$	$\{q_0 q_1 q_2 q_3\}$	$\{q_0 q_1 q_3\}$

$\delta(q_0 q_1 q_2, b) = \delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_2, b)$

$= \{q_0\} \cup \{q_1\} \cup \{q_3\}$

$\{q_0 q_1 q_3\}$

$\delta(q_0q_1q_2q_3, a)$

$= \delta(q_0, a) \cup \delta(q_1, a)$
$\cup\, \delta(q_2, a)$
$\cup\, \delta(q_3, a)$
$= \{q_0, q_1\} \cup \{q_2\} \cup \{q_3\}$
$\cup\, \phi$

$\delta(q_3, a) = \phi$

$= \{q_0q_1q_2q_3\}$

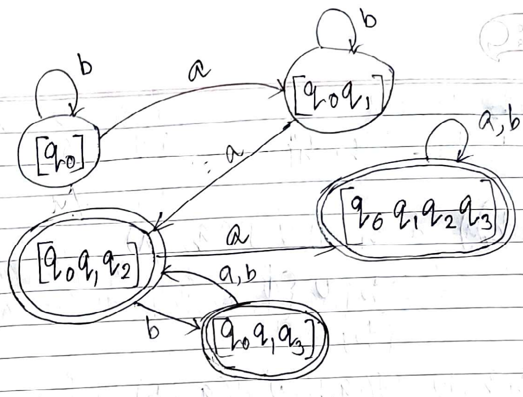
	a	b
$*\{q_0q_1q_2q_3\}$	$\{q_0q_1q_2q_3\}$	$\{q_0q_1q_2q_3\}$
$*\{q_0q_1q_3\}$	$\{q_0q_1q_2\}$	$\{q_0q_1q_2\}$

$\delta(q_0q_1q_2q_3, b)$

$= \delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_2, b) \cup \delta(q_3, b)$
$= \{q_0\} \cup \{q_1\} \cup \{q_3\} \cup \{q_2\}$
$= \{q_0q_1q_2q_3\}$

$\delta(q_0q_1q_3, a) = \delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_3, a)$
$= \{q_0, q_1\} \cup \{q_2\} \cup \phi$
$= \{q_0q_1q_2\}$

$\delta(q_0q_1q_3, b) =$
$\delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_3, b)$
$= \{q_0\} \cup \{q_1\} \cup \{q_2\}$
$= \{q_0q_1q_2\}$



q_2 and q_3 are the final states of NFA. $\therefore$ states containing either q_2 or q_3 or both are final states in DFA

- Convert NFA to DFA

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

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

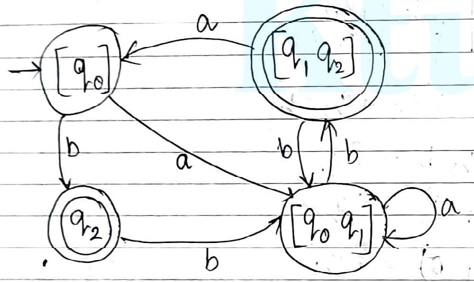
$\delta(q_0q_1, a) = \delta(q_0, a) \cup \delta(q_1, a)$
$= \{q_0q_1\} \cup \{q_0q_1\}$
$= \{q_0q_1\}$
$\delta(q_0q_1, b) = \delta(q_0, b) \cup \delta(q_1, b) =$
$= \{q_2\} \cup \{q_1\} = \{q_1q_2\}$

δ/Σ	a	b
$\rightarrow q_0$	$\{q_0q_1\}$	$\{q_2\}$
$\{q_0, q_1\}$	$\{q_0q_1\}$	$\{q_1q_2\}$
$*\{q_2\}$	ϕ	$\{q_0q_1\}$
$*\{q_1, q_2\}$	$\{q_0\}$	$\{q_0q_1\}$

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

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

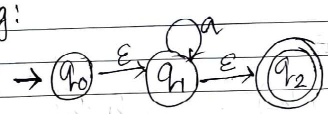
$\delta(q_1 q_2, b) = \delta(q_1, b) \cup \delta(q_2, b)$
$= \{q_1\} \cup \{q_0 q_1\}$
$= \{q_0 q_1\}$



ε-NFA or NFA with ε Transitions

- an NFA which allows transition on ε
- Formally defined as 5 tuples
$(Q, \Sigma, \delta, q_0, F)$
all are the tuples are defined in the same way as in NFA except δ.
$\delta : Q \times \Sigma \cup \{\varepsilon\} \rightarrow 2^Q$

for eg:



ε-NFA = NFA

Conversion of ε. NFA to NFA

ε-closure (q)

ε-closure(q) is defined as set of all states of the automata (ENFA) which can be reach from q on a path labelled by ε, ie without consuming any input symbol. of any state will be non-empty.

Q. Convert ENFA to an NFA

ε-closure (q)

$\varepsilon.\text{CLOSE}(q_0) = \{q_0, q_1, q_2\}$ (1)
$\varepsilon.\text{CLOSE}(q_1) = \{q_1, q_2\}$ (2)
$\varepsilon.\text{CLOSE}(q_2) = \{q_2\}$ (3)

$\delta(\{q_0 q_1 q_2\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0)$
$= \{q_0\} \cup \phi \cup \{q_2\}$
$= \{q_0, q_2\}$

$\delta(\{q_0 q_1 q_2\}, 1) = \delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1)$
$= \phi \cup \{q_1\} \cup \phi$
$= \{q_1\}$

Now ε close

ε close $(q_0 q_2) = \{q_0 q_1 q_2\}$
ε close $(q_1) = \{q_1 q_2\}$

$\delta(\{q_1 q_2\}, 0) = \delta(q_1, 0) \cup \delta(q_2, 0)$
$= \phi \cup \{q_2\} = \{q_2\}$

$\delta(\{q_1 q_2\}, 1) = \{q_1\} \cup \phi = \{q_1\}$

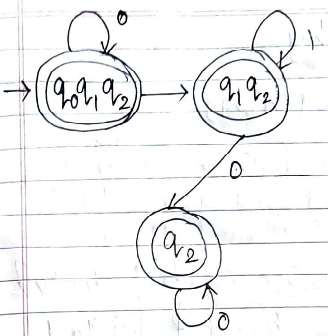
ε close $(q_2) = \{q_2\}$

ε close $(q_1) = \{q_1 q_2\}$

$\delta(\{q_2\}, 0) = \delta(q_2, 0) = \{q_2\}$

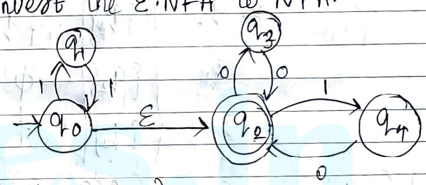
$\delta(\{q_2\}, 1) = \delta(q_2, 1) = \phi$

ε-CLOSE $(q_2) = \{q_2\}$
ε-CLOSE $(\phi) = \phi$



	0	1
→* $\{q_0 q_1 q_2\}$	$\{q_0 q_1 q_2\}$	$\{q_1 q_2\}$
* $\{q_1 q_2\}$	$\{q_2\}$	$\{q_1 q_2\}$
* $\{q_2\}$	$\{q_2\}$	$\{\phi\}$

- Convert the ε.NFA to NFA.



ε. Closure (q)

ε. CLOSE $(q_0) = \{q_0 q_2\}$
ε. CLOSE $(q_1) = \{q_1\}$
ε. CLOSE $(q_2) = \{q_2\}$
ε. CLOSE $(q_3) = \{q_3\}$
ε. CLOSE $(q_4) = \{q_4\}$

$\delta(\{q_0 q_2\}, 0) = \delta(q_0, 0) \cup \delta(q_2, 0)$
$= \phi \cup \{q_3\}$
$= \{q_3\}$

$\delta(\{q_0 q_2\}, 1) = \delta(q_0, 1) \cup \delta(q_2, 1)$
$= \{q_1\} \cup \{q_4\}$
$= \{q_1, q_4\}$

q_0q_2 | $\{q_1, q_4\}$

$\varepsilon.CLOSE(q_3) = \{q_3\}$

$\varepsilon.CLOSE(q_1, q_4) = \{q_1, q_4\}$

not a new state

	0	1
$\rightarrow * [q_0 q_2]$	$\{q_3\}$	$\{q_1, q_4\}$
q_1	ϕ	$\{q_0 q_2\}$
$* q_2$	$\{q_3\}$	$\{q_4\}$
q_3	$\{q_2\}$	ϕ
q_4	$\{q_2\}$	ϕ

$\varepsilon.CLOSE(\delta(\{q_1\}, 0)) = \varepsilon.CLOSE(\delta(q_1, 0))$

$= \varepsilon.CLOSE(\phi) = \underline{\phi}$

$\varepsilon.CLOSE(\delta(\{q_1\}, 1)) = \varepsilon.CLOSE(\delta(q_1, 1))$
$= \varepsilon.CLOSE(q_0) = \{q_0 q_2\}$

$\varepsilon.CLOSE(\delta(\{q_2\}, 0)) = \varepsilon.CLOSE(\delta(q_2, 0))$

$\varepsilon.CLOSE(q_3) = \{q_3\}$
$\varepsilon.CLOSE(\delta(\{q_2\}, 1)) = \varepsilon.CLOSE(\delta(q_2, 1))$
$= \varepsilon.CLOSE(q_4) = \{q_4\}$

$\varepsilon.CLOSE(\delta(\{q_3\}, 0)) = \varepsilon.CLOSE(\delta(q_3, 0))$

$= \varepsilon.CLOSE(q_2) = \{q_2\}$

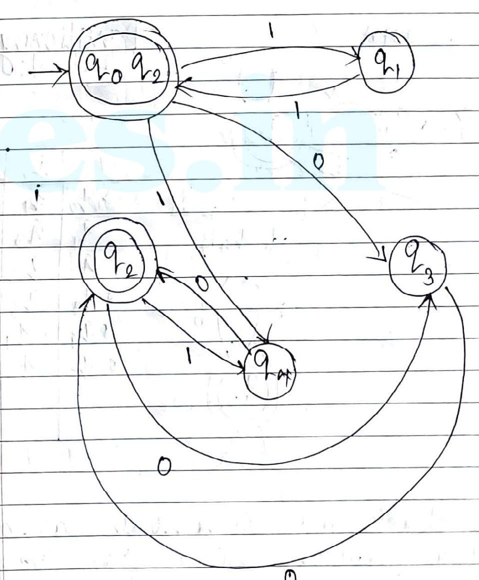
$\varepsilon.CLOSE(\delta(\{q_3\}, 1)) = \varepsilon.CLOSE(\delta(q_3, 1))$
$= \varepsilon.CLOSE(\phi) = \phi$

$\varepsilon.CLOSE(\delta(\{q_4\}, 0)) = \varepsilon.CLOSE(\delta(q_4, 0))$

$= \varepsilon.CLOSE(q_2) = \{q_2\}$

$\varepsilon.CLOSE(\delta\{q_4\}, 1) = \varepsilon.CLOSE(\delta(q_4, 1))$

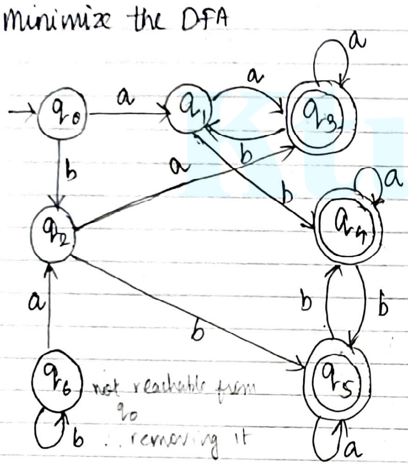
$= \varepsilon.(\phi) = \phi$



Minimisation of DFA

* To get the minimal FA
* 2 step process
  (i) Remove unreachable states
  (ii) Identify Indistinguishable / equivalent states and merge them.
* Can be acheived by 2 methodologies
  (i) Partitioning method
  (ii) Table filling method.

Q. Minimize the DFA



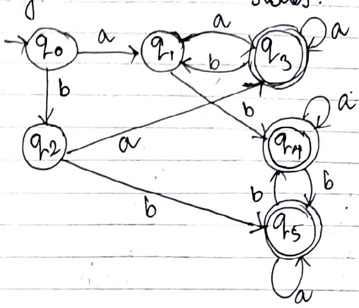
Partitioning method:

	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_3	q_4
q_2	q_3	q_5
$* q_3$	q_3	q_1
$* q_4$	q_4	q_5
$* q_5$	q_5	q_4

An unreachable state is a state that cannot be reached from the initial state on any transitions.

π_{i-1}, then π_i will be the reference. Like wise in π_1, π_0 will be the reference.

Removing unreachable states:



$\delta(q_1, a) = q_3$
$\delta(q_2, a) = q_3$

$\delta(q_1, b) = q_4$
$\delta(q_2, b) = q_5$

Equivalent classes:

0-Equivalent

$\pi_0 = \{F\} \ \underbrace{\{Q-F\}}_{}$ Non-final state

$\pi_0 = \{q_0 q_1 q_2\} \{q_3 q_4 q_5\}$

$\pi_1 = \{q_0\} \quad \{q_1 q_2\}$

$\{q_3\} \quad \{q_4 q_5\}$

$\delta(q_0, b) = q_2$
$\delta(q_1, b) = q_4$

$\delta(q_0, a) = q_1$
$\delta(q_1, a) = q_3$
$\therefore q_0$ & q_1 are not equivalent so in π_1 they write in separate sets

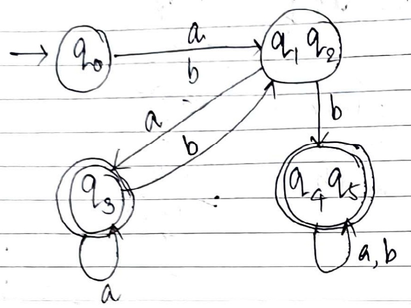
$\delta(q_3, a) = q_3$	$\delta(q_3, b) = q_1$
$\delta(q_4, a) = q_4$	$\delta(q_4, b) = q_5$

$\delta(q_0, a) = q_1$	$\delta(q_0, b) = q_2$
$\delta(q_2, a) = q_3$	$\delta(q_2, b) = q_5$

$\delta(q_3, a) = q_3$	$\delta(q_3, b) = q_1$	$\delta(q_1, a) = q_3$	$\delta(q_1, b) = q_4$
$\delta(q_5, a) = q_5$	$\delta(q_5, b) = q_4$	$\delta(q_2, a) = q_3$	$\delta(q_2, b) = q_5$

$\delta(q_4, a) = q_4$	$\delta(q_4, b) = q_5$	$\delta(q_4, a) = q_4$	$\delta(q_4, b) = q_5$
$\delta(q_5, a) = q_5$	$\delta(q_5, b) = q_4$	$\delta(q_5, a) = q_5$	$\delta(q_5, b) = q_4$

$\Pi_2 = \{q_0\} \ \{q_1 q_2\} \ \{q_3\} \ \{q_4 q_5\}$



b) Table filling method :

1) To remove all unreachable states
2) Create a q x q table where q is the total number of states of the corresponding DFA

	q_0	q_1	q_2	q_3	q_4	q_5
q_0						
q_1						
q_2						
q_3						
q_4						
q_5						

	q_0	q_1	q_2	q_3	q_4	q_5
q_1	✓					
q_2	✓					
q_3	✓	✓	✓			
q_4	✓	✓	✓	✓		
q_5	✓	✓	✓	✓		

	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_3	q_4
q_2	q_3	q_5
$*q_3$	q_3	q_1
$*q_4$	q_4	q_5
$*q_5$	q_5	q_4

mark final and NF state with a ✓
- marking non equivalent states.

3) Identify all unmark pairs, take one by one and check the transition on all input states.

$\delta(q_0 q_1, a)$
$= \delta(q_0, a) \cup \delta(q_1, a)$
$= \{q_1, q_3\}$

atleast one is non equivalent - mask it

$\delta(q_0 q_1, b) = \delta(q_0, b) \cup \delta(q_1, b)$
$= q_2 \cup q_4$
$= \{q_2, q_4\}$

$\delta(q_1 q_2, a) = \delta(q_1, a) \cup \delta(q_2, a)$
$= q_3 \cup q_3$
$= \{q_3\}$

$\delta(q_1 q_2, b) = \delta(q_1, b) \cup \delta(q_2, b)$
$= \{q_4, q_5\}$

$\delta(q_3 q_4, a) = \delta(q_3, a) \cup \delta(q_4, a)$
$= q_3 \cup q_4$
$= \{q_3, q_4\}$

$\delta(q_3 q_4, b) = \{q_1, q_5\}$

$\delta(q_3 q_5, a) = \delta(q_3, a) \cup \delta(q_5, a)$
$= \{q_3, q_5\}$

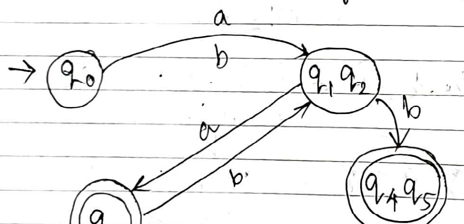
$\delta(q_3 q_5, b) = \delta(q_3, b) \cup \delta(q_5, b)$
$= \{q_1, q_4\}$

$\delta(q_4 q_5, a) = \delta(q_4, a) \cup \delta(q_5, a)$
$= \{q_4, q_5\}$

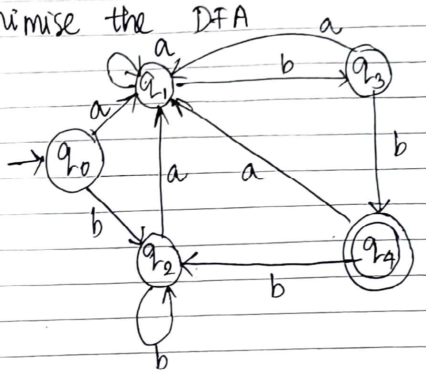
$\delta(q_4 q_5, b) = \delta(q_4, b) \cup \delta(q_5, b)$
$= \{q_5, q_4\}$

4) After one iteration, perform one more iteration, there are new mark, take unmarked pair, else repeat iteration until no new marking happens in the iteration.

5) The unmarked pair are equivalence class and can be merged.



Q. Minimise the DFA



no unreachable states

	a	b
q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	q_4
* q_4	q_1	q_2

Final state $= q_4$

Non final states $= q_0 q_1 q_2 q_3$

$\pi_0 = \{q_0 q_1 q_2 q_3\} \{q_4\}$

$\pi_1 = \{q_0 q_1 q_2\} \{q_3\} \{q_4\}$

$\pi_2 = \{q_0 q_2\} \{q_1\} \{q_3\} \{q_4\}$

$\pi_3 = \{q_0 q_2\} \{q_1\} \{q_3\} \{q_4\}$

$\pi_2 = \pi_3$

So we can stop.

$\delta(q_0, a) = q_1$	$\delta(q_0, b) = q_2$
$\delta(q_1, a) = q_1$	$\delta(q_1, b) = q_3$
$\delta(q_0, a) = q_1$	$\delta(q_0, b) = q_2$
$\delta(q_2, a) = q_1$	$\delta(q_2, b) = q_2$
$\delta(q_0, a) = q_1$	$\delta(q_0, b) = q_2$
$\delta(q_3, a) = q_1$	$\delta(q_3, b) = q_4$
$\delta(q_1, a) = q_1$	$\delta(q_1, b) = q_3$
$\delta(q_2, a) = q_1$	$\delta(q_2, b) = q_2$
$\delta(q_1, a) = q_1$	$\delta(q_1, b) = q_3$
$\delta(q_3, a) = q_1$	$\delta(q_3, b) = q_4$

$M = (Q, \Sigma, \delta, q_0, ...$

Grammar

- Language Generator
- Formally defined as a quadruple (4 tuples) (V, T, P, S)

V- finite non empty set of Non-terminals / variables (written in uppercase)

T- finite, non-empty set of terminals (written in lowercase)

P- set of production rules.

$S \rightarrow$ Start symbol, $S \in V$

Productions are of the form

$$\alpha \rightarrow \beta$$

where

$$\alpha \Rightarrow \{V \cup T\}^* V \{V \cup T\}^*$$

$$\beta \Rightarrow \{V \cup T\}^*$$

(single non terminal must be on the left hand side)

$A \rightarrow a$ ✓

$a \rightarrow A$ ✗

↳ terminal

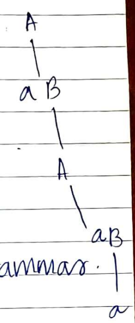
Regular Grammar / Type 3 Grammar

A grammar is said to be regular, if it is either right linear or left linear. Regular grammar generates regular language. Formally a regular grammar is defined by a Quadruple (V, T, P, S) where V is finite non empty set of non-terminals, T is finite non empty set of terminals, P-set of production rules and S is the start symbol.

Production rules are defined as

$\alpha \rightarrow \beta$ and productions are of the form:

$A \rightarrow aB$		$A \rightarrow Ba$
$A \rightarrow a$	or	$A \rightarrow a$
$A \rightarrow \varepsilon$		$A \rightarrow \varepsilon$
(Right Linear)		(Left Linear)



Where $A \in V$

$a \in T$

$S \rightarrow aS$

$S \rightarrow \varepsilon$

$S \rightarrow aS / \varepsilon$

Derivation of string aaaa

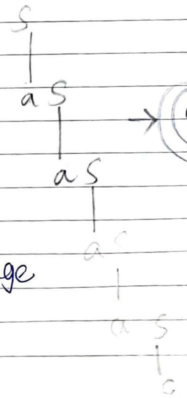
$S \rightarrow aS$

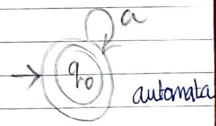
$S \rightarrow aaaaS$

$S \rightarrow aaaa.\varepsilon$

$S \rightarrow aaaa$

$S \rightarrow aS/\varepsilon$ - Grammar.





$\{\varepsilon, a, aa, aaa \dots\}$

$a^0 \; a^1 a^2 \; a^3 \dots a^n$

$L = \{a^n / n \geq 0\}$ Language

ε-NFA $\cong$ NFA $\cong$ DFA

$A \to \varepsilon$: ε production.

Conversion of Regular Grammar to Finite Automata

Step1: Construct an ε-free regular grammar (Grammar without ε productions) G' from G.

Step2: Create an FSA 'M' with state for every non-terminal in G'. Set the state representing the start symbol in G' as the start state.

Step3: Add another new state q_f has final state.

Step4: For every production of the form $A \to aB$ (A & B are single non terminals and a is a terminal), Add a transition from state A to state B on input signal a.

Step5: for every production $A \to a$, you will add a transition from $A \xrightarrow{a} q_f$

Step6: If there is a production $S \to \varepsilon$ in G', where $S \to$ start symbol in G', you will make 'S' a final state.

Removing ε productions / creating ε free grammar

A Regular grammar is said to be ε-free, if it has no ε productions except possibly for the production $S \to \varepsilon$ where 'S' is the start symbol and 'S' doesn't appear on the right hand side of any production.

Step1: Copy all non ε productions from G to G'

Step2: For any non-terminal 'N' which can become ε (ie $N \to \varepsilon$) we copy every rule in which 'N' appears on the RHS both with & without N.

step3: If 'S' gives ε was in the original set of rules, add a new start symbol S_1 in G' and add the production $S_1 \to \varepsilon$ and copy all the production rules of 'S' to the RHS of S_1.

$G = \{\{S,A,B\}\ \{a,b,c\}, P, S\}$

- Construct a FSA for the RG

G

$S \to a \mid aA \mid bB \mid \varepsilon$

$A \to aA \mid aS$

$B \to cS \mid \varepsilon$

$S \to \varepsilon$

$B \to \varepsilon$

1.

② S വരുന്നിടത്ത് ε

① $S \to a \mid aA \mid bB$

$A \to aA \mid aS$

$B \to cS$

$A \to aS$

$A \to a \cdot \varepsilon$

$A \to a$

② $S \to a \mid aA \mid bB$ (no S here)

$A \to aA \mid aS \mid a$

$B \to cS \mid c$

$B \to cS$

$B \to c\varepsilon$

$B \to c$

$A \rightarrow aB$

$A \xrightarrow{a} B$

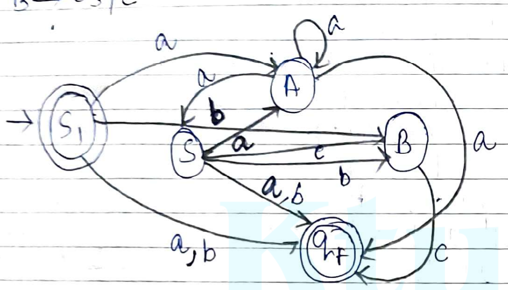
$S \rightarrow a \mid aA \mid bB \mid b$
$A \rightarrow aA \mid aS \mid a$
$B \rightarrow cS \mid c$

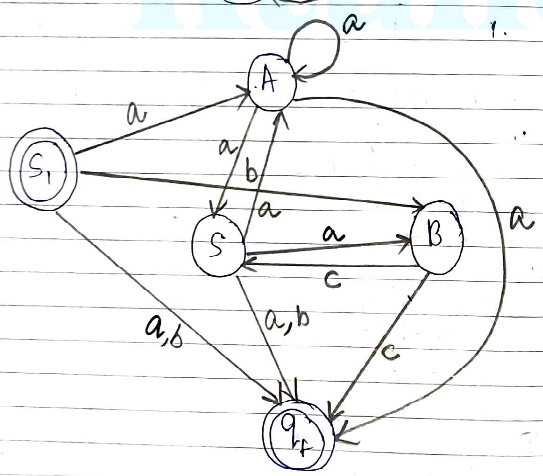
G'

$S \rightarrow bB$
$S \rightarrow b$

③ $S_1 \rightarrow \varepsilon \mid a \mid aA \mid bB \mid b$
$S \rightarrow a \mid aA \mid bB \mid b$
$A \rightarrow aA \mid aS \mid a$
$B \rightarrow cS \mid c$

2.

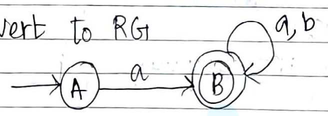




Conversion of FSA to Regular Grammar:

① Begin from start state
② For every transition
$\delta(Q, a) \rightarrow P$
add production
$Q \rightarrow aP$
③ For every $Q \in F$,
add production $Q \rightarrow \varepsilon$

Q. Convert to RG



B - final state

$\delta(A, a) \rightarrow B$
$A \rightarrow aB$

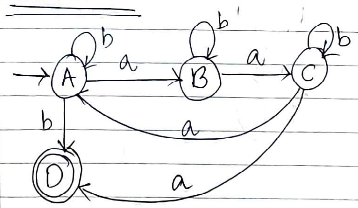
$\delta(B, a) \rightarrow B$ $\quad B \rightarrow aB$
$\delta(B, b) \rightarrow B$ $\quad B \rightarrow bB$
$B \rightarrow \varepsilon$

$A \rightarrow aB$
$B \rightarrow aB$
$B \rightarrow bB$
$B \rightarrow \varepsilon$

or

$A \rightarrow aB$
$B \rightarrow aB \mid bB \mid \varepsilon$

Q.



$\delta(A,a) = B$
$\delta(A,b) = A$

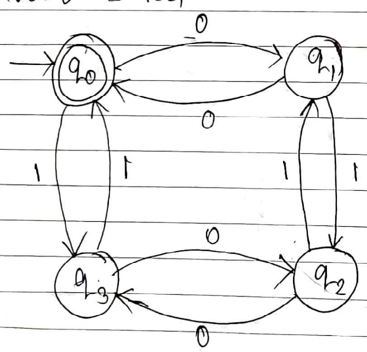
$\delta(B,a) = C$
$\delta(B,b) = B$

$\delta(C,a) = A$
$\delta(C,a) = D$
$\delta(C,b) = C$

$A \to aB$
$A \to bA$
$B \to aC$
$B \to bB$
$C \to aD$
$C \to aA$
$C \to bC$
$D \to \varepsilon$

$A \to aB \mid bA$
$B \to aC \mid bB$
$C \to aD \mid aA \mid bC$
$D \to \varepsilon$

Q. Convert to RG



$\delta(q_0,0) = q_1$ $\quad q_0 \to 0q_1$
$\delta(q_0,1) = q_3$ $\quad q_0 \to 1q_3$

$\delta(q_1,0) = q_0$ $\quad q_1 \to 0q_0$
$\delta(q_1,1) = q_2$ $\quad q_1 \to 1q_2$

$\delta(q_2,0) = q_3$ $\quad q_2 \to 0q_3$
$\delta(q_2,1) = q_1$ $\quad q_2 \to 1q_1$

$\delta(q_3,0) = q_2$ $\quad q_3 \to 0q_2$
$\delta(q_3,1) = q_0$ $\quad q_3 \to 1q_0$

$q_0 \to \varepsilon$

$q_0 \to 0q_1 \mid 1q_3 \mid \varepsilon$
$q_1 \to 0q_0 \mid 1q_2$
$q_2 \to 0q_3 \mid 1q_1$
$q_3 \to 0q_2 \mid 1q_0$