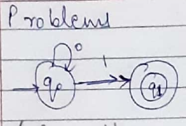
* Problems

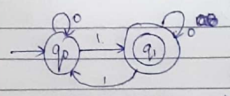
(1) 

$Q = \{q_0, q_1\}$ $\quad \delta(q_0, 0) = q_0$
$\Sigma = \{0, 1\}$ $\quad \delta(q_0, 1) = q_1$

(Transition should be defined for every state. Since q_1 states $\delta(q_1, 0) = ND$
$\delta(q_1, 1) = ND$

Language $L = \{01, 001, 0001, 00001 \ldots\}$ is accepted

~~011~~

(2) 

It's DFA

$L = \{01, 0100, 100, 10, 1011 \ldots\}$
(is accepted)

~~Language is~~ accepted
$\delta(q_0, w) \in P$.
PGF
L accepted

* $\Sigma^* \to$ Considers empty string
$\Sigma \to$ Does not consider empty string
} The difference is presence of epsilon.

Question to be asked

* Pattern Recognition — * ends.
* length — * having a substring
* starts — * divisible by counter 'k'

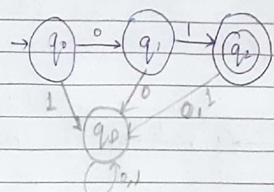
[Eg.] Construct a DFA, over an ~~end~~ input alphabet $\{0, 1\}$ which accepts the string only '01'.

$\Sigma = \{0, 1\}$ $\quad L = \{01\}$

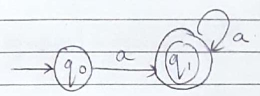
Min string = 2 (n)
No of states = 2+1
(n+1)

To construct:
• Identify L, Σ
• Basic / Initial DFA for min string

$\delta(q_0, 0) = q_1$
$\delta(q_1, 1) = q_2$
$\delta(q_0, 1) = q_D$
$\delta(q_1, \cancel{1}) = q_D$
$\delta(q_2, 0) = q_D$
$\delta(q_2, 1) = q_D$

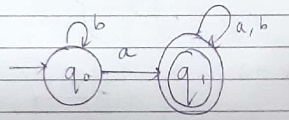


[Eg2] Construct a DFA having atleast one 'a'
$\Sigma = \{a\}$ $\quad L = \{a, aa, aaa, aaaa \ldots\}$ $\quad \epsilon$ - denotes zero number of symbol

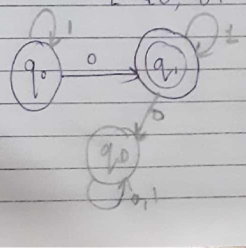


δ	a
q_0	q_1
* q_1	q_1

[Eg3] Construct a DFA $\Sigma = \{a, b\}$ atleast one 'a'.
$L = \{ab, aab, bab, a, abb \ldots\}$



[Eg4] Construct a DFA over an input alphabet $\Sigma = \{0, 1\}$ having exactly one '0'
$L = \{0, 01, 011, 0111, 01111 \ldots\}$

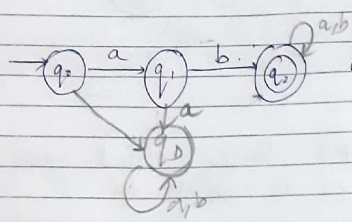


δ	0	1
q_0	q_1	q_0
* q_1	q_D	q_1
q_D	q_D	q_D

Eg 5. Construct a DFA over an input alphabet {a, b} which starts with the string ab.

L = { ab, abb, abab, abbb ... }

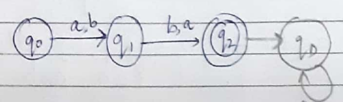
$\delta(q_0, a) = q_1$
$\delta(q_1, b) = q_2$
$\delta(q_1, b)$



Eg 6. Construct a DFA $\Sigma = \{a, b\}$ having length = 2.

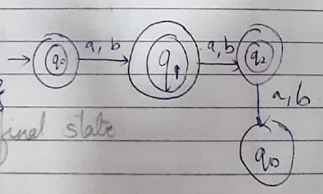
L = {a, b, ba, bb, aa}

δ	a	b
q_0	q_1	q_1
q_1	q_2	q_2
$*q_2$	q_D	q_D



7. Having length ≤ 2

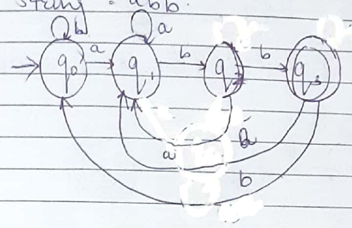
L = { ε, a, b, ba, bb, aa }
↳ Initial state as final state



δ	a	b
$*q_0$	q_1	q_1
$*q_1$	q_2	q_2
$*q_2$	q_D	q_D
q_D	q_D	q_D

8. $\Sigma = \{a, b\}$ which ends with {a, bb}.

L = { aabb, aababb, babb ... }
min string = abb.



	δ	a	b
ε	q_0	q_1	q_0
a	q_1	q_1	q_2
ab	q_2	q_1	q_3
abb*	q_3	q_1	q_0

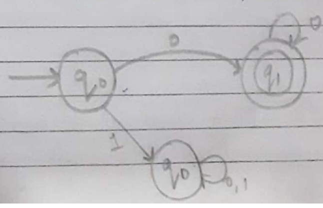
• Power of an alphabet – Σ^K – Set of strings with length K. For eg: $\Sigma = \{0, 1\}$
$\Sigma^1 = \{0, 1\}$
$\Sigma^2 = \{00, 01, 10, 11\}$
$\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

• $\Sigma^* = \Sigma^0 \cup \Sigma^+$
$= \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$

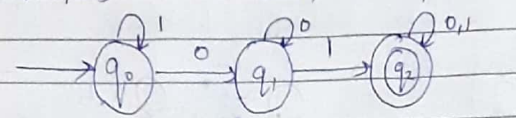
→ $\Sigma^0 = \epsilon$, ∴ Σ^* contains ε (epsilon – empty string or zero ")

9. Construct a DFA over input alphabet {0, 1} which starts with string 0.

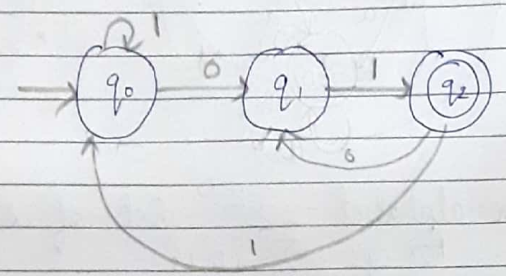
→ Minimum string acceptable by the DFA = 0.
L = { 0, 01, 00, 000, 0011, ... }



10. a) Construct / Design a DFA that has substring 01,
$\Sigma = \{0,1\}$ $L = \{01, 1001, 00001, 100001, \ldots\}$

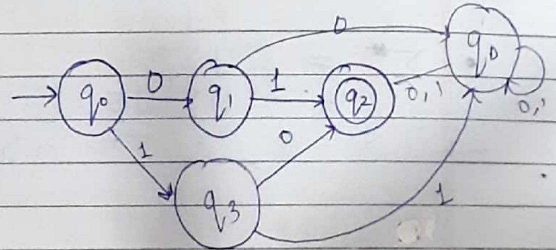


b) Ends with '01' $L = \{01, 0001, 11101, 0101, \ldots\}$



11. Construct a DFA which read strings only '01' or '10' over Input $\Sigma = \{0,1\}$.

Language = $\{01, 10\}$
States = $2+1 = 3$

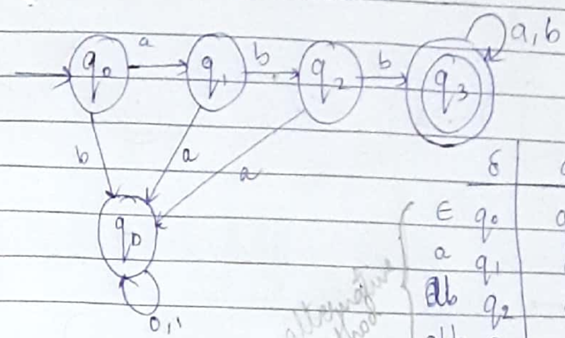


Transition table.

δ	0	1
q_0	q_1	q_3
q_1	q_D	q_2
$*q_2$	q_D	q_D
q_3	q_1	q_D
q_D	q_D	q_D

12. Construct a DFA which accepts strings over $\{a,b\}$ such that all strings should start with abb.

$L = \{abb, abba, abab, abbaa, \ldots\}$
States = $3+1 = 4$



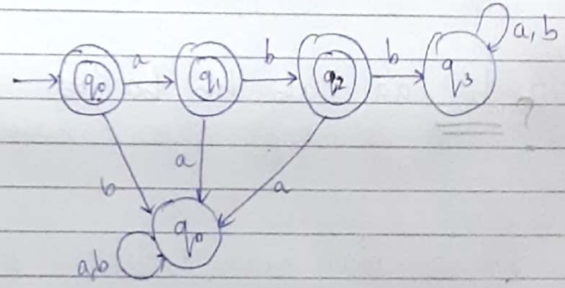
alternative method

	δ	a	b
ε	q_0	q_1 (ε a→a)	q_D (ε b→b)
a	q_1	q_D (aa)	q_2 (ab)
ab	q_2	q_D (aba)	q_3 (abb)
abb	q_3	q_3 (abba)	q_3 (abbb)
	q_D	q_D	q_D

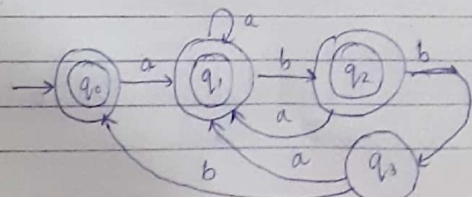
13. CDFA over input, $\Sigma = \{a,b\}$ such that all string should 'not' start with abb.

→ When 'not' is encountered (referring to above problem)
Soln:— 1. Construct DFA
2. Change final state to non-final state
3. Change non-final state to final state

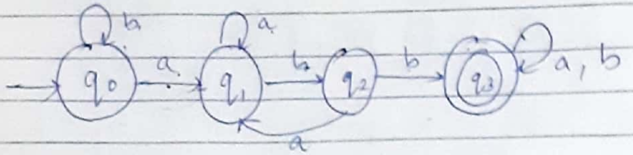
Confusing



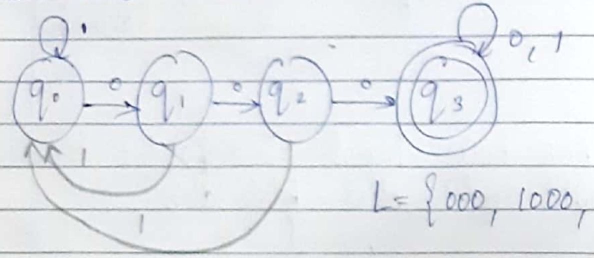
14. CDFA over $\Sigma = \{a,b\}$ that does not ends with abb



15 CDFA $\Sigma = \{a, b\}$ having a substring abb.
language = {abb, aabbb, babba, bbabba ...}



16. CDFA over input alphabets $\Sigma = \{0,1\}$ having consecutive zeros '000'.



L = {000, 1000, 10001, 11000}

W = length of string

17 Construct DFA over $\Sigma = \{a\}$ such that $|W| \bmod 3 = 0$

18. having a substring 100, $\Sigma = \{0,1\}$

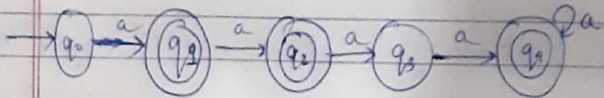
19 Ends with aba {a,b}.

20 Begins with 101 {0,1}.

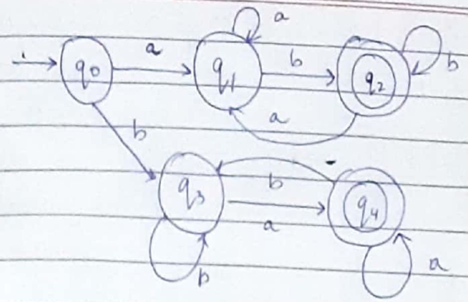
21 Construct a DFA to accept the language, $L = \{a^n / n \geq 1, n \neq 3\}$

$\Sigma = \{a\}$

L = {a, aa, aaa, aaaa, aaaaa ...}

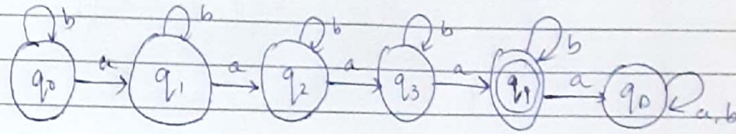


23 CDFA over ~~an~~ input alphabet $\Sigma = \{a,b\}$ starts and ends with different symbol



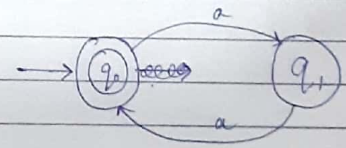
24 CDFA over an input alphabet $\Sigma = \{a,b\}$ which accepts 4 a's

L = { aaaa, aaaab, aaaba ... }

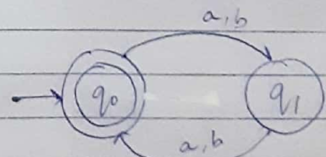


25 a) CDFA over an input alphabet $\Sigma = \{a\}$, which accepts even number of a's / $|W| \bmod 2 = 0$ {a} / divisible by 2

→ 0,1 → (2) → 2 states

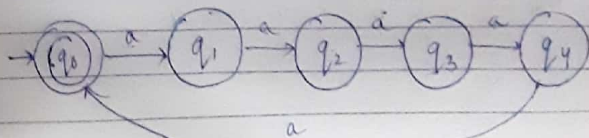


b) $\Sigma = \{a,b\}$, even number of a's and b's.



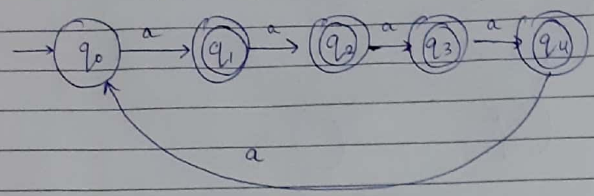
26 $|W| \bmod 5 = 0$ $\Sigma = \{a\}$, 5 5 states

0, 1, 2, 3, 4



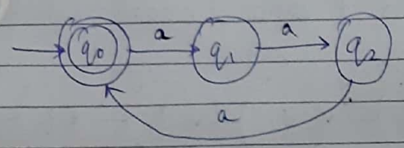
27. Construct a DFA $L = \{ |W| \bmod 5 \neq 0,\ W \in \{a\}^* \}$

⇒ Remainders = 0, 1, 2, 3, 4 and it can accept all excluding '0', rest all are made as final state.



17.

$|W| \bmod 3 = 0$.

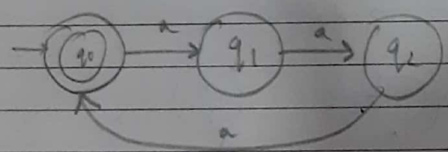


28. $|W| \bmod 3 > |W| \bmod 2$. $\Sigma = \{a\}$

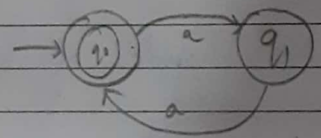
1st Machine: 0, 1, 2

2nd Machine: 0, 1

1st M_1



2nd M_2



Take cross product of both the machines

$\{(0,0), (0,1), (1,0), (1,1), (2,0), (2,1)\}$ — M_1, M_2

$\delta(q_0(0,0), a) = (1,1) \rightarrow q_1$ — M_1, M_2

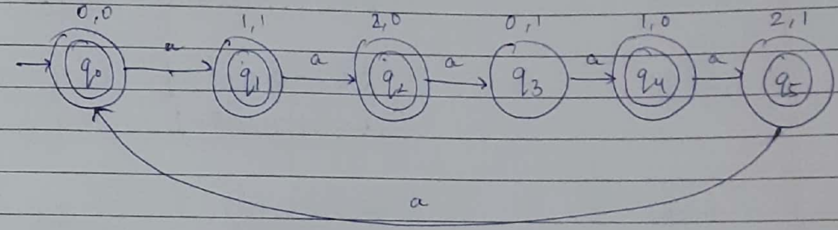
$\delta(q_0(1,1), a) = (2,0) \rightarrow q_2$

$\delta(q_0(2,0), a) = (0,1) \rightarrow q_3$

$\delta(q_0(0,1), a) = (1,0) \rightarrow q_4$

$\delta(q_0(1,0), a) = (2,1) \rightarrow q_5$

$\delta(q_0(2,1), a) = (0,0) \rightarrow q_0$



29. Construct a DFA over input alphabet $\{0,1\}$ where ~~which~~ the strings are binary number divisible by 5

Divisibility – $\delta(q_i, a) = q_j$

$j = (r * i + d) \bmod k$

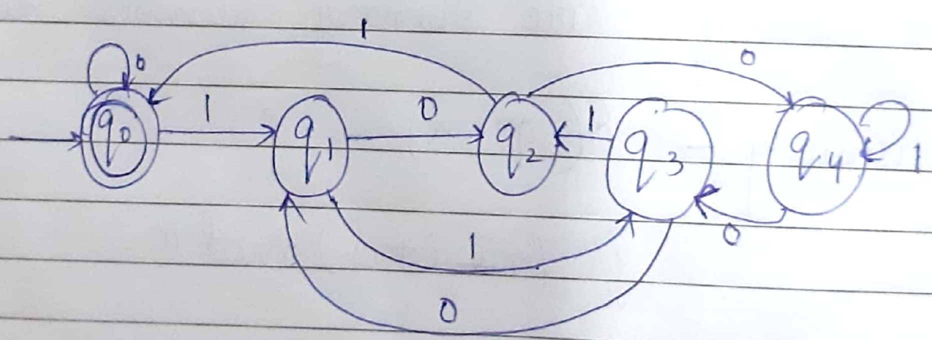
$r \rightarrow$ input — binary 2, decimal 10

$i \rightarrow$ remainder — k - divisor

$d \rightarrow$ digits

$r = 2$ $i = 0, 1, 2, 3, 4$ $k = 5$ $d = \{0, 1\}$

remainder	digits (d)	(r * i + d) mod k = j	$\delta(q_i, d) = q_j$
i = 0	0	(2*0+0) mod 5 = 0	$\delta(q_0, 0) = q_0$
	1	(2*0+1) mod 5 = 1	$\delta(q_0, 1) = q_1$
i = 1	0	(2*1+0) mod 5 = 2	$\delta(q_1, 0) = q_2$
	1	(2*1+1) mod 5 = 3	$\delta(q_1, 1) = q_3$
i = 2	0	(2*2+0) mod 5 = 4	$\delta(q_2, 0) = q_4$
	1	(2*2+1) mod 5 = 0	$\delta(q_2, 1) = q_0$
i = 3	0	(2*3+0) mod 5 = 1	$\delta(q_3, 0) = q_1$
	1	(2*3+1) mod 5 = 2	$\delta(q_3, 1) = q_2$
i = 4	0	(2*4+0) mod 5 = 3	$\delta(q_4, 0) = q_3$
	1	(2*4+1) mod 5 = 4	$\delta(q_4, 1) = q_4$



30. Construct a DFA which is interprated as binary {0,1} divisible by 4

r = 2 i = 0, 1, 2, 3 k = 4 d = {0, 1}

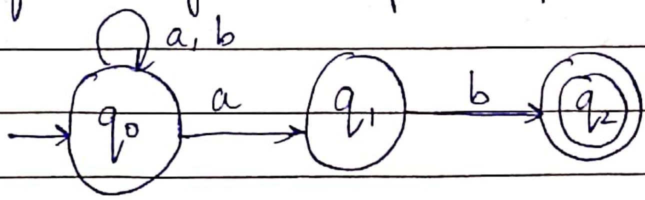
31. decimal divisible by 3.

$r = 10$ $i = 0, 1, 2$ $d = \{0,1,2,3,4,5,6,7,8,9\}$ $K = 3$.

remainder digits.

NFA – More number of transition function (or states) for a given input alphabet.

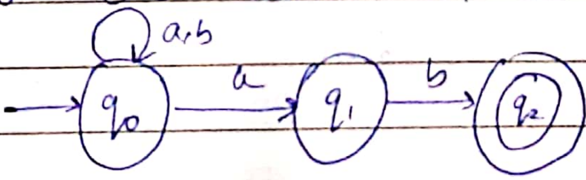
for eg:-



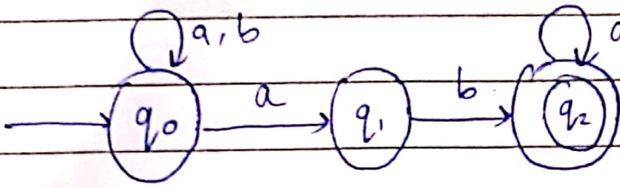
Note:- We don't have to find transition state for every input alphabet over a particular state.

Examples.

1. Construct a NFA with input alphabet $\Sigma = \{a, b\}$ that ends with a, b $L = \{aab, ab, bab, bbab, aaab \ldots\}$.

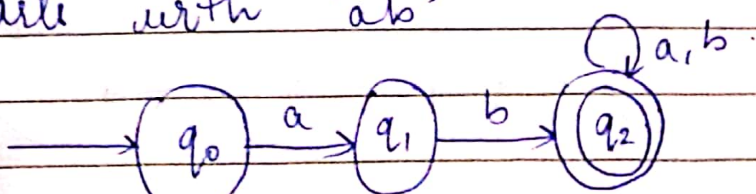


2. CNFA has substring 'ab'.



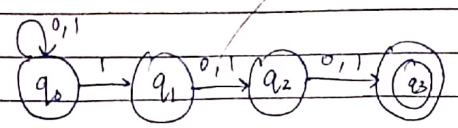
δ	a	b
q_0	$\{q_0, q_1\}$	q_0
q_1	ϕ	q_2
q_2	q_2	q_2

3. CNFA starts with 'ab'

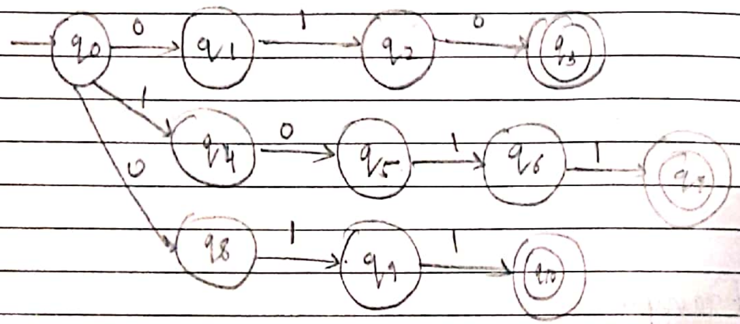


δ	a	b
q_0	q_1	ϕ
q_1	ϕ	q_2
$*q_2$	q_2	q_2

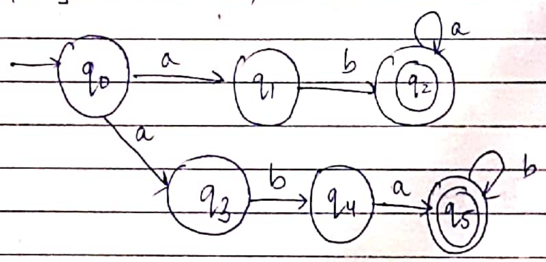
4. CNFA over $\{0,1\}$ such that 3^{rd} symbol from right end should be 1.



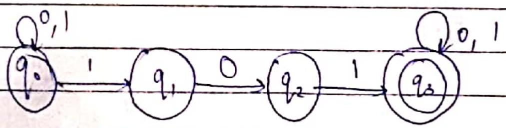
5. CNFA over $\{0,1\}$ which accepts $\{0101, 1011, 011\}$



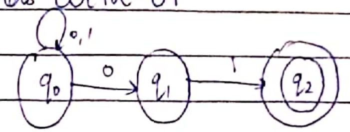
6. $\{a,b\}$ aba^n, $abab^n$.



7. Having substring 101.



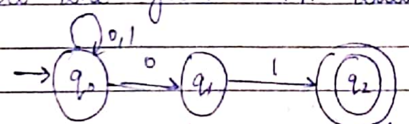
8. Ends with 01



Conversion of NFA to DFA.

1. Subset construction method
2. Lazy evaluation method.

* Convert the given NFA into its equivalent DFA using subset construction method.



1st Step: In the given NFA, the start state and input alphabet for NFA ~~are~~ remains same in DFA.

2nd step NFA → n states
DFA → 2^n states

3rd step Power set. $\{\phi, q_0, q_1, q_2, \{q_0, q_1\}, \{q_1, q_2\}, \{q_0, q_2\}, \{q_0, q_1, q_2\}\}$.

4th step → The final state of NFA if contained in the subset of the power set, then that particular subset becomes the final state in DFA.

5th step → $\delta(\{q_0, q_1\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0)$
(If the set contains more states)
$= \{q_0, q_1\} \cup \phi$
$= \{q_0, q_1\}$

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

1. q_0 is start state in DFA and $\Sigma = \{0,1\}$.

2. $\{\phi, q_0, q_1, q_2, \{q_0, q_1\}, \{q_1, q_2\}, \{q_0, q_2\}, \{q_0, q_1, q_2\}\}$

3. Transition states :-

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

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

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

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

$= \{q_0, q_1\}$

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

$= q_0 \cup q_2$

$= \{q_0, q_2\}$

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

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

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

$= q_0 \cup \phi = q_0$

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

$= \phi \cup \phi = \phi$

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

$= q_2 \cup \phi = q_2$

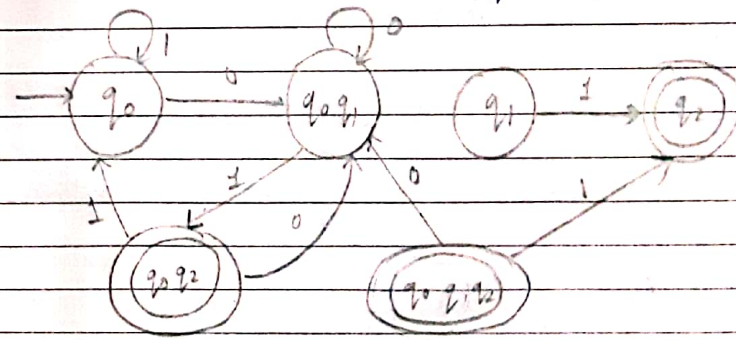
$\delta(\{q_0, q_1, q_2\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0)\ \delta(q_2, 0)$

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

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

$= q_0 \cup \phi \cup q_2$

$= \{q_0, q_2\}$



Lazy Evaluation Method

→ Transition states :-

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

$\delta(q_0, 1) = q_0$

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

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

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

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

classmate

Date ______
Page ______