DOMS Page No.
Date / /

NFA with epsilon transition.

We extend the class of NFAs by allowing instantaneous ε transitions-

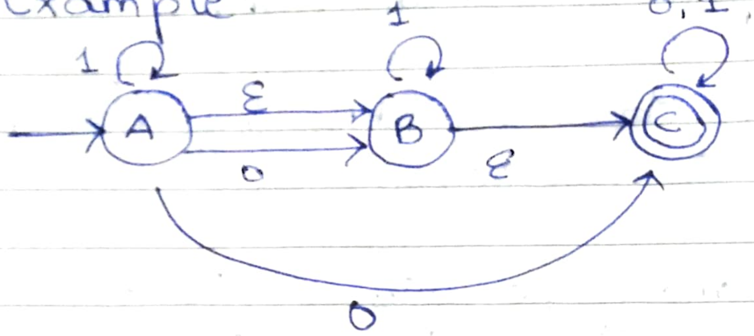
The automaton may be allowed to change its state without reading the input symbol.
In diagrams, such trasitions are depicted by labeling the appropriate arcs with ε.
Note that this does not mean that ε has become an input symbol. On the contrary, we assume that the symbol ε does not belong to any alphabet.

- ε-NFAs add a convenient feature but they bring us nothing new. They donot extend the class of languages that can be represented.
- Both NFAs and ε-NFAs recognize exactly the same languages.

→ Epsilon (ε)- closure
Epsilon closure for a given state X is a set of states which can be reached from the states X with only (null) or ε moves including the state X itself.

In other words, ε-closure for a state can be obtained by union operation of the ε-closure of the states which can be reached from X with a single ε move in a recursive manner.

* Example:

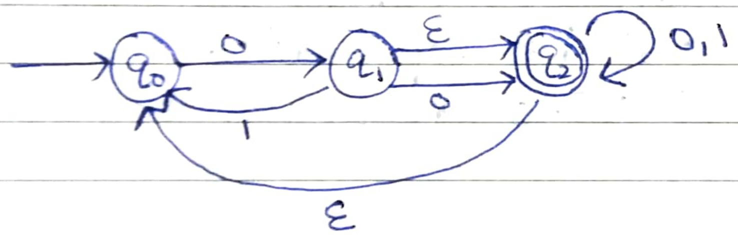
1 

For given eg: ε closure are as follows

ε - closure (A) = {A, B, C}

ε - closure (B) = {B, C}

ε - closure (C) = {C}

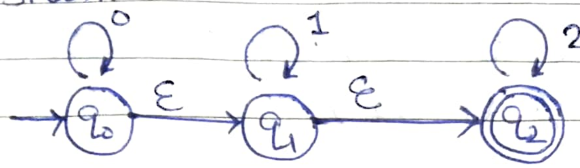
2. 

ε - closure (q_0) = {q_0}

ε - closure (q_1) = {q_1, q_2, q_0}

ε - closure (q_2) = {q_0, q_2}.

* Convert the given NFA with epsilon to NFA without epsilon.



$M = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{0, 1, 2\}$
$q_0 = q_0$
$F = q_2.$

Step 1: $\varepsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$
$\varepsilon\text{-closure}(q_1) = \{q_1, q_2\}$
$\varepsilon\text{-closure}(q_2) = \{q_2\}$

Step 2: Now we will obtain $\delta 1$ transitions for each state on each input symbol.

$\delta'(q_0, 0) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0, \varepsilon), 0)).$
$= \varepsilon\text{-closure}(\delta(\varepsilon\text{-closure}(q_0), 0))$
$= \varepsilon\text{-closure}(\delta(q_0, q_1, q_2), 0))$
$= \varepsilon\text{-closure}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0))$
$= \varepsilon\text{-closure}(q_0 \cup \phi \cup \phi)$
$= \varepsilon\text{-closure}(q_0)$
$= \{q_0, q_1, q_2\}$

$\delta'(q_0, 1) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0, \varepsilon), 1))$
$= \varepsilon\text{-closure}(\delta(q_0, q_1, q_2), 1))$
$= \varepsilon\text{-closure}(\delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1))$
$= \varepsilon\text{-closure}(\phi \cup q_1 \cup \phi)$
$= \varepsilon\text{-closure}(q_1)$
$= \{q_1, q_2\}$

$\delta'(q_0, 2) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_0, \varepsilon), 2))$
$= \varepsilon\text{-closure}(\delta(q_0, q_1, q_2), 2))$
$= \varepsilon\text{-closure}(\delta(q_0, 2) \cup \delta(q_1, 2) \cup \delta(q_2, 2))$
$= \varepsilon\text{-closure}(\phi \cup \phi \cup q_2)$
$= \varepsilon\text{-closure}(q_2)$
$= \{q_2\}$

$\delta'(q_0, 0) = \{q_0, q_1, q_2\}$
$\delta'(q_0, 1) = \{q_1, q_2\}$
$\delta'(q_0, 2) = \{q_2\}$

$\delta'(q_1, 0) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_1, \varepsilon), 0))$
$= \varepsilon\text{-closure}(\delta(q_1, q_2), 0))$
$= \varepsilon\text{-closure}(\delta(q_1, 0) \cup \delta(q_2, 0))$
$= \varepsilon\text{-closure}(\phi \cup \phi)$
$= \varepsilon\text{-closure}(\phi)$
$= \phi$

$\delta'(q_1, 1) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_1, \varepsilon), 1))$
$= \varepsilon\text{-closure}(\delta(q_1, q_2), 1))$
$= \varepsilon\text{-closure}(\delta(q_1, 1) \cup \delta(q_2, 1))$
$= \varepsilon\text{-closure}(q_1 \cup \phi)$
$= \varepsilon\text{-closure}(q_1) = \{q_1, q_2\}$

$\delta'(q_1, 2) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_1, \varepsilon), 2))$
$= \varepsilon\text{-closure}(\delta(q_1, q_2), 2)$
$= \varepsilon\text{-closure}(\delta(q_1, 2) \cup \delta(q_2, 2))$
$= \varepsilon\text{-closure}(\phi \cup q_2)$
$= \{q_2\}$

Similarlity

$\delta'(q_2, 0) = \phi$
$\delta'(q_2, 1) = \phi$
$\delta'(q_2, 2) = \{q_2\}$

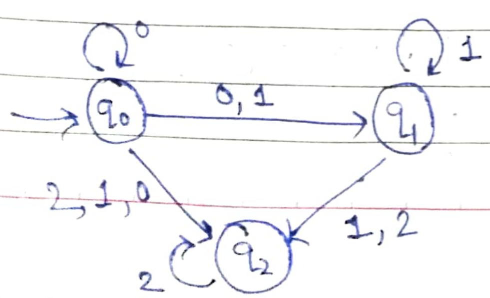
Step 3:
Now we will summarize all transitions
$\delta'(q_0, 0) = \{q_0, q_1, q_2\}$
$\delta'(q_0, 1) = \{q_1, q_2\}$
$\delta'(q_0, 2) = \{q_2\}$

$\delta'(q_1, 0) = \{\phi\}$
$\delta'(q_1, 1) = \{q_1, q_2\}$
$\delta'(q_1, 2) = \{q_2\}$

$\delta'(q_2, 0) = \{\phi\}$
$\delta'(q_2, 1) = \{\phi\}$
$\delta'(q_2, 2) = \{q_2\}$

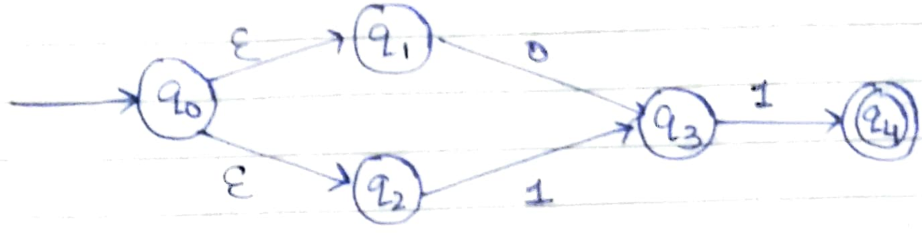
Step 4:
The transition table for NFA without epsilon:

State \ Σ	0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	ϕ	$\{q_1, q_2\}$	$\{q_2\}$
q_2	ϕ	ϕ	$\{q_2\}$



NFA diagram without epsilon.

2).



ε - closure $(q_0) = \{q_0, q_1, q_2\}$
ε - closure $(q_1) = \{q_1\}$
ε - closure $(q_2) = \{q_2\}$
ε - closure $(q_3) = \{q_3\}$
ε - closure $(q_4) = \{q_4\}$.

$\delta'(q_0, 0) = \varepsilon$ - closure $(\delta(\hat{\delta}(q_0, \varepsilon), 0))$

$= \varepsilon$ - closure $(\delta(q_0, q_1, q_2), 0)$
$= \varepsilon$ - closure $(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0))$
$= \varepsilon$ - closure $(\phi \cup q_3 \cup \phi)$
$= \varepsilon$ - closure (q_3)
$= \{q_3\}$

$\delta'(q_0, 1) = \varepsilon$ - closure $(\delta(\hat{\delta}(q_0, \varepsilon), 1))$
$= \varepsilon$ - closure $(\delta(q_0, q_1, q_2), 1)$
$= \varepsilon$ - closure $(\delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1))$
$= \varepsilon$ - closure $(\phi \cup \phi \cup q_3)$
$= \{q_3\}$

$\delta'(q_0, 2) = \varepsilon$ - closure $(\delta(\hat{\delta}(q_0, \varepsilon), 2))$
$= \varepsilon$ - closure $(\delta(q_0, 2) \cup \delta(q_1, 2), \cup \delta(q_2, 2))$
$= \varepsilon$ - closure $(\phi \cup$

$\delta'(q_1, 0) = \varepsilon$ - closure $(\delta(\hat{\delta}(q_1, \varepsilon), 0))$
$= \varepsilon$ - closure $(\delta(q_1), 0)$
$= \varepsilon$ - closure $(\delta(q_1, 0)) = \{q_3\}$

DOMS Page No.
Date / /

$\delta'(q_1, 1) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_1, \varepsilon), 1))$
$= \varepsilon\text{-closure}(\delta(q_1, 1)) = \{\phi\}$

$\delta'(q_2, 0) = \varepsilon\text{-closure}(\delta(\hat{\delta}(q_2, \varepsilon), 0))$
$= \varepsilon\text{-closure}(\delta(q_2), 0)$
$= \varepsilon\text{-closure}(\delta(q_2, 0)) = \phi$

$\delta'(q_2, 1) = \varepsilon\text{-closure}(\delta(q_2, 1)) = \{q_3\}$

$\delta'(q_3, 0) = \varepsilon\text{-closure}(\delta(q_3, 0)) = \phi$

$\delta'(q_3; 1) = \varepsilon\text{-closure}(\delta(q_3, 1)) = \{q_4\}$

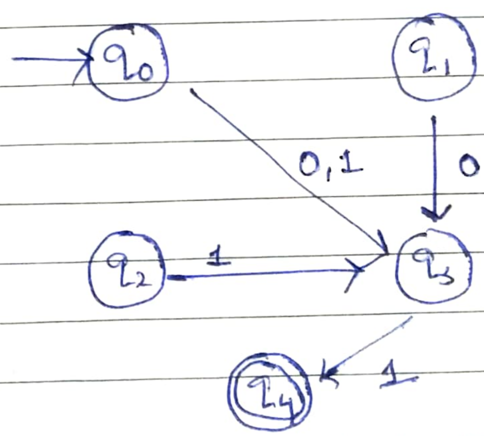
$\delta'(q_4, 0) = \varepsilon\text{-closure}(\delta(q_4, 0)) = \phi$
$\delta'(q_4, 1) = \varepsilon\text{-closure}(\delta(q_4, 1)) = \phi$

$\delta'(q_0, 0) = \{q_3\}$ $\delta'(q_1, 0) = \{q_3\}$ $\delta'(q_2, 0) = \{\phi\}$
$\delta'(q_0, 1) = \{q_3\}$ $\delta'(q_1, 1) = \{\phi\}$ $\delta'(q_2, 1) = \{q_3\}$

$\delta'(q_3, 0) = \{\phi\}$ $\delta'(q_4, 0) = \{\phi\}$
$\delta'(q_3, 1) = \{q_4\}$ $\delta'(q_4, 1) = \{\phi\}$

Transition table

State \ Σ	0	1
q_0	$\{q_3\}$	$\{q_3\}$
q_1	$\{q_3\}$	$\{\phi\}$
q_2	$\{\phi\}$	$\{q_3\}$
q_3	$\{\phi\}$	$\{q_4\}$
q_4	$\{\phi\}$	$\{\phi\}$



Nullable variabe → Directly producing ε

Page No.
Date / /

* Eliminating ε - productions.

Given $G = (V, T, P, S)$

Algorithm:

1. Detect all nullable variables in G
2. Then construct $G_1 = (V, T, P_1, S)$ as follows
   i. For each production of the form: $A \to X_1 X_2 \ldots X_k$ where $k \geq 1$, suppose m out of the k x_i's are nullable symbols.
   ii Then G_1 will be have 2^m versions for this production.
      i.e. all combinations where each X_i is either present or absent.
   iii Alternatively, if a production is of the form: $A \to \varepsilon$, then remove it.

DOMS Page No.
Date / /

eg. 1) $S \to AB$
$A \to aA \mid \epsilon$
$B \to bB \mid A$

Step 1:
$A \to \epsilon$
$B \to A$ $(\because A \to \epsilon)$.
$B \to \epsilon$

S, A, B are nullable variable / NT.

Step 2:
$S \to AB$
$S \to \epsilon . B$ --- $(A \to \epsilon)$
$S \to B$

$S \to AB$
$\to A \cdot \epsilon$ .. $(B \to \epsilon)$
$S \to A$

$S \to AB$
$S \to \epsilon \cdot \epsilon$
$S \to \epsilon$

$A \to aA$	$B \to bB$
$A \to a\epsilon$ .. $(A \to \epsilon)$	$\to b\epsilon$ $(B \to \epsilon)$
$A \to a$	$B \to b$

$S \to AB \mid A \mid B$
$A \to aA \mid a$
$B \to bB \mid b \mid A$

DOMS Page No.
Date / /

2) . $S \rightarrow XYX$
$X \rightarrow 0X \mid \varepsilon$
$Y \rightarrow 1Y \mid \varepsilon$.

$X \rightarrow \varepsilon$
$Y \rightarrow \varepsilon$
$S \rightarrow XYX$
$S \rightarrow \varepsilon \cdot \varepsilon \cdot \varepsilon$ $(\because X \rightarrow \varepsilon, Y \rightarrow \varepsilon)$
$S \rightarrow \varepsilon$

S, X, Y are nullable variables.

$X \rightarrow 0X$	$Y \rightarrow 1Y$
$X \rightarrow 0 \cdot \varepsilon$ $(X \rightarrow \varepsilon)$	$Y \rightarrow 1 \cdot \varepsilon$ $(Y \rightarrow \varepsilon)$
$X \rightarrow 0$	$Y \rightarrow 1$

$S \rightarrow XYX$	$S \rightarrow XYX$	$S \rightarrow XYX$
$S \rightarrow \varepsilon YX$	$S \rightarrow X \varepsilon X$	$S \rightarrow XY\varepsilon$
$S \rightarrow YX$	$S \rightarrow XX$	$S \rightarrow XY$

$S \rightarrow XYX \mid YX \mid XX \mid XY \mid X \mid Y$
$X \rightarrow 0X \mid 0$
$Y \rightarrow 1Y \mid 1$

$S \rightarrow XYX$	$S \rightarrow XYX$
$\rightarrow \varepsilon\varepsilon X$	$\rightarrow \varepsilon Y \varepsilon$
$\rightarrow X$	$\rightarrow Y$

Regular Expression

$\cdot$ $a \cdot b$

$+$ $(1 - \infty)$

$*$ $(0 - \infty)$

$a^+ = \{a, aa, aaa \ldots\}$

$a^* = \{\text{' '}, a, aa, aaa \ldots\}$

$(a \cdot b)^* = \{\text{' '}, ab, abab, \ldots\}$

$(a + b) = \{a, b\}$

$(aa + bb) = \{aa, bb\}$

$(a+b)^* = \{\overset{0}{\text{' '}}, \overset{1}{a}, \overset{1}{b}, \overset{2}{aa}, \overset{2}{ab}, \overset{2}{bb}, \overset{2}{ba}, \ldots\}$

Q Write a regular expression for language l $\Sigma(a, b)$ where it denotes all the Strings ending with abb.

Regular Expression : $(a+b)^* abb$

Q Write a RE for language l $\Sigma(a, b)$ where it denote all String starts with abb or ends with bbb

$abb(a+b)^* + (a+b)^* bbb$

abbb

→ String either starting with abb or ending with bbb.

$$[abb\,(a+b)^*\,(aaa+aab+abb+bba+baa+bab+aba)] + [(aaa+aab+aba+bbb+bba+bab+baa)\,(a+b)^*\,bbb]$$

$$+\ abb + bbb$$

→ All the odd binary numbers

$$(0+1)^*\,1$$

→ RE for even length string

$$(0+1)^* \cdot (0+1)^* \qquad \text{Ans}\ (00+11+01+10)^*$$

$$((0+1)\cdot(0+1))^*$$

→ RE for odd length string

$$(00+11+01+10)^*\,(0+1)$$

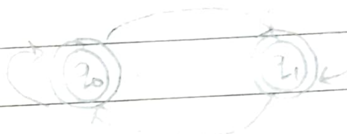
$$((0+1)(0+1))^*\,(0+1)$$

HW

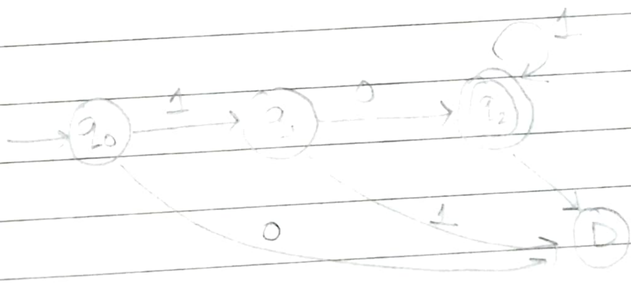
Design an expression for a language where binary number generates all numbers divisible by 3

Regular Expression to DFA

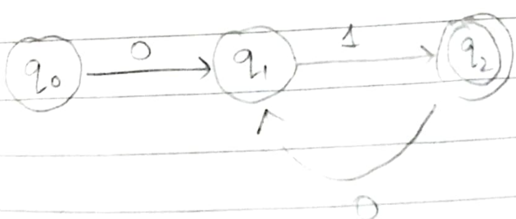
$(0+1)^*$ →



101^* →

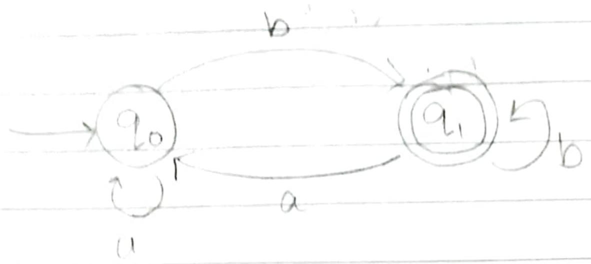


$(01)^*$ → , 01, 0101,

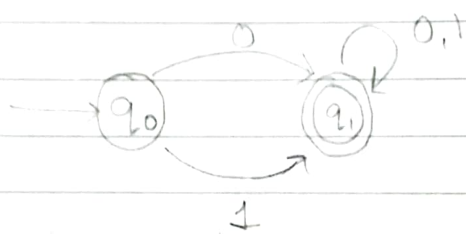


$(a+b)^* \, abb$ '', a, b; ab, aa, ..., bb

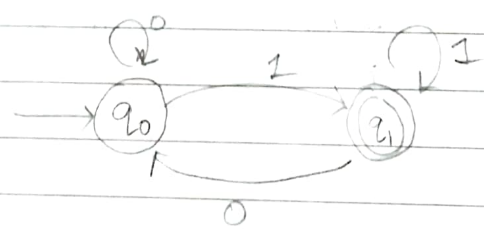
ababb



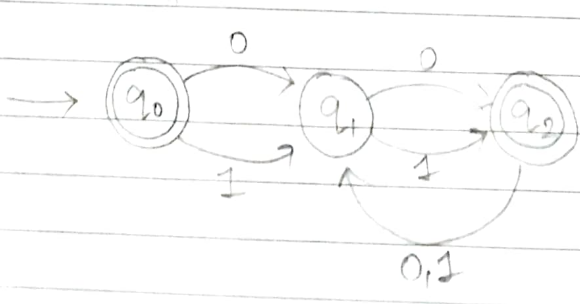
$(0+1)^+$

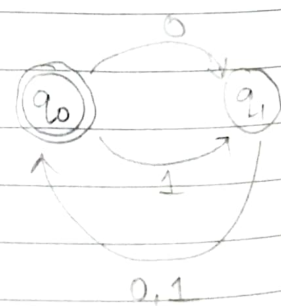


$(0+1)^* \, 1 \rightarrow$

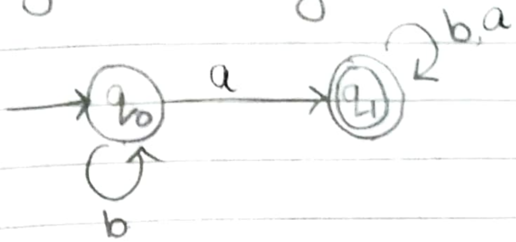


$(01+10+11+00)^*$

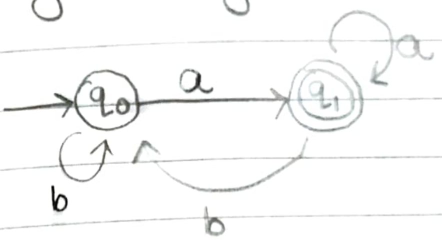




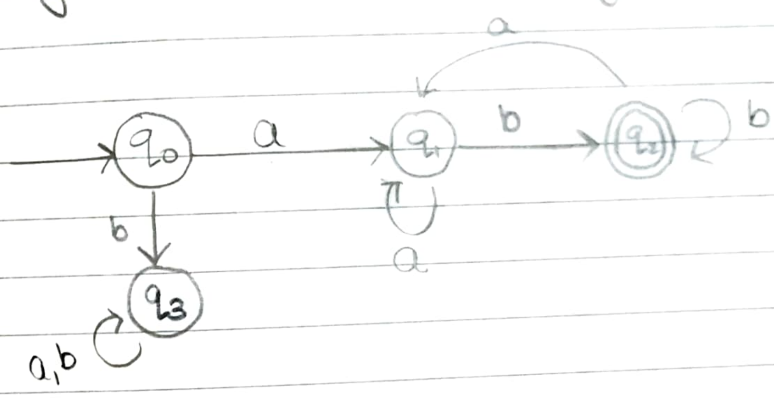
* String containing 'a'



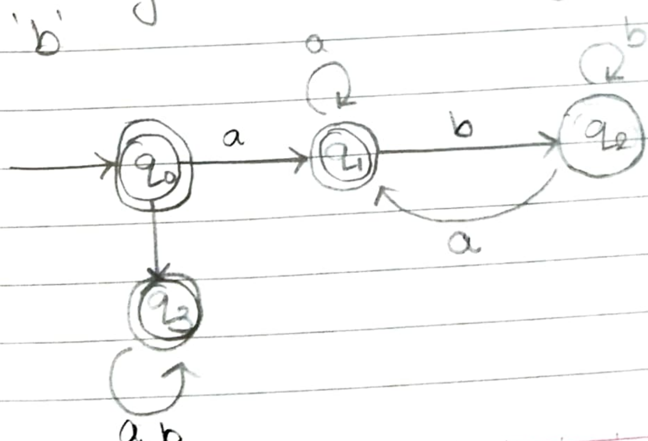
* String ending with 'a'



* Starting with 'a' and ending with 'b'



* String not starting with 'a' or not ending with 'b'
We can say this as starting with 'a' and ending with 'b'



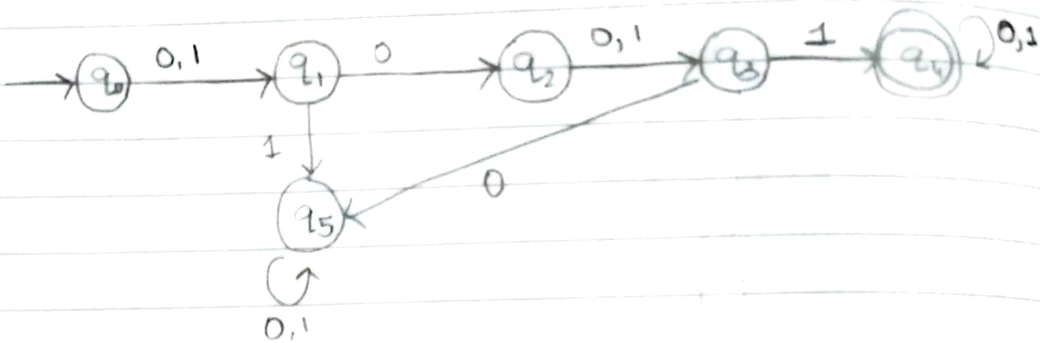
while taking compliment we change final state to not final state and vice versa

Note: In this type of DFA minimum states are $(n+1)$; n = length of str

* $\Sigma(0,1)$

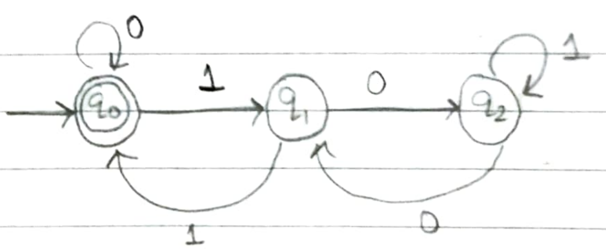
All Strings over $(0,1)$ in which second symbol is '0' and fourth symbol is '1'.

_ 0 _ 1



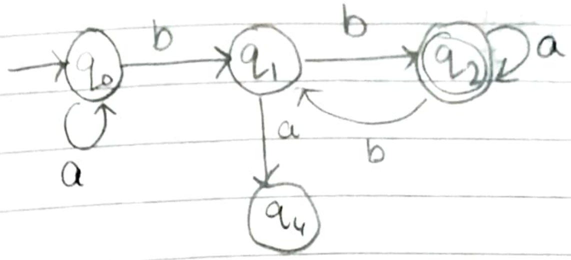
* Construct a DFA which accepts a language of all binary strings divisible by three over $\Sigma(0,1)$

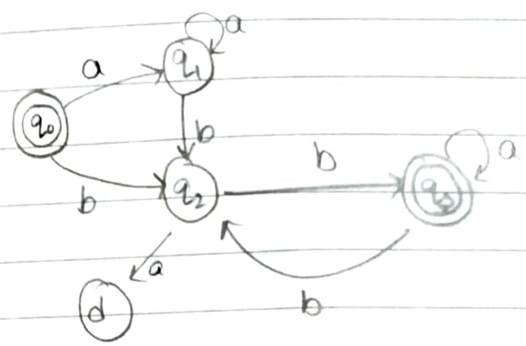
Possible Remainder	States
0	q_0
1	q_1
2	q_2



① $(a^* bb a^*)^*$

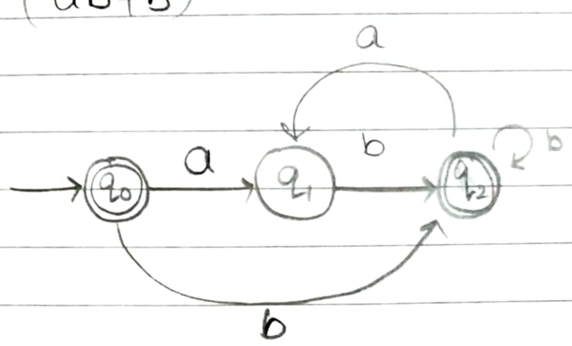
L: always 2 Consecutive b





② $(ab+b)^*$

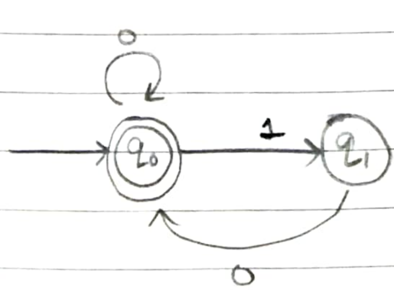
L: one 'a' followed by 'b'



IMP

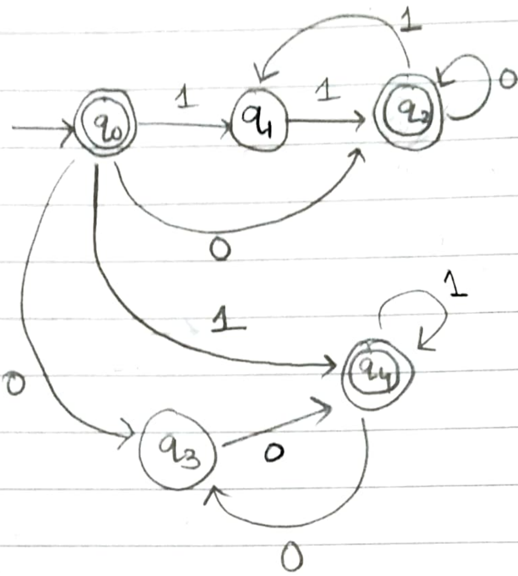
③ $(0^* 1 0^* 1 0^*)^*$

L: count of 1 is even



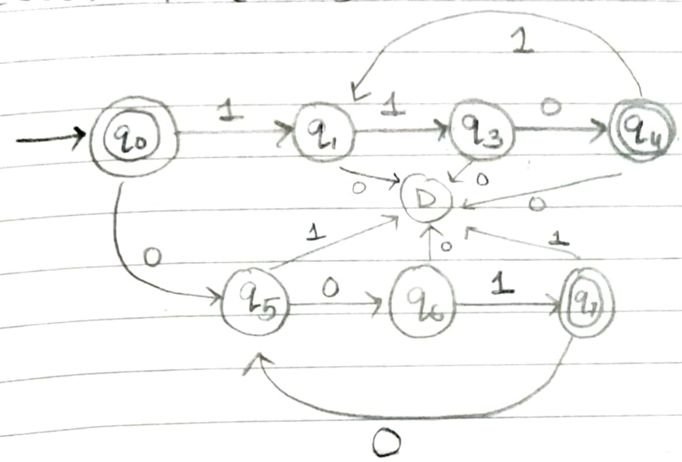
IMP

(4) $(11+0)^* + (1+00)^*$



NFA

5. $(110)^* + (001)^*$

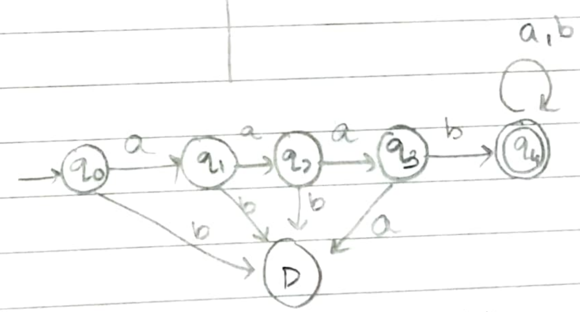


6. $a(aab)^+ ab$

$a(aab)^+(a+b)^*$

$= a(aab)(aab)^*(a+b)^*$

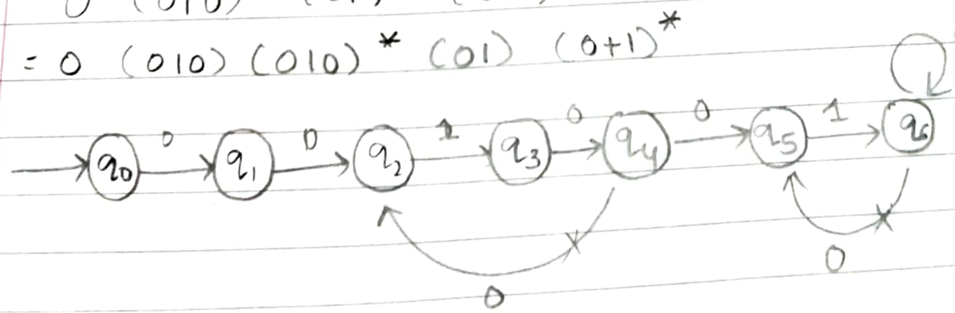
$= a(aab)(a+b)^*$

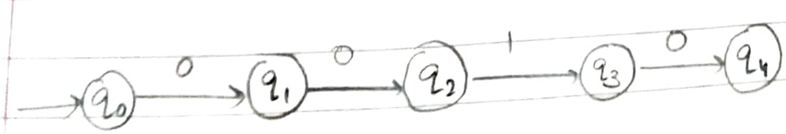


7. $0(010)^+(01)^+(0+1)^*$

$= 0(010)(010)^*(01)(0+1)^*$

0010010

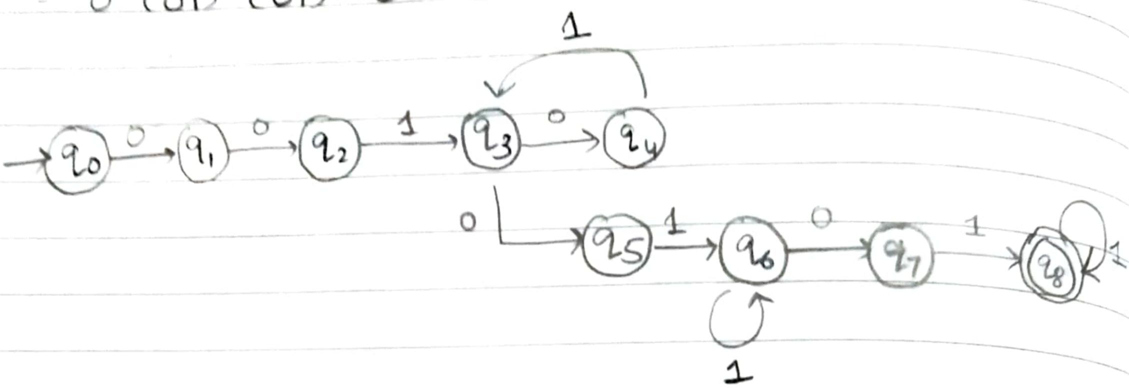




8

$0(01)^+ (01)^* 0 1^+ 0 1^+$

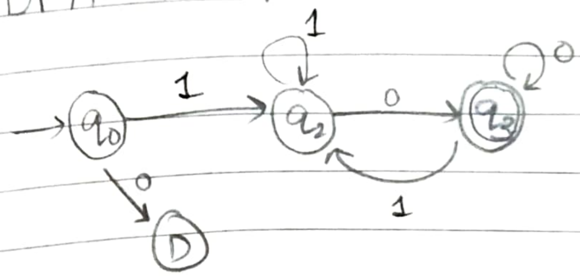
$= 0 (01) (01)^* 0 1^+ 0 1^+$



δ_N	0	1
q_0	$\{q_1\}$	$\{D\}$
q_1	$\{q_2\}$	$\{D\}$
q_2	$\{D\}$	$\{q_3\}$
q_3	$\{q_4, q_5\}$	$\{D\}$
q_4	$\{D\}$	$\{q_3\}$
q_5	$\{D\}$	$\{q_6\}$
q_6	$\{q_7\}$	$\{q_6\}$
q_7	$\{D\}$	$\{q_8\}$
q_8	$\{q_8\}$	$\{q_8\}$

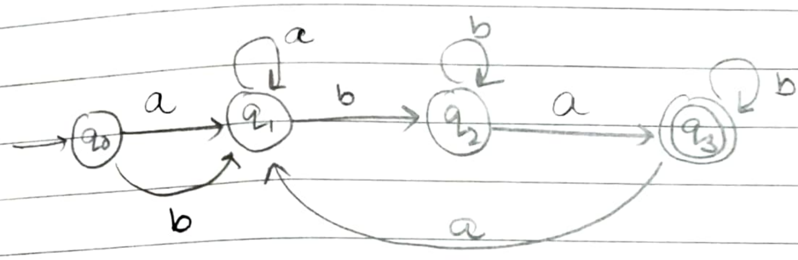
State \ Σ	0	1
$[q_0]$	$[q_1]$	$[D]$
$[q_1]$	$[q_2]$	$[D]$
$[q_2]$	$[D]$	$[q_3]$
$[q_3]$	$[q_4 q_5]$	$[D]$
$[q_4 q_5]$	$[D]$	$[q_3 q_6]$
$[q_3 q_6]$	$[q_4 q_5 q_7]$	$[q_6]$
$[q_4 q_5 q_7]$	$[D]$	$[q_3 q_6 q_8]$
$[q_3 q_6 q_8]$	$[q_4 q_5 q_7 q_8]$	$[q_6 q_8]$
$[q_4 q_5 q_7 q_8]$	$[q_7 q_8]$	$[q_3 q_6 q_8]$
$[q_6 q_8]$	$[q_7 q_8]$	$[q_6 q_8]$
$[q_7 q_8]$	$[q_8]$	$[q_8]$
$[q_8]$		

* DFA to RE

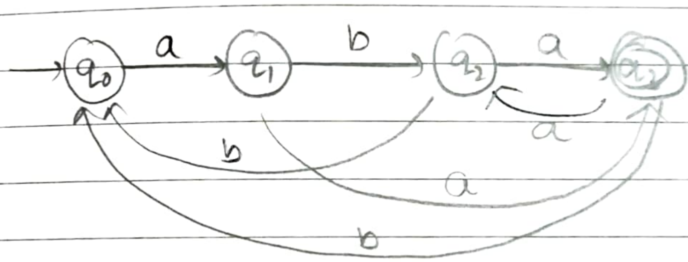
1. 

$11^*00^*(11^*00^*)^*$

$= 1^+0^+(1^+0^+)^*$

2. 

$(a+b)a^*bb^*ab^*(aa^*bb^*ab^*)^*$

3. 

$a(ba+a)(aa)^*(ba(ba+a)(aa)^*)^*$

$a(b(bab)^*a+a)$

$a(b(a(aa)^*$