CS0322 Theory of Computation YVYash

Mid Semester Examination [2021-22]

Name: YASH VERMA 19075083 CSE B.tech

---

1. Given:

$\Sigma = \{0,1\}$ and $A \subseteq \Sigma^*$ be a regular language

$B = \{x \in \Sigma^*$: there exists a string $w \in A$ such that x is obtained from w by deleting symbols $\}$

Prove/Disprove: B is regular

Proof: If we are able to find a DFA or NFA that accepts B, then we can say it is regular.

We know that A is a regular language.

This means there is a DFA say $D_0 = (Q, \Sigma, \delta, q, F)$ that accepts A.

Now, let us define a NFA $N_0 = (Q, \Sigma, \delta', q_0, F)$ where we define δ' as

i) $\delta'(q, x) = \delta(q, x) \;\forall q \in Q, x \in \Sigma$

i.e. if the input symbol belongs to Σ then the transition in the NFA N_0 then the transition in the NFA N_0 is same as DFA D_0.

(ii) $\delta'(q, \varepsilon) = \delta(q, y) \;\forall q \in Q, y \in \Sigma$

i.e. ε-transition from a state to another state in NFA N_0 which is same as the transition in DFA D_0.

Now, this NFA is obtained by adding ϵ-transition to DFA D_0 and the ϵ-transition is corresponding to transition in D_0. We can also state clearly that this NFA N_0 accepts B as we can put ϵ in place of any ~~default~~ deleted symbol and use ϵ-transition.

Since, there exist a ϵ-NFA that accepts B. This means there also exists a NFA and DFA that accepts B.

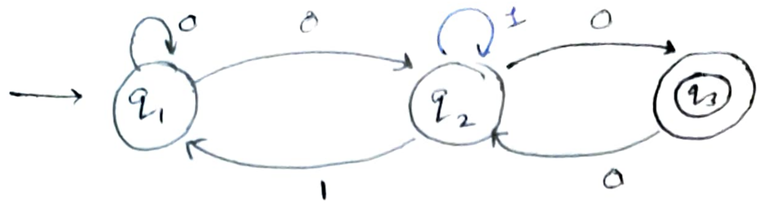
$\Rightarrow$ B is a regular language.

Hence Proved.

Yash Verma

YvYash

2.



For the given finite automata, the equations for q_1, q_2 and q_3 can be found:

The equations are:

$$q_1 = q_1 \cdot 0 + q_2 \cdot 1 + \varepsilon \quad -(1)$$

$$q_2 = q_1 \cdot 0 + q_2 \cdot 1 + q_3 \cdot 0 \quad -(2)$$

$$q_3 = q_2 \cdot 0 \quad -(3)$$

Now solving the equations,

by using (3) in (2) we get

$$\cancel{q_3}\; q_2 = q_1 \cdot 0 + q_2 \cdot 1 + q_2 \cdot 0 \cdot 0$$

$$\Rightarrow q_2 = q_1 \cdot 0 + q_2 \cdot (1 + 0 \cdot 0)$$

$$\Rightarrow q_2 = q_1 \cdot 0 + q_2 \cdot (1 + 0 \cdot 0)$$

The equation for q_2 is of the form

$$R = Q + RP$$

and we know in this case $R = QP^*$

$$\Rightarrow q_2 = q_1 \cdot 0 \cdot (1 + 0 \cdot 0)^* \quad -(4)$$

Yash Verma
YVYash

Putting (4) in (1)

$$q_1 = q_1 \cdot 0 + q_2 \cdot 1 + \varepsilon$$

$$\Rightarrow q_1 = q_1 \cdot 0 + q_1 \cdot 0 \cdot (1 + 0 \cdot 0)^* \cdot 1 + \varepsilon$$

Equation is again of the form

$$R = Q + RP$$

$$R = QP^*$$

$$\Rightarrow q_1 = \varepsilon\,(0 + 0.(1 + 0.0)^*.1)^* \quad -(5)$$

Putting (5) in (4)

$$q_2 = (0 + 0.(1 + 0.0)^*.1)^*.0.(1 + 0.0)^*.0 \quad -(6)$$

~~This is the accepting state. Hence, Regular expression accepted by Finite Automata.~~

$$\Rightarrow (0 + 0.(1 + 0.0)^*.1)^*.0.(1 + 0.0)^*.0$$

Putting (6) in (3)

$$q_3 = (0 + 0.(1 + 0.0)^*.1)^*.0.(1 + 0.0)^*.0$$

This is the accepting state. Hence, regular expression accepted by Finite Automata.

$$\Rightarrow (0 + 0.(1 + 0.0)^*.1)^*.0.(1 + 0.0)^*.0$$

Yash Verma

Yrvyash

3.) Minimise using Myhill Nerode Theorem

Transition Table:

	a	b
→ A	B	C
B	C	E
C	D	C
D	C	E
(E)	B	E

Step 1: Dividing the states as final and non-final

F = {E}

NF = {A, B, C, D}

Step 2: Marking all pairs of final and non-final state by x.

	A	B	C	D	E
A	-	-	-	-	-
B		-	-	-	-
C			-	-	-
D				-	-
E	x	x	x	x	-

Such pairs are:

(A, E), (B, E), (C, E), (D, E)

Yash Verma

yvyash

Step 3: Marking x's for NF State pairs:

NF state pairs are:

(A, B), (A, C), (A, D)

(B, C), (B, D), (C, D)

(A, B) $\quad \delta((A,B), a) = (B, C)$ No
$\quad\quad\quad\; \delta((A,B), b) = (C, E)$ Yes $\Big\} \times$

(A, C) $\quad \delta((A,C), a) = (B, D)$ No
$\quad\quad\quad\; \delta((A,C), b) = (C, C)$ No

(A, D) $\quad \delta((A,D), a) = (B, C)$ No
$\quad\quad\quad\; \delta((A,D), b) = (C, E)$ Yes $\Big\} \times$

(B, C) $\quad \delta((B,C), a) = (C, D)$ No
$\quad\quad\quad\; \delta((B,C), b) = (E, C)$ Yes $\Big\} \times$

(B, D) $\quad \delta((B,D), a) = (C, C)$ No
$\quad\quad\quad\; \delta((B,D), b) = (E, E)$ No

Yash Verma
YvYash

(C, D) $\quad \delta((C,D), a) = (D, C)$ No
$\quad\quad\quad\; \delta((C,D), b) = (C, E)$ Yes $\Big\} \times$

	A	B	C	D	E
A	–	–	–	–	–
B	×	–	–	–	–
C	O	×	–	–	–
D	×	O	×	–	–
E	×	×	×	×	–

~~Step~~

The equivalence classes are $\{A, C\}$, $\{B, D\}$, $\{E\}$

Step 4: Making the leftover blocks by O (as ~~to~~ no more x's can be marked)

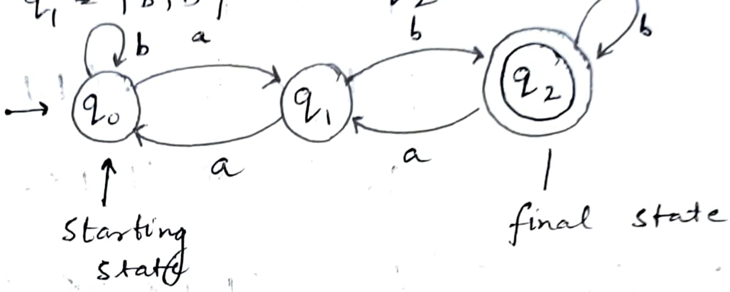
Step 5: The minimized machine is given by the state pairs for which there are marked 0's. (combined actually) and the remaining states (singleton)

i.e. The states are: (A, C), (B, D), E

$\therefore$ Minimized FA:

Let's denote $q_0 = \{A, C\}$, $q_1 = \{B, D\}$ and $q_2 = \{E\}$

	a	b
$\rightarrow \{A, C\}$	$\{B, D\}$	$\{A, C\}$
$\{B, D\}$	$\{A, C\}$	$\{E\}$
$* \{E\}$	$\{B, D\}$	$\{E\}$



Finding Language Recognized by FA:

For finding the language, we find the Regular Expression corresponding to this FA using Ardon's Theorem.

The equation obtained are:

$$q_0 = q_1 \cdot b + q_2 \cdot a + \varepsilon \quad - (1)$$

$$q_1 = q_0 \cdot a + q_2 \cdot a \quad - (2)$$

$$q_2 = q_1 \cdot b + q_2 \cdot b \quad - (3)$$

Yash Verma
YVYash

From eq^n (3):

$$q_2 = q_1 \cdot b + q_2 \cdot b$$

$$q_2 = q_1 b \cdot b^* \quad - (4)$$

find minimum

Putting (4) in (2) :

Yash Verma
YVYash

$$q_1 = q_0 \cdot a + q_2 \cdot a$$

$$= q_0 \cdot a + q_1 \cdot b \cdot b^* \cdot a$$

$$q_1 = q_0 \cdot a \cdot (b \cdot b^* a)^* \quad - (5)$$ [Arden's theorem]

Putting (5) in (1) :

$$q_0 = \varepsilon + q_0 \cdot b + q_1 \cdot a$$

$$= \varepsilon + q_0 \cdot b + q_0 \cdot a \cdot (b \cdot b^* a)^* \cdot a$$

$$= \varepsilon + q_0 [b + a \cdot (b \cdot b^* a)^* \cdot a]$$

$$\Rightarrow q_0 = \varepsilon [b + a \cdot (b \cdot b^* a)^* \cdot a]^* \quad - (6)$$

Putting (6) in (5)

$$q_1 = [b + a \cdot (b \cdot b^* a)^* \cdot a]^* a (b b^* a)^* \quad - (7)$$

Putting (7) in (4), we have :

$$q_2 = [b + a \cdot (b \cdot b^* a)^* \cdot a]^* a (b b^* a)^* b b^*$$

As, q_2 is final state of automata,

The Language accepted by FA has regular expression $\Rightarrow$

$$\boxed{(b + a \cdot (b \cdot b^* \cdot a)^* \cdot a)^* a (b b^* a)^* b b^*}$$

Language accepted by FA :

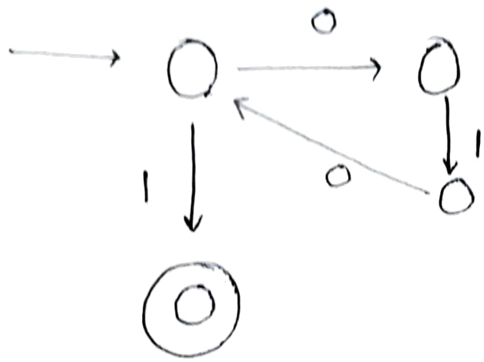
$$\boxed{L = \{\{b\} \cup \{a \{b b^* a\}^* a\{b b^* a\}^* b b^*}$$

4. $R = (010)^*1 + (1^*0)^*$ over $\Sigma = \{0, 1\}$

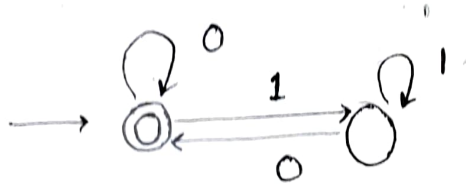
We will find minimum state finite automata for R:

1.)

1.) for part 1: $(010)^*1$:-



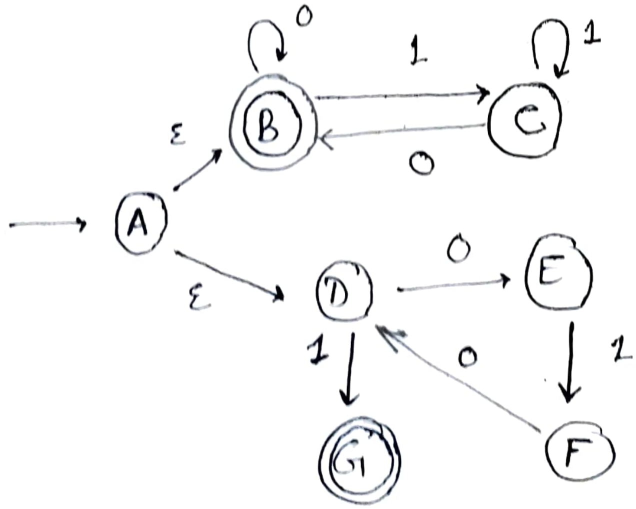
2. For part 2: $(1^*0)^*$:-



We construct an ε-NFA for regular expression R.

R = part 1 + part 2

So, we can combine them is such a way :-



Yash Verma

Yv Yash

We can use ECLOSE() to remove the ε-transition

ECLOSE(A) = {A, B, D}

ECLOSE(B) = {B}

ECLOSE(C) = {C}

ECLOSE(D) = {D}

ECLOSE(E) = {E}

ECLOSE(F) = {F}

ECLOSE(G) = {G}

In our ε-NFA B and G were in final states so all states containing B/G will be our final states in this finite ~~aut~~ automata.

Transition Table:

δ	0	1
→ *{A,B,D}	{B,E}	{C,G}
*{B, E}	{B}	{C, F}
*{C, G}	{B}	{C}
* {B}	{B}	{C}
{C,F}	{B, D}	{C}
{C}	{B}	{C}
*{B, D}	{B, E}	{C, G}

Yash Verma
YvYash

Final states: {{A, B, D}, {B,E}, {C,G}, {B}, {B,D}}

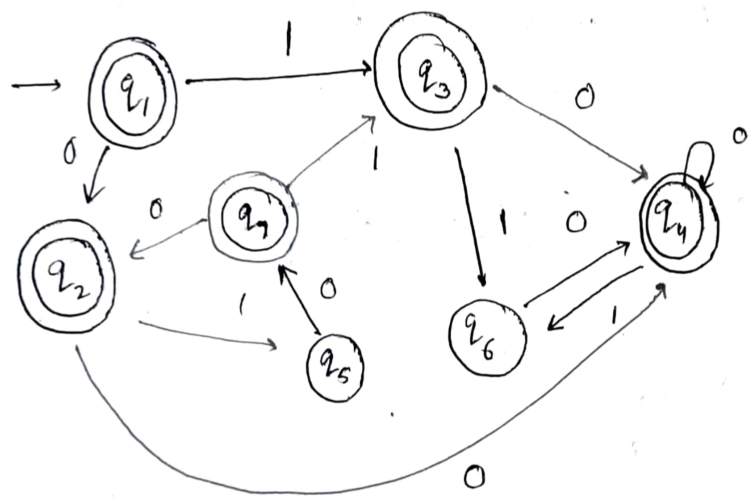
Renaming:

$\{A, B, D\} \rightarrow q_1$ $\{B, E\} \rightarrow q_2$, $\{C, G\} \rightarrow q_3$

$\{B\} \rightarrow q_4$ $\{C, P\} \rightarrow q_5$, $\{C\} \rightarrow q_6 \rightarrow \{B, D\} \rightarrow q_7$

	0	1
$\rightarrow *q_1$	q_2	q_3
$*q_2$	q_4	q_5
$*q_3$	q_4	q_6
$*q_4$	q_4	q_6
$*q_5$	q_7	q_6
q_6	q_4	q_6
$*q_7$	q_2	q_3

Yash Verma
YvYash



We will now use k-equivalent method.

All states are 0-equivalent (empty string)

Let $S_0 = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$

Q_1 : Accepting states
$\{q_1, q_2, q_3, q_4, q_7\}$

Q_2 : non-accepting states
$\{q_5, q_6\}$

$S_1 = \{\{q_5, q_6\}, \{q_1, q_2, q_3, q_4, q_7\}\}$

$$\begin{tikzcd} & Q_1 \arrow[dl] \arrow[dr] & \\ Q_3 & & Q_4 \end{tikzcd}$$

Q_3: $\{q_1, q_7\}$ always go to accepting state

Q_4: $\{q_2, q_3, q_4\}$ go to non-accepting state of any 1 input.

$S_2 = \{\{q_5, q_6\}, \{q_1, q_7\}, \{q_2, q_3, q_4\}\}$

Yash Verma

yvYash

$$\begin{tikzcd} & Q_2 \arrow[dl] \arrow[dr] & \\ Q_5 & & Q_6 \end{tikzcd}$$

Q_5: $\{q_5\}$ go to Q_3 for some input

Q_6: $\{q_6\}$ go to Q_4 for some input.

$S_3 = \{\{q_5\}, \{q_6\}, \{q_1, q_7\}, \{q_2, q_3, q_4\}\}$

$$\begin{tikzcd} & Q_4 \arrow[dl] \arrow[dr] & \\ Q_7 & & Q_8 \end{tikzcd}$$

Q_7: $\{q_2\}$ go to Q_5 for some input.

Q_8: $\{q_3, q_4\}$ go to Q_6 for some input

$S_4 = \{\{q_5\}, \{q_6\}, \{q_1, q_7\}, \{q_2\}, \{q_3, q_4\}\}$

$S_4 = \{\{q_1, q_7\}, \{q_2\}, \{q_3, q_4\}, \{q_5\}, \{q_6\}\}$

... ϵ-NFA:

S_4 cannot be divided further.

So, S_4 is our final partition.

Renaming:

$\{q_1, q_7\} \rightarrow S_0$

$\{q_2\} \rightarrow S_1$ — final state/accepted state.

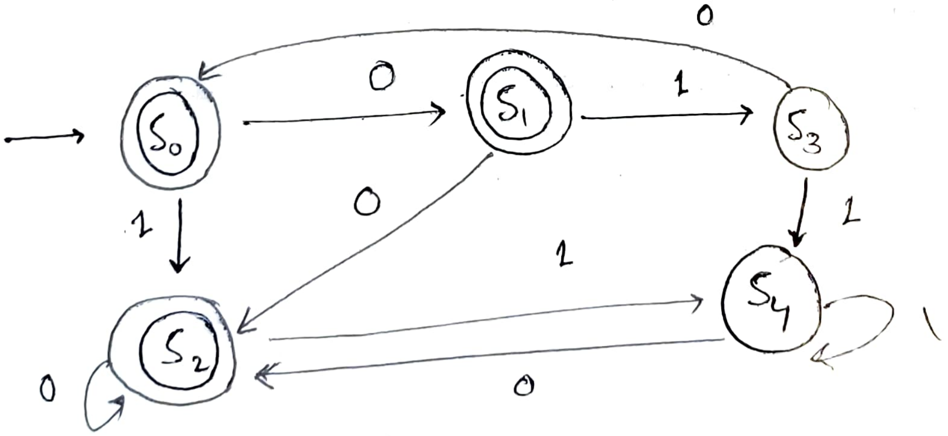
$\{q_3, q_4\} \rightarrow S_2$

$\{q_5\} \rightarrow S_3$

$\{q_6\} \rightarrow S_4$

Yash Verma
YVYash

	0	1
$\rightarrow * S_0$	S_1	S_2
$* S_1$	S_2	S_3
$* S_2$	S_2	S_4
S_3	S_0	S_4
S_4	S_2	S_4



This is our required minimized finite automata.

Now, we find Grammar for minimized FA,

Let $G = \{V_n, T, P, S\}$

$\rightarrow S = S$

$\rightarrow T = \{0, 1\}$

Representing states as:

$S_0 \rightarrow Z_0$, $S_1 \rightarrow Z_1$, $S_2 \rightarrow Z_2$, $S_3 \rightarrow Z_3$, $S_4 \rightarrow Z_4$

$V_n = \{Z_0, Z_1, Z_2, Z_3, Z_4\}$

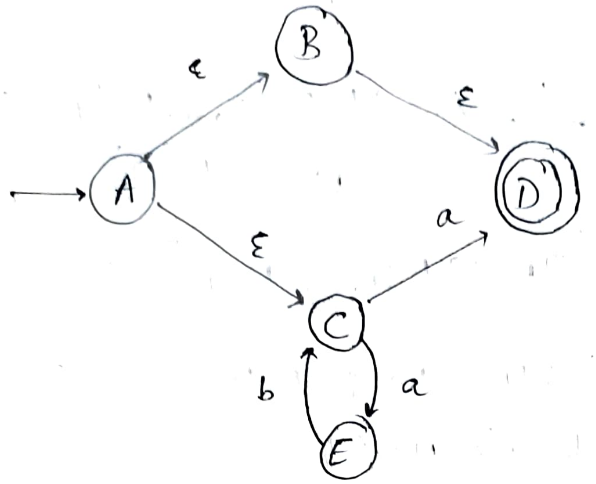
Yash Verma

YVYash

$$P = \begin{cases} S \rightarrow Z_0 \\ Z_0 \rightarrow \varepsilon \mid 0Z_1 \mid 1Z_2 \\ Z_1 \rightarrow \varepsilon \mid 1Z_3 \mid 0Z_2 \\ Z_2 \rightarrow \varepsilon \mid 0Z_2 \mid 1Z_4 \\ Z_3 \rightarrow 0Z_0 \mid 1Z_4 \\ Z_4 \rightarrow 0Z_2 \mid 1Z_4 \end{cases}$$

This is required Grammar.

5. Given ϵ - NFA :



Yash Verma
YVYash

Conversion of ϵ-NFA to FA :

Finding E-closure of all states :

ECLOSE (A) = {A, B, C, D}
ECLOSE (B) = {B, D}
ECLOSE (C) = {C}
ECLOSE (D) = {D}
ECLOSE (E) = {E}

Constructing transition table for equivalent DFA :

δ_D	a	b
→ * {A, B, C, D}	{B, D, E}	ϕ
*{B, D, E}	{B, D}	{C}
* {B, D}	{B, D}	ϕ
{C}	{D, E}	ϕ
* {D, E}	ϕ	{C}

Now, states are inserted into the table to explore and find other reachable states.

Starting state of DFA would be the E-CLOSURE of the starting state of E-NFA i.e. $\{A, B, C, D\}$

$$\delta_D(\{A,B,C,D\}, a) \cdot ECLOSE(\delta_E(A,a) \cup \delta_E(B,a) \cup \delta_E(C,a) \cup \delta_E(D,a))$$
$$= ECLOSE(\phi \cup \{B\} \cup \{D,E\} \cup \phi\}$$
$$= ECLOSE(B) \cup ECLOSE(D) \cup ECLOSE(E)$$
$$= \{B,D\} \cup \{D\} \cup \{E\}$$
$$= \{B, D, E\}$$

$$\delta_D(\{A,B,C,D\}, b) = ECLOSE(\delta_E(A,b) \cup \delta_E(B,b) \cup \delta_E(C,b) \cup \delta_E(D,b))$$
$$= ECLOSE(\phi \cup \phi \cup \phi \cup \phi)$$
$$= \phi$$

$$\delta_D(\{B,D,E\}, a) = ECLOSE(\delta_E(B,a) \cup \delta_E(D,a) \cup \delta_E(\overset{E,a}{\cancel{D,b}}))$$
$$= ECLOSE(B \cup \phi \cup \phi)$$
$$= \{B, D\}$$

Yash Verma
YVYash

$$\delta_D(\{B,D,E\}, b) = ECLOSE(\delta_E(B,b) \cup \delta_E(D,b) \cup \delta_E(E,b))$$
$$= ECLOSE(\phi \cup \phi \cup C)$$
$$= \{C\}$$

$$\delta_D(\{B,D\}, a) = ECLOSE(\delta_E(B,a) \cup \delta_E(D,a))$$
$$= ECLOSE(B \cup \phi)$$
$$= \{B, D\}$$

$$\delta_D(\{B,D\}, b) = ECLOSE(\delta_E(C,a)) = ECLOSE(\{D,E\}) = \{D,E\}$$
$$= ECLOSE(\phi \cup \phi)$$
$$= \phi$$

$\delta_D(\{C\}, a) = ECLOSE(\delta_E(C, a)) =$
$= ECLOSE(\{D, E\}) = \{D, E\}$

$\delta_D(\{C\}, b) = ECLOSE(\delta_E(C, b)) = ECLOSE(\phi) = \phi$

$\delta_D(\{D, E\}, a) = ECLOSE(\delta_E(D, a) \cup \delta_E(E, a))$
$= ECLOSE(\phi \cup \phi)$
$= \phi$

$\delta_D(\{D, E\}, b) = ECLOSE(\delta_E(D, b) \cup \delta_E(E, b))$
$= ECLOSE(\phi \cup \{C\})$
$= \{C\}$

The final states is the constructed DFA would be sets of states containing final states of the ϵ-NFA which are:
$\{A, B, C, D\}$, $\{B, D, E\}$, $\{B, D\}$ and $\{D, E\}$

Let $q = \{A, B, C, D\}$, $q_1 = \{B, D, E\}$, $q_2 = \{B, D\}$, $q_3 = \{C\}$,
$q_4 = \{D, E\}$.

Yash Verma
YVYash

Then required DFA D is given as:

$D = \{Q_D, \Sigma, \delta, q_0, F_D\}$

where

$Q_D = \{q_0, q_1, q_2, q_3, q_4\}$

$F_D = \{q_0, q_1, q_2, q_4\}$

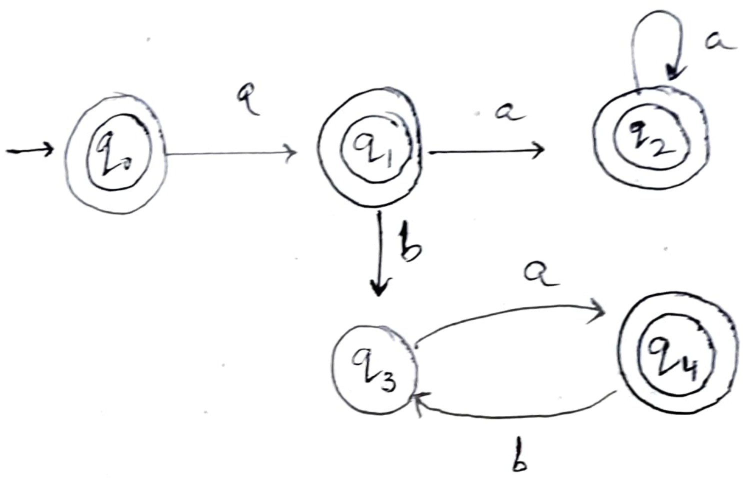
$\Sigma = \{a, b\}$ and

~~δ_D is given as:~~

	~~a~~	~~b~~
~~q_0~~		
~~q_1~~		
~~q_2~~		

δ_D is given as:

	a	b
q_0	q_1	ϕ
q_1	q_2	q_3
q_2	q_2	ϕ
q_3	q_4	ϕ
q_4	ϕ	q_3



Yash Verma

YvYash

6. Given :

$M = (Q, \Sigma, q_0, \delta, A)$ is ϵ-NFA

The recursive definition of ϵ-CLOSURE ECLOSE(q), q is state ~~p~~ in Q.

Basis: ECLOSE (q) = {q}

Induction: If $p \in$ ECLOSE (q), then if

$\delta(p, \epsilon) = r \Rightarrow$ ECLOSE (q) = ECLOSE (q) $\cup$ {r}

$\Rightarrow r \in$ ECLOSE(q)

Now, the recursive definition of for $S \subseteq Q$.

ECLOSE (S):

Basis: So each $p \in S$, $p \in$ ECLOSE (S)

Induction: If $p \in$ ECLOSE(S), then if $\delta(p, \epsilon) = r$

ECLOSE(S) = ECLOSE(S) $\cup$ {r}

$\Rightarrow \boxed{r \in \text{ECLOSE}(S)}$

Yash Verma
YVYash

(a) S, T are subsets of Q, such that $S \subseteq T$, then ECLOSE(S) $\subseteq$ ECLOSE (T)

Now, constructing the ECLOSE (S), ECLOSE(T) simultaneously.

Basis: If $p \in S$, $p \in$ ECLOSE(S)

Induction: If $p \in$ ECLOSE (S), then if $\delta(p, \epsilon) = r$

ECLOSE (S) = ECLOSE (S) $\cup$ {r}

ECLOST(T) = ECLOSE (T) $\cup$ {r}

$\Rightarrow r \in$ ECLOSE(T), $r \in$ ECLOSE (S)

By repeating the processo we observe every ~~allotment~~ element in ECLOSE(s) is also belongs to ECLOSE(T)

$\therefore$ ECLOSE(s) $\subseteq$ ECLOSE(T)

b.) If $S, T \subseteq Q$, then ECLOSE($S \cup T$) = ECLOSE(S) $\cup$ ECLOSE(T)

Now, constructing the ECLOSE($S \cup T$), ECLOSE(S) & ECLOSE(T) simultaneously using recursive definition

Basis: If $p \in S, q \in T \Rightarrow p, q \in$ ECLOSE(S)
$q \in$ ECLOSE(T)

as $p \in S, q \in T \Rightarrow p, q \in S \cup T$

$\Rightarrow p, q \in$ ECLOSE($S \cup T$)

Till now, if $p \in S$ or $p \in T$

$\Rightarrow p \in$ ECLOSE(S) $\cup$ ECLOSE(T)

as $p \in S \cup T \Rightarrow p \in$ ECLOSE($S \cup T$)

$\Rightarrow$ ECLOSE(S) $\cup$ ECLOSE(T) $\subseteq$ ECLOSE($S \cup T$)

and as if $q \in S \cup T \Rightarrow$ there $q \in$ ECLOSE($S \cup T$)

then $q \in S$ or $q \in T$

$\Rightarrow q \in$ ECLOSE(S) or ECLOSE(T)

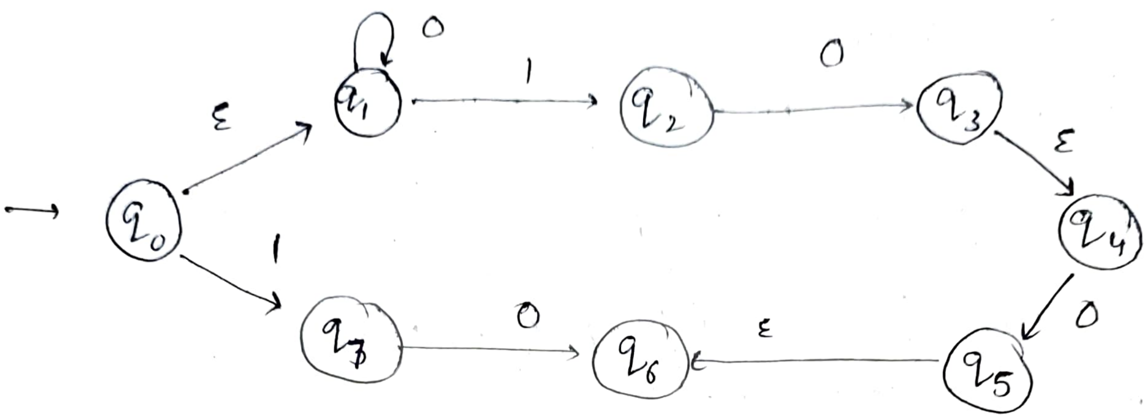
$\Rightarrow q \in$ ECLOSE(S) $\cup$ ECLOSE(T)

Yash Verma
YVYash

$\therefore$ ECLOSE(S ∪ T)

= ECLOSE(S) ∪ ECLOSE(T)

(c)



Consider $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$

and $S, T \subseteq Q \Rightarrow S = \{q_0, q_5\}$

$T = \{q_3, q_5\}$

$\Rightarrow S \cap T \Rightarrow \{q_5\}$ using recursive definition

$ECLOSE(q_0) = \{q_0, q_1\}$

$ECLOSE(q_3) = \{q_3, q_4, q_5, q_6\}$

$ECLOSE(q_5) = \{q_5, q_6\}$

$ECLOSE(S \cap T) = ECLOSE(q_5) = \{q_5, q_6\}$

$ECLOSE(S) \Rightarrow ECLOSE(\{q_0, q_5\})$

$= \{q_0, q_1, q_3, q_8\}$

$ECLOSE(T) = ECLOSE(\{q_3, q_5\})$

$= \{q_3, q_4, q_0, q_1, q_5, q_6\}$

Yash Verma
YvYash

Induction:

if $p \in ECLOSE(S)$ or

$p \in ECLOSE(T)$

then if $\delta(p, \epsilon) \to r$

$r \in ECLOSE(S) \cup ECLOSE(T)$

As, if $p \in S$ or $p \in T \Rightarrow p \in S \cup T$

$\Rightarrow p \in ECLOSE(S \cup T)$ if $p \in ECLOSE(S)$

or $p \in ECLOSE(T)$

then $\exists\, q \in S$ (or) $q \in T$ such that

$\delta(q, \epsilon) = p$

$\Rightarrow q \in S \cup T \Rightarrow q \in ECLOSE(S \cup T)$

Then, $\delta(q, \epsilon) = p \in ECLOSE(S \cup T)$ $\cancel{ECLOSE(S \cup T)}$
$\cup \{r\}$

~~$\delta(p, \epsilon) = r \Rightarrow ECLOSE(S \cup T)$~~

$\delta(p, \epsilon) = r \Rightarrow ECLOSE(S \cup T) = ECLOSE(S \cup T) \cup \{r\}$

$r \in ECLOSE(S \cup T)$

$ECLOSE(S) \cup ECLOSE(T) \subseteq ECLOSE(S \cup T)$

If $p \in ECLOSE(S \cup T)$, then either $p \in S \cup T$ or

$\exists\, q \in S \cup T \Rightarrow \delta(q, \epsilon) = p$

Similiarly, as above either, $p \in S$ or $p \in T$;

$q \in S$ or, $q \in T$

$p \in ECLOSE(S) \cup ECLOSE(T)$ or $q \in ECLOSE(S)$
$\cup ECLOSE(T)$

$\therefore p \in ECLOSE(S) \cup ECLOSE(T)$

$ECLOSE(S \cup T) \subseteq ECLOSE(S) \cup ECLOSE(T)$

Yash Verma
YvYash

$\therefore$ ECLOSE(S) $\cap$ ECLOSE(T)

$= \{q_0, q_1, q_5, q_6\}$

$\therefore$ ECLOSE$(S \cap T) \neq$ ECLOSE(S) $\cap$ ECLOSE(T)

<u>Relation</u> b/w <u>the sets</u>

$\Rightarrow$ ECLOSE$(S \cap T) \subseteq$ ECLOSE(S) $\cap$ ECLOSE(T)

not vice-versa

as we can see from example it is also satisfied.

Yash Verma

YVYash

7. Given : $L = \{0^n 1^n \mid n \geq 0\}$

(a) Find two distinct strings x and y that are indistinguishable w.r.t to L.

We know that x and y are distinguishable with respect to L if there exists a string z, so that exactly one of the strings xz and yz is in L. else the strings are indistinguishable.

The language L contains $\{01, 0011, 000111, \ldots\}$
where $\Sigma = \{0, 1\}$

$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11 \ldots\}$

Let $x = 110$ and $y = 1110$ $\quad x, y \in \Sigma^*$

$xz = 110z = 1^2 0z$
$yz = 1110z = 1^3 0z$

Since, every string in L starts with 0, neither xz or yz lies in L. Hence, the strings x and y are indistinguishable with respect to L.
where $\boxed{x = 110 \quad y = 1110}$

Yash Verma
YVYash

(b) To prove: L is a non-regular language

<u>Proof:</u> We know that a language is not regular if there exists an ~~ifi~~ infinite set of strings, any two of which are distinguishable w.r.t L.

So, we need ~~to~~ one such that set. A set S of strings is said to be pairwise distinguishable with respect to a language L if only every pair of strings in S is distinguishable w.r.t L.

Let $S = \{0^n : n \geq 0\}$

Let $x = 0^a$, $y = 0^b$ $\quad a \neq b$ (distinct)

Here, $xz = 0^a z$

$yz = 0^b z$

xz to be in L, $z = 1^a \Rightarrow yz = 0^b 1^a \notin L$.

Since, $a \neq b$.

For yz to be in L, $z = 1^b \Rightarrow xz = 0^a 1^b \notin L$, since $a \neq b$.

So, $\forall\, x, y \in S$, $\exists\, z$ such that exactly one of xz and yz ~~is~~ is in L.

Since, S is an infinite set of ~~p~~ pair-wise distinguishable strings with respect to L, L is non-regular.

Yash Verma

YV Yash

8.) $\Sigma = \{0, 1\}$

Given: $MIDDLE = \{u0v : u, v \in \Sigma^*$ and $|u| = |v|\}$

To prove: MIDDLE is not regular

Proof: We will prove this by method of contradiction but first, Let us first state Pumping Lemma.

For every alphabet Σ and every regular language $A \subseteq \Sigma^*$, there exists a pumping length $n \geq 1$ for A that satisfies the following property. For every string $w \in A$ with $|w| \geq n$, it is possible to write $w = xyz$ for strings $x, y, z \in \Sigma^*$ such that

Yash Verma
YVYash

(a) $y \neq \varepsilon$

(b) $|xy| \leq n$, and

(c) $xy^iz \in A$ for every $i \in N$.

Now, let us assume that MIDDLE is regular. By Pumping Lemma, there exists a pumping length $p \geq 1$ with the property described above.

Let $w = 1^p 0 1^p$ given that $w \in MIDDLE$ and $|w| = 2p+1 \geq p$, there must exist string $x, y, z \in \Sigma^*$ such that $w = xyz$ and for which the above ~~there~~ three item (a), (b), (c) are satisfied. As $y \neq \varepsilon$ and $|xy| \leq p$, we conclude that $y = 1^k$ for some choice of an integer k with $k \geq 1$.

As $xy^iz \in$ MIDDLE for every $i \geq 0$, k

we conclude that,

$$1^{p+k} 0 1^p = xy^2z \in \text{MIDDLE}$$

However, the string $1^{p+k} 0 1^p$ is not contained in MIDDLE, as the only 0 in this string is not in the middle position. We have, therefore proved by contradiction, that MIDDLE is not regular.

Hence proved.

19075083
Yash Verma
YvYash

9. A Context Free Grammar (CFG) can be represented as:

$$G = (V_n, \Sigma, S, P)$$

- V_n and Σ are disjoint sets of non-terminal/variable and terminal/alphabets symbols.
- P is Productions. Productions/Grammar rules is a finite set of formulas of form: $A \to \alpha$ $A \in V_n$, $\alpha \in (V_n \cup \Sigma)^*$
- S is starting symbol.

Now, we are given a string such that 0 can occur ~~consect~~ consecutively but not 1.
Start with non-terminal symbols.
We take S and A into account.

Yash.Verma
YVYash

Production produced by S:

$S \to 0S$, for the case that two 0s can occur consecutively.

We get a production rule:

$S \to 0$, as string can have last digit as '0'.

Similiarly, we get:

$S \to 1$, as string can end with '1' also.

String can have starting digit '1' which gives ~~S→1's~~ $S \to 1S$, But it cannot be followed by another 1, so production should be $S \to 1A$

$\therefore$ A cannot product '1', A can only produce '0'

So : $A \to 0$

If string start with '1', the production rule $S \to 1A$ is applied.

A may produce :

$A \to SA$, as we cannot have consecutive 1s in our language. But, in such a case 1 can only occur at the last. There is no chance of 1 to occur except at last. So, production should be:

$A \to 0S$ and not $A \to 0A$

Yash Verma
YVYash

Finally we can say :-

$S \to 0S \mid 1A \mid 0 \mid 1$

$A \to 0S \mid 0$

We have $G = (V_n, \Sigma, S, P)$ where:

$V_n = \{S, A\}$

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

$S = \{S\}$

$P = \{S \to 0S \mid 1A \mid 0 \mid 1,\ A \to 0S \mid 0\}$