Homework - 1

Qst^n: Using the pumping lemma show that $L_1 = \{w \mid w$ is a binary string with equal number of zeroes and ones$\}$ isn't a regular language.

Sol^n: If L_1 is a regular language, then L_1 has a pumping length or number of states of DFA called 'p' such that any string 'w' where $|w| \geq p$ may be divided into 3 parts $w = xyz$ such that the following conditions must be true:
i) $|y| > 0$ ii) $|xy| \leq p$ iii) $xy^iz \in L_1$ for every $i \geq 0$

From given language L_1, we get accepted languages $L_1 = \{01, 10, 0110, 1010, 0101, \ldots\}$. Following contradiction way, assume L_1 is a regular language and p be the number of states of DFA. String w considered as, $w = 0^p 1^p$ or, $1^p 0^p$ or, combinationally where $|w| \geq p$. Let $p = 2$ then $w = 0^2 1^2$ means $w = 0011$ and split it as xyz where $x = 0$, $y = 0$ and $z = 11$ because $|xy| \leq p$.
Now, for all $i \geq 0$ let $i = 2$ $\therefore xy^iz = xy^2z$

$\therefore w = \underset{x}{0}\,\underset{y}{00}\,\underset{z}{11} \notin L_1$. So, L_1 isn't a regular language.

$\boxed{\text{Showed}}$

Homework-2

Q.$5t^n$: Using the pumping lemma show that $L_2 = \{w \mid w$ is a binary string of the form $0^n 1^n$ where $n \geq 1\}$ isn't a regular language.

Sol^n: From given language L_2, we get accepted languages, $L_2 = \{01, 0011, 000111, \ldots\}$. By following contradiction way,

i) Assume L_2 is a regular language.

ii) Let, P be the number of states of DFA

iii) Given string ~~$w = 0^P$~~ considered as $w = 0^P 1^P$ where $|w| \geq P$

iv) Now split the string w into 3 parts as $w = xyz$ such that ⓐ $y > 0$ ⓑ $|xy| \leq$ and ⓒ for all $i \geq 0$, $xy^iz \in L_2$.

v) Let $P = 2$ then $w = 0^2 1^2 = \underset{x}{0}\,\underset{y}{0}\,\underset{z}{11}$ cause here $|xy| \leq P$.

vi) Now, for all $i \geq 0$ let $i = 2$ $\therefore xy^iz = xy^2z$

$\therefore w = \underset{x}{0}\,\underset{y}{00}\,\underset{z}{11} \notin L_2$

So, L_2 isn't a regular language. $\boxed{\text{showed}}$

Homework - 3

Q$_{si}^{n}$: Using the pumping lemma show that $L_3 = \{w \mid w$ is a binary string of the form $0^n 1 0^n$, where $n \geq 1\}$ isn't a regular language.

Sol^n: From given language L_3, we get accepted languages $L_3 = \{010, 00100, 0001000, \ldots\}$.

By following contradiction way,

i) Assume L_3 is a regular language.

ii) Let, P be the number of states of DFA.

iii) Given string considered as $w = 0^P 1 0^P$ where $|w| \geq P$.

iv) Now split the string w into 3 parts as $w = xyz$ such that (a) $y > 0$ (b) $|xy| \leq P$ and (c) for all $i \geq 0$ $xy^iz \in L_3$.

v) Let $P = 2$ then $w = 0^2 1 0^2 = \underset{x}{0}\,\underset{y}{0}\,\underbrace{100}_{z}$

because here $|xy| \leq P$.

vi) Now, for all $i \geq 0$ let $i = 2$ $\therefore xy^iz = xy^2z$

$\therefore w = \underset{x}{0}\,\underbrace{00}_{y}\,\underbrace{100}_{z} \notin L_3$.

So, L_3 isn't a regular language $\boxed{\text{Showed}}$

Scanned with CamScanner

Homework - 4

Qs 1^n: Using the pumping lemma show that $L_4 = \{w \mid w$ is a binary string of the form $0^m 1^n$, where $m < n$ and $m, n \geq 0$ integers$\}$ can't a regular language.

Sol^n: From given language L_4, we get accepted languages $L_4 = \{1, 011, 00111, 0001111, \dots\}$.

By following contradiction way,

i) Assume L_4 is a regular language.

ii) Let, P be the number of states of DFA.

iii) Given string considered, as $0^P 1^{P+1}$
where $p < p+1$ and $P, P+1 \geq 0$ with $|w| \geq P$

iv) Now split the string w into 3 parts as
$w = xyz$ such that ⓐ $y > 0$ ⓑ $|xy| \leq P$
and ⓒ for all $i \geq 0$. $xy^iz \in L_4$.

v) Let, $P = 2$ then $w = 0^2 1^{2+1} = 0^2 1^3 = \underset{x}{0}\,\underset{y}{0}\,\underbrace{111}_{z}$
because here $|xy| \leq P$.

vi) Now for all $i \geq 0$ let $i = 2$ $\therefore$ $xy^iz = xy^2z$

$\therefore w = \underset{x}{0}\,\underbrace{00}_{y}\,\underbrace{111}_{z} \notin L_4$.

So, L_4 isn't a regular language $\boxed{\text{showed}}$

Scanned with CamScanner

Homework-5

Qsn no: Using the pumping lemma show that
$L_5 = \{ w \mid w$ is a binary string of the form
0^n where n is a perfect square (i.e. $n = i^2$
or, 0^{i^2}, for $i \geq 1\}$ isn't a regular language.

Sol: From given languag L_5, we get accepted
languages $L_5 = \{0, 0000, 000000000, \ldots\}$.
By following contradiction way,
i) Assume L_5 is a regular language.
ii) Let, p be the number of states of DFA.
iii) Given string considered as 0^p where $|w| \geq p$.
iv) Now split the string w into 3 parts as
$w = xyz$ such that (a) $y > 0$ (b) $|xy| \leq p$
and (c) for all $i \geq 0$ $xy^iz \in L_5$.
v) ~~Let p=2 then w~~
Here,
$$w = 0^p \text{ where } p = i^2$$
$$\therefore xy = 0^q \text{ where } q \leq p$$
$$y = 0^r \text{ where } r < q$$
and $z = 0^{p-q}$ ~~where s=q~~

Scanned with CamScanner

As, $xy^iz = xyy^{i-1}z$

$= (0)^q (0^r)^{i-1} (0)^{p-q}$

$= (0)^{q+ir-r+p-q}$

$= (0)^{r(i-1)+p}$

First consider $i=1$, $xy^1z = (0)^{r(1-1)+p}$

$= (0)^p$

$= (0)^{i^2} \in L_5$

Then consider $i=2$, $xy^2z = (0)^{r(2-1)+p}$

$= (0)^{r+p}$

$= (0)^{r+i^2} \notin L_5$

$\therefore$ L_5 isn't a regular language $\boxed{\text{showed}}$

Homework-6 (Slide No-31)

Qstn: Proof using regular expressions that if L is regular then reversal of L, (L^R) is also regular.

Soln: i) Assume L is defined by a regular expression E.

ii) We show that there is another regular expression E^R such that $L(E^R) = (L(E))^R$ that is, the language of E^R is the reversal of the language of E.

Basis: If E is ϵ, ϕ or a, then $E^R = E$.

Induction: There are three cases, depending on the form of E.

(a) $E = F + G$. Then $E^R = F^R + G^R$.

Justification: The reversal of the union of two languages is obtained by computing the reversal of the two languages and taking the union of these languages.

(b) $E = F.G$ Then $E^R = G^R . F^R$

$\boxed{\text{Note}}$ We reverse the order of the two languages, as well as reversing the languages themselves.

Justification : In general, if a word $w \in L(E)$ is the concatenation of $w_1 \in L(F)$ and $w_2 \in L(G)$, then $w^R = w_2^R . w_1^R$

(c) $E = F^*$ Then $E^R = (F^R)^*$

Justification :

(i) Any string $w \in L(E)$ can be written as $w_1 . w_2 , \ldots , w_n$ where each w_i is in $L(F)$. But $w^R = w_n^R \ldots , w_2^R . w_1^R$ and each w_i^R is in $L(E^R)$. So w^R is in $L((F^R)^*)$.

(ii) Conversly, any string in $L((F^R)^*)$ is of the form $w_1 w_2 \ldots w_n$ where each w_i is the reversal of a string in $L(F)$. The reversal of this string is in $L(F^*)$ which is in $L(E)$. We have shown that a string is in $L(E)$ iff its reversal is in $L((F^R)^*)$.

Homework-7 (Slide No-42)

Qsn: Showing equivalence of DFAs between comparing two DFAs $L(DFA_1) == L(DFA_2)$.

Soln:

Conditions:

i) Each pair of states $\{q_i, q_j\}$ the transition for input $a \in \Sigma$, is defined by $\{q_a, q_b\}$ where $\delta\{q_i, a\} = q_a$ and $\delta\{q_j, a\} = q_b$. If for pair (q_a, q_b), two automata aren't equivalent if from q_a or q_b one is initial or intermediate and other one is final state.

ii) For first automation if initial state is final state then for the 2^{nd} automation also occur the same case for them to equivalent.

Scanned with CamScanner

Example:

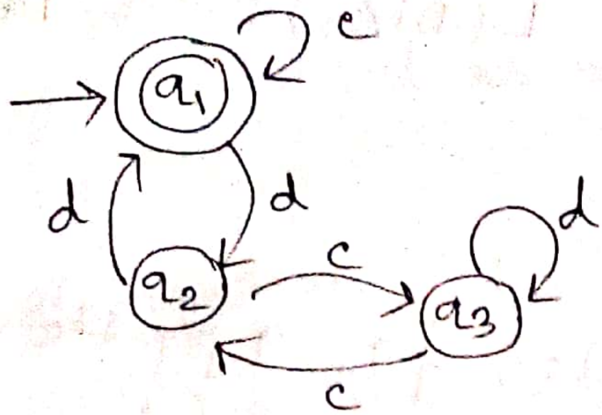


Fig: DFA$_1$

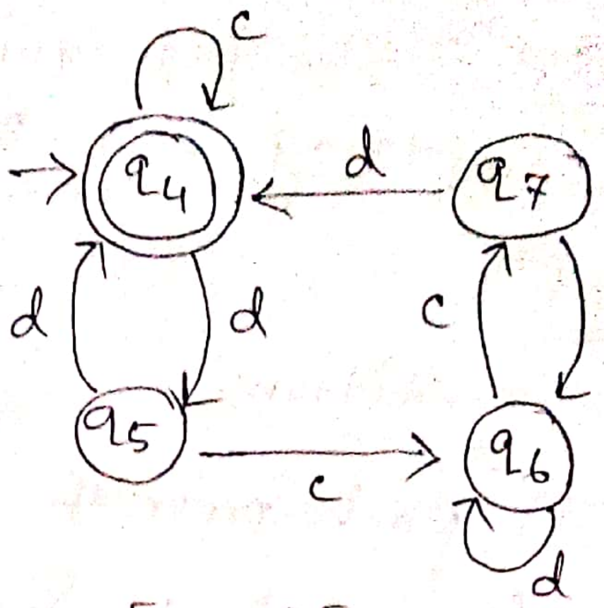


Fig: DFA$_2$

State table (IS = ~~initial~~ intermidiate state; FS = final state)

States	c	d
$\{q_1, q_4\}$	$\{q_1, q_4\}$ (FS, FS)	$\{q_2, q_5\}$ (IS, IS)
$\{q_2, q_5\}$	$\{q_3, q_6\}$ (IS, IS)	$\{q_1, q_4\}$ (FS, FS)
$\{q_3, q_6\}$	$\{q_2, q_7\}$ (IS, IS)	$\{q_3, q_6\}$ (IS, IS)
$\{q_2, q_7\}$	$\{q_3, q_6\}$ (IS, IS)	$\{q_1, q_4\}$ (FS, FS)

No combination of new pairs and for any pairs of ~~initial~~ intermidiate state and final state.

Also, for first automata initial state is the final state and for 2nd automata occur same.

So, $L(DFA_1) == L(DFA_2)$ | Showed

Homework - 8 (Slide No - 42)

Q.n: How to minimize a DFA ?

Sol^n: ~~Type-1~~ When DFA contains unreachable state (s) then we need to remove those state(s).

A state is said to be unreachable if there is no way it can be reached from the initial state. It will only have outgoing transitions and will not have incoming transitions (w/o self loop).

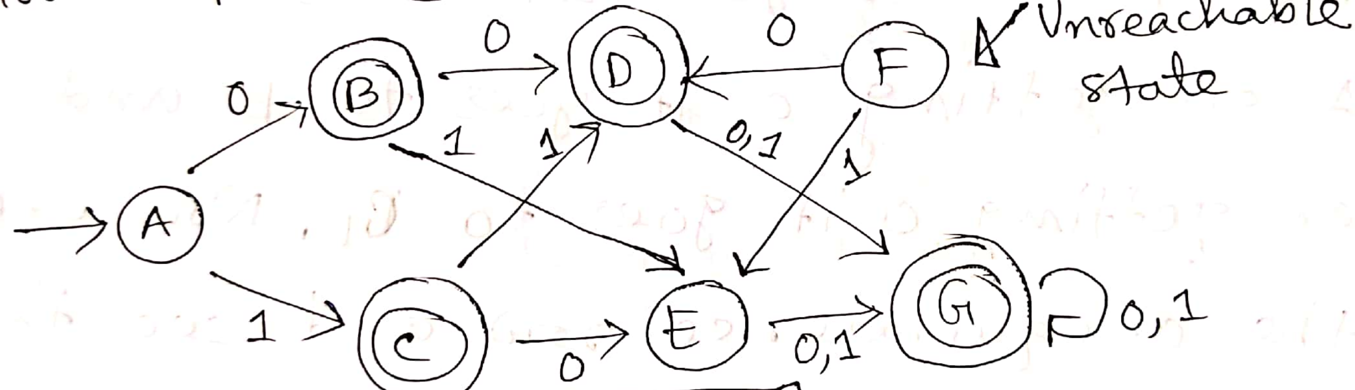


Fig: DFA 1

i) First remove the unreachable state (F) and draw the transition table.

	0	1
→A	B	C
*B	D	E
*C	E	D
D	G	G
E	G	G
*G	G	G

ii) Now check for equivalence among given states.

0-equivalence: {Non final states} {Final states}

$= \{A, D, E\} \; \{B, C, G\}$

1- equivalence: Now ~~I am~~ we are going to check

If A and D are on equivalent to each other from transition table and see where does A and D go when gets 0 and 1.

A on getting 0 it goes to B and D on getting 0 it goes to G. Now check the 0-equivalence row and see that B and G ~~aren't~~ are exist into the same set. But condition isn't complete. we ~~A~~ have to check for input 1 also.

A on getting 1 it goes to C and D on getting 1 it goes to G. Now check the 0-equivalence row and see that

C and G are exist into the same set.
So, A and D are one-equivalent.

Now ~~I am~~ We are going to check E either with A or D. Let's check A and E. Now from transition table, A on getting 0 it goes to B and E on getting 0 it goes to G. Now check the 0-equivalence row and see that B and G fall into the same set. But condition isn't complete we have to check for input 1 also. A on getting 1 it goes to C and E on getting 1 it goes to G. Now check the 0-equivalence row and see that C and G are exist into the same set.
So, A, D and E are one-equivalent.

Following the same procedure we see that for the final states B, C ~~and G~~

are one-equivalent but not G. So, we should make a new set for G.

So for 1-equivalence finally we get the set of states: {A, D, E} {B, C} {G}

2-equivalence: By following the same procedure of 1-equivalence and now check from 1-equivalence row for the 2-equivalence we get the set of states: {A} {D, E} {B, C} {G}

3-equivalence: By following the same procedure of 1-equivalence and now check from 2-equivalence row for the 3-equivalence we get the set of states: {A} {D, E} {B, C} {G}

From the rows of 2 and 3-equivalence we see the same result of set of states. So, it's time to stop the process.

From the final (3)-equivalence we get transition table for input 0 and 1 that's given below:

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

Let's assume {D,E} as {Q} and {B,C} as {P}

So the minimal version of DFA 1 becomes below:

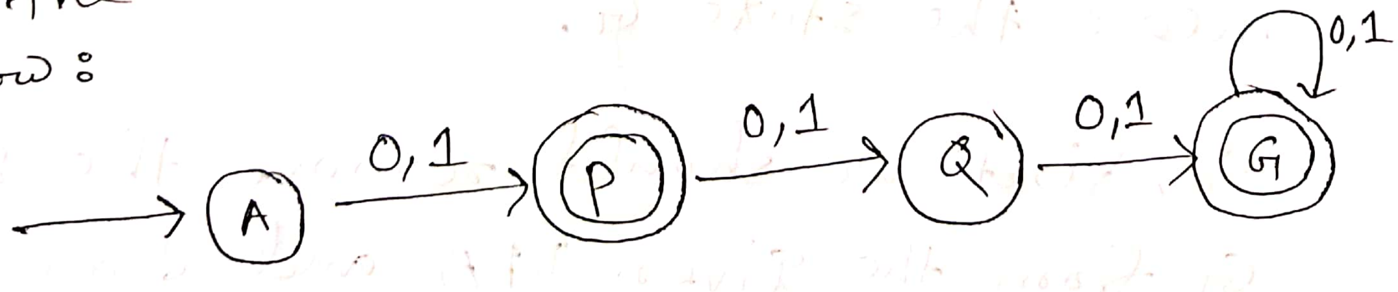
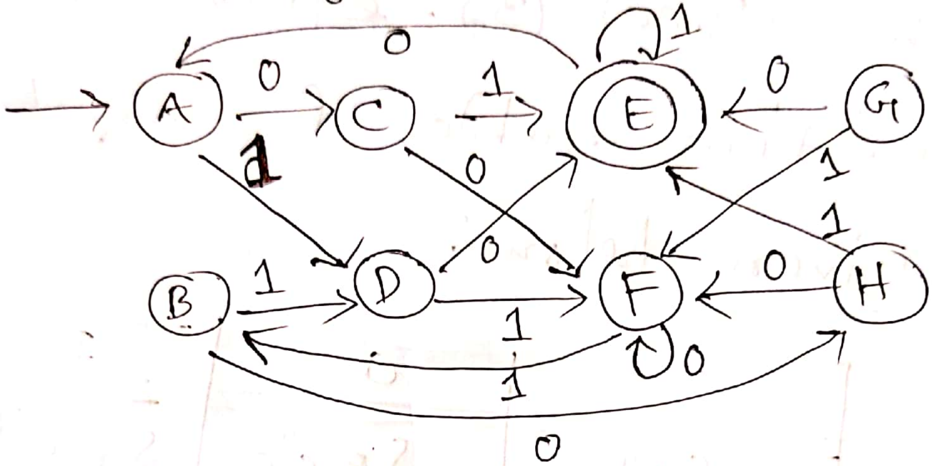


Fig: Minimal version of DFA_1

Homework-9 (Slide No-44)

Qst^n: Using table filling method or Myhill-Nerode th^m minimize or computing equivalent states in the given DFA.



Sol^n: Step-1: Draw a table for all pairs of states (P, Q) where not exist unreachable state. From above diagram we see that G is the only unreachable state. Because from the initial state A ~~it has no path to~~ there has no way to reach near the state G.

So, first we should remove the state G from the given DFA and draw a table with states A, C, E, B, D, F and H.

After removing unreachable state G, we get the state diagram of mentioned DFA is drawn below:

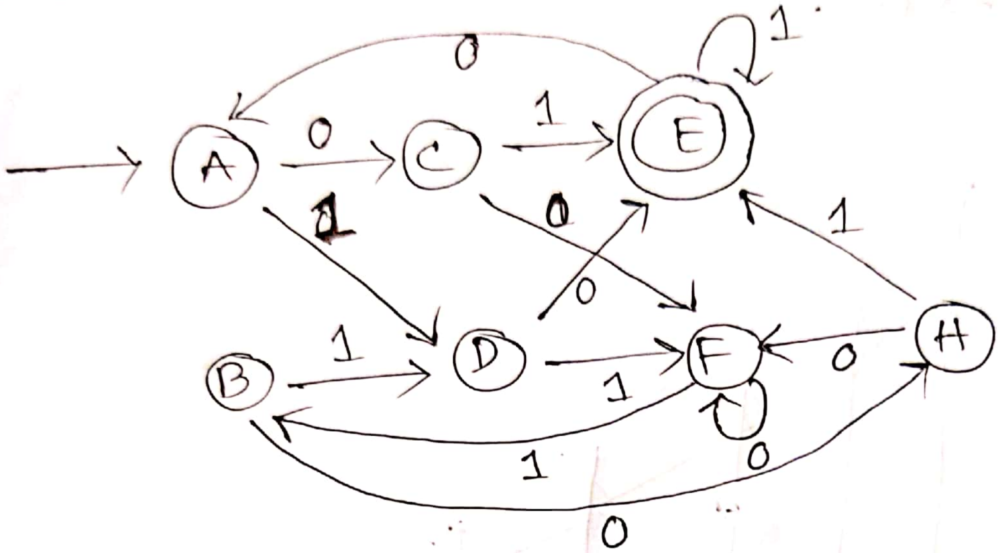


Fig: A

	A	B	C	D	E	F	H
A							
B							
C							
D							
E							
F							
H							

Fig: Table-1

From the table when we check sells of any pairs then we found 2 selfs for each pairs. So, we need to avoid the right half of the table.

Step-2: Marks all pair where $P \in F$ and $Q \notin F$ or, $P \notin F$ and $Q \in F$. Here, F means the final state.

A							
B							
C							
D							
→ E	X	X	X	X			
F					X		
H					X		
	A	B	C	D	E ↑	F	H

First operation

Fig: Table-2

According to the conditions, when ~~Q~~ a pair of states P and Q contains different state means one is final state and the other one is non-final state then we need to mark by (X) of that position into the table that's show above by first operation.

Step-3: If there are any unmarked pairs (P, Q) such that $[\delta(P, x), \delta(Q, x)]$ is marked, then mark [P, Q] where 'x' is an ~~output~~ input symbol. Repeat this until no more markings can be made.

From the step-2 table first check with pair (A, B) because it's the first unmark pair. We should also follow the state diagram Fig-A.

$\boxed{\text{check-1}}$

$(A, B) \Rightarrow \left.\begin{array}{l}\delta(A, 0) = C \\ \delta(B, 0) = H\end{array}\right\}$ new pair (C, H) that's already unmarked.

$\Rightarrow \left.\begin{array}{l}\delta(A, 1) = D \\ \delta(B, 1) = D\end{array}\right\}$ new pair (D, D) that self or pair position doesn't exist into the table.

$\boxed{\text{Check-2}}$

$(A,C) \Rightarrow \left.\begin{array}{l}\delta(A,0)=C\\ \delta(C,0)=F\end{array}\right\}$ new pair (C,F) that's already unmarked.

$\Rightarrow \left.\begin{array}{l}\delta(A,1)=D\\ \delta(C,1)=E\end{array}\right\}$

Here new pair (D,E) is already marked into the table. So, according to the step-3 rule now, we need to marked the position of pair (A,C) and the table-2 will be updated that I drawn after finishing all checking process.

$\boxed{\text{check-3}}$

$(B,C) \Rightarrow \left.\begin{array}{l}\delta(B,0)=H\\ \delta(C,0)=F\end{array}\right\}$ ~~Doesn't exist~~ Already unmarked

$\Rightarrow \left.\begin{array}{l}\delta(B,1)=D\\ \delta(C,1)=E\end{array}\right\}$ The position of pair (B,C) updated by marking (X).

$\boxed{\text{check-4}}$

(A, D) $\Rightarrow$ $\left.\begin{array}{l}\delta(A,0) = C \\ \delta(D,0) = E\end{array}\right\}$ The position of the pair (A, D) updated by marking (x).

$\Rightarrow$ $\left.\begin{array}{l}\delta(A,1) = D \\ \delta(D,1) = F\end{array}\right\}$ Already unmarked

$\boxed{\text{check-5}}$

(B, D) $\Rightarrow$ $\left.\begin{array}{l}\delta(B,0) = H \\ \delta(D,0) = E\end{array}\right\}$ The pair (B, D) updated ~~Doesn't exist~~ by marking (x)

$\Rightarrow$ $\left.\begin{array}{l}\delta(B,1) = D \\ \delta(D,1) = F\end{array}\right\}$ ~~Doesn't exist~~ Already unmarked

$\boxed{\text{check-6}}$

(C, D) $\Rightarrow$ $\left.\begin{array}{l}\delta(C,0) = F \\ \delta(D,0) = E\end{array}\right\}$ (C, D) position updated

$\Rightarrow$ $\left.\begin{array}{l}\delta(C,1) = E \\ \delta(D,1) = F\end{array}\right\}$ same

$\boxed{\text{check-7}}$

(A, F) $\Rightarrow$ $\left.\begin{array}{l}\delta(A,0) = C \\ \delta(F,0) = F\end{array}\right\}$ Already unmarked

$\Rightarrow$ $\left.\begin{array}{l}\delta(A,1) = D \\ \delta(F,1) = B\end{array}\right\}$ (A, F) position updated

$\boxed{\text{check-8}}$

$(B, F) \Rightarrow \left.\begin{array}{l} \delta(B, 0) = H \\ \delta(F, 0) = F \end{array}\right\}$ Already unmarked

$\Rightarrow \left.\begin{array}{l} \delta(B, 1) = D \\ \delta(F, 1) = B \end{array}\right\}$ (B, F) position updated

$\boxed{\text{check-9}}$

$(C, F) \Rightarrow \left.\begin{array}{l} \delta(C, 0) = F \\ \delta(F, 0) = F \end{array}\right\}$ Doesn't exist

$\Rightarrow \left.\begin{array}{l} \delta(C, 1) = E \\ \delta(F, 1) = B \end{array}\right\}$ (C, F) position updated

$\boxed{\text{check-10}}$

$(D, F) \Rightarrow \left.\begin{array}{l} \delta(D, 0) = E \\ \delta(F, 0) = F \end{array}\right\}$ (D, F) position updated

$\Rightarrow \left.\begin{array}{l} \delta(D, 1) = F \\ \delta(F, 1) = B \end{array}\right\}$ same

$\boxed{\text{check-11}}$

$(A, H) \Rightarrow \left.\begin{array}{l} \delta(A, 0) = C \\ \delta(H, 0) = F \end{array}\right\}$ (A, H) position updated

$\Rightarrow \left.\begin{array}{l} \delta(A, 1) = D \\ \delta(H, 1) = E \end{array}\right\}$ same

$\boxed{\text{check-12}}$

$(B, H) \Rightarrow \left.\begin{array}{l} \delta(B, 0) = H \\ \delta(H, 0) = F \end{array}\right\}$ ~~Doesn't exist~~ (B, H) position updated

$\Rightarrow \left.\begin{array}{l} \delta(B, 1) = D \\ \delta(H, 1) = E \end{array}\right\}$ (B, H) position updated

$\boxed{\text{check-13}}$

$(C, H) \Rightarrow \left.\begin{array}{l} \delta(C, 0) = F \\ \delta(H, 0) = F \end{array}\right\}$ Doesn't exist

$\Rightarrow \left.\begin{array}{l} \delta(C, 1) = E \\ \delta(H, 1) = E \end{array}\right\}$ Doesn't exist

$\boxed{\text{check-14}}$

$(D, H) \Rightarrow \left.\begin{array}{l} \delta(D, 0) = E \\ \delta(H, 0) = F \end{array}\right\}$ (D, H) position updated

$\Rightarrow \left.\begin{array}{l} \delta(D, 1) = F \\ \delta(H, 1) = E \end{array}\right\}$ same

$\boxed{\text{check-15}}$

$(F, H) \Rightarrow \left.\begin{array}{l} \delta(F, 0) = F \\ \delta(H, 0) = F \end{array}\right\}$ Doesn't exist

$\Rightarrow \left.\begin{array}{l} \delta(F, 1) = B \\ \delta(H, 1) = E \end{array}\right\}$ (F, H) position updated.

After checking all the unmarked pair positions from table-2, we get some new pairs that we need to updated or marked by (X) into the table that's drawn below:

	A	B	C	D	E	F	H
A							
B							
C	X	X					
D	X	X	X				
E	X	X	X	X			
F	X	X	X	X	X		
H	X	X		X	X	X	

Table-3

Step-4: Combine all the unmarked pairs and make them a single state in the minimized DFA. So from the above table (A, B) and (C,H) are the single state.

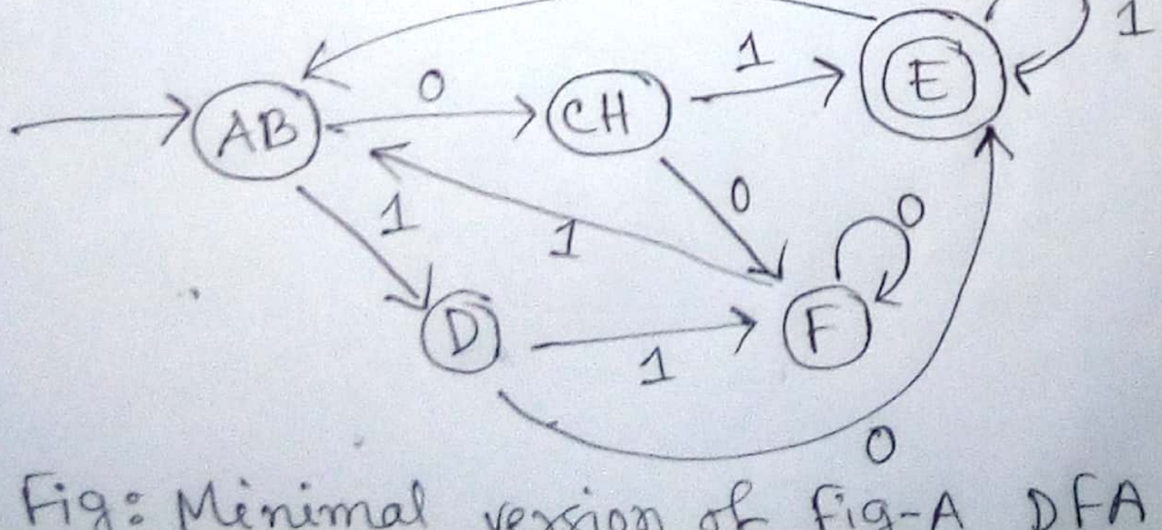


Fig: Minimal version of Fig-A DFA