Pumping lemma for regular language:-

⇒ Pumping lemma is used to determine the class of language of finite automata. i.e. it is used to determine a language is not regular.

Statement:-

Let 'L' be a regular Language and a string $w \in L$. Let 'm' be the total no. of states of FA and 'n' be the Length of the string such that $m \leq n$.

Then, w can be decomposed into substrings x, y, z Such that,

① $y \neq \epsilon$ decompose गर्दा null हुन पाइँदैन।

② $|xy| \leq n$

③ $xy^iz \in L$ for all $i \geq 0$

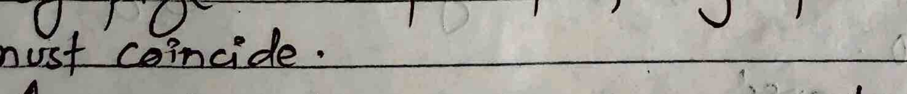
proof:-

Let L be a regular language and $w \in L$. ~~let~~

Let $w = a_1 a_2 a_3 ----- a_n$ and the set of states are $q_1, q_2, q_3 ----- q_m$.

~~let w = xyz belongs to~~ let $w = xyz \in L$

Now, its finite automata is



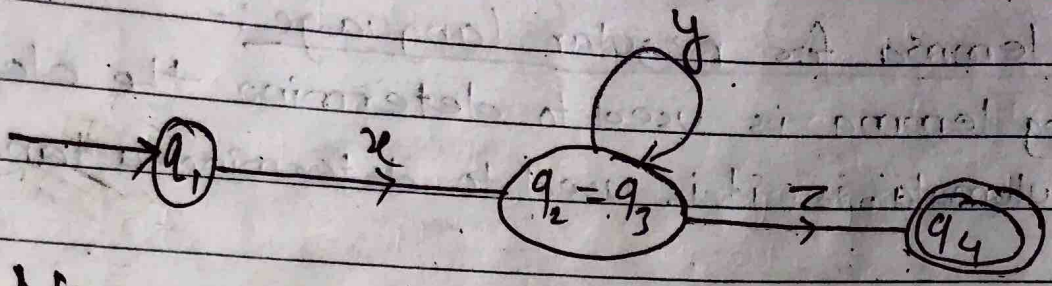
here, $|w| = 3$, but total states are 4.

By pigeon hole principle, any of the two states of FA must coincide.

According to our requirement state q_2 & q_3 coincide.

Date: Page:

ie.



Now in general form :-

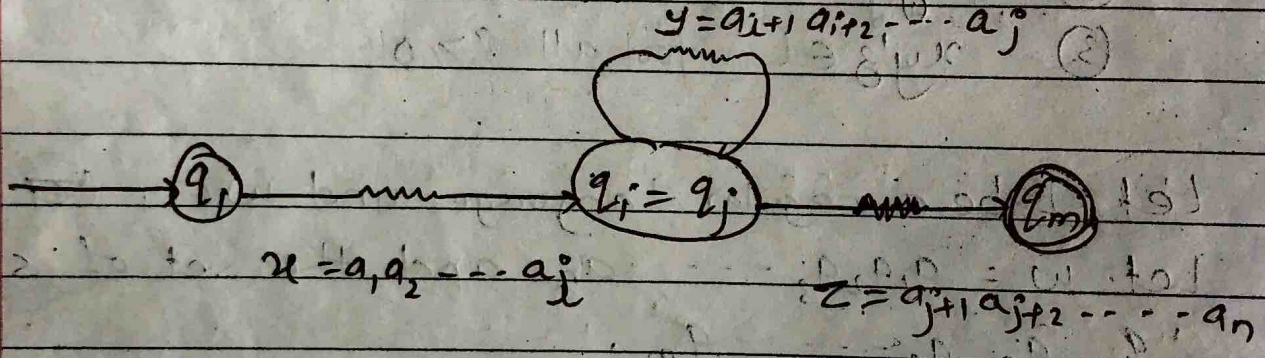
$w = a_1 a_2 a_3 \dots a_n$ can be decomposed as,

$x = a_1 a_2 a_3 \dots a_i$

$y = a_{i+1} a_{i+2} a_{i+3} \dots a_j$

$z = a_{j+1} a_{j+2} \dots a_n$

Now our FA becomes,



Now we have to show that $xy^iz \in L$.

for $i \geq 0$

When $i=0$, we get $w = xz$ which is processed by FA.

When $i=1$, we get $w = xyz$, it also accepted by FA.

Similarly, FA processes the strings for all the values of i. Hence, we say that $xy^iz \in L$ for all $i \geq 0$.

proved

Date: Page:

example:-

Show that the language $L = \{a^n b^n : n>0\}$ is not regular.

⇒ Let L be a regular Language and $w \in L$. (proof by contradiction)

let $w = a^p b^p \quad p \geq 1$

Now,

Using pumping lemma, w can be decomposed into x, y and z as,

$x = a^q$

$y = a^r, \quad r>0$

$z = a^{p-(q+r)} b^p$

(to simplicity we do only of 'a' not of 'b')

Now,

$xy^2z = a^q (a^r)^2 a^{p-(q+r)} b^p$ (Let pumping factor i=2)

$= a^{q+2r+p-q-r} b^p$

$= a^{p+r} b^p$

∵ $r>0$, $p+r > p$

Therefore, $a^{p+r} b^p$ is not of the form $a^p b^p$.

i.e. $xy^2z \notin L$

Hence, the given language is not regular by contradiction.

Proof:-

$w = a^p b^p = \underbrace{aa}_{x}\underbrace{aaa}_{y}\underbrace{bbbbb}_{z} \quad p=5$

$x = a^q = aa \quad q=2$

$y = a^r = aaa \quad r=3$

$z = a^{5-(2+3)} b^5$

$= a^0 b^5 = bbbbb$

$|xy| \leq n$

$5 \leq 10$

$\therefore xy^iz = xy^2z = aa\,aaaaa\,bbbbb$

$a^8 b^5 \neq a^5 b^5$

Adesh pandey

$L = \{0^n \mid n \text{ is a perfect square}\} \Rightarrow 0^{p^2}$

$\{ \text{---}\ ,, \text{---}\ \text{perfect cube}\} \Rightarrow 0^{p^3}$

$L = \{0^k\}\ k \text{ is prime number}\} \Rightarrow 1, 2, 3, 5, 7, 11 \ldots$

Date: Page:

Show that the Language $\{L = 1^{n^2} : n > 0\}$ is not regular.

$\Rightarrow$ Let L be a regular Language and $w \in L$.

let $w = 1^{p^2}$ $p \geq 1$

now,

Using pumping lemma, w can be decomposed into x, y and z as,

$x = 1^{q^2}$

$y = 1^{r^2}, r > 0$

$z = 1^{p^2 - (q^2 + r^2)}$

now, let pumping factor is $i = 2$

$\therefore xy^2z = 1^{q^2} \cdot (1^{r^2})^2 \cdot 1^{p^2 - (q^2 + r^2)}$

$= 1^{q^2 + 2r^2 + p^2 - q^2 - r^2}$

$= 1^{p^2 + r^2}$.

$\because r > 0,\ r^2 > 0$

$\therefore p^2 + r^2 > p^2$

Therefore, $1^{p^2 + r^2}$ is not of the form 1^{p^2}

Hence $xy^2z \notin L$. and is not regular by contradiction.

Rough:

$w = 1^9 = \underset{x}{1}\ \underset{y}{1111}\ \underset{z}{1111}$ $p = 3 \therefore p^2 = 9$

$x = 1^1 = 1$ $q = 1 \therefore q = 1$

$y = 1^4 = 1111$ $r = 2$ $\therefore r^2 = 4$

$z = 1^{9 - (1 + 4)} = 1^4 = 1111$

$1^{2^2} \ 1^{4}$ ✗, $1^{3^2} \ 1^9$ ✗, $1^{4^2} \ 1^{16}$ ✗

$\therefore xy^iz = xy^2z = \underset{x}{1}\ \underset{y^2}{1111\ 1111}\ 1111 = 1^{13} \neq 1^{3^2} = 1^9$

Show that the language $\{L = xx^R ; x \in \{0,1\}\}$ is not regular.

⇒

Let L be a regular Language and $w \in L$

let $x = 1^n 0^n$

then $x^R = 0^n 1^n$

$xx^R = 1^n 0^n 0^n 1^n$

Let $w = 1^P 0^P 0^P 1^P$

now,

Using pumping lemma, w can be decomposed into x, y & z as,

$x = 1^q$

$y = 1^r$

$z = 1^{p-(q+r)} 0^P 0^P 1^P$

now,

$xy^2z = 1^q 1^{(r)2} 1^{p-(q+r)} 0^P 0^P 1^P$

$= 1^{q+2r+p-q-r} 0^P 0^P 1^P$

$= 1^{p+r} 0^P 0^P 1^P$

$\because r > 0$

$\therefore p + r > p$

Hence, $1^{p+r} 0^P 0^P 1^P$ is not of the form $1^P 0^P 0^P 1^P$

therefore $xy^2z \notin L$.

Rough

$w = 1^3 0^3 0^3 1^3$ $p = 3$

$= \underline{1}\,\underline{11}\,000000111$ (x, y)

$x = 1^1$ $q = 1$

$y = 1^2$ $r = 2$

$z = 1^{9-(2+1)} 0^3 0^3 1^3 = 1^6 0^3 0^3 1^3$

$xy^iz_{i=2} = xy^2z = \underline{1}\,\underline{1111} \cdot 111111\, 0^3 0^3 1^3$

$= 1^{11} 0^3 0^3 1^3$

decision properties:-

1. Is the language described empty? (reachable/unreachable)
2. Is the particular string w in the described language (for DFA w ends in accepting state)
3. Do the two descriptions of a language actually describe the same language? (equivalance).

Minimization of DFA:-

⇒ DFA minimization is the task of transforming a given DFA into an equivalent DFA that has a minimum no. of states. Here, two DFAs are called equivalent if they recognize the same language.

⇒ During the course of minimization, it involves - identifying the equivalent state & distinguishable states.

Equivalent states:-

- Two states P and q are called equivalent states, denoted by $P \equiv q$ if and only if for each input string x, $\hat{\delta}(P,x)$ is a final state & iff $\hat{\delta}(q,x)$ is a final state.

Distinguishable states:-

- Two states P and q are said to be distinguishable state if there exists a string x, such that $\hat{\delta}(P,x)$ is a final state, $\hat{\delta}(q,x)$ is a non final state.

Table filling algorithm (Myhill nerode algorithm)

⇒ steps:-

1) Draw a table for all pairs of states (P,Q)
2) Mark all pairs where $P \in F$ and $Q \notin F$.
3) If there are any unmarked pairs (P,Q)

Date:________ Page:________

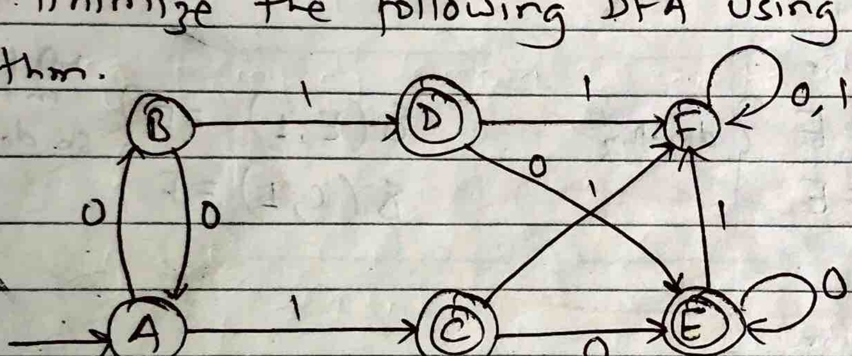
such that $[\delta(p, x), \delta(q, x)]$ is marked, then mark $[p, q]$ where x is an input symbol.
Repeat this until no more markings can be made.

4) Combine all the unmarked pairs and make them a single state in the minimized DFA.

⟹ ~~(1, 2) steps~~

~~[illegible]~~

Q. Minimize the following DFA using table filling algorithm.



Draw a table as,

Step 1 & 2

	A	B	C	D	E	F
A						
B	AB					
C	✓	✓				
D	✓	✓				
E	✓	✓				
F	✓	✓	✓	✓	✓	

(C, A) ∵ C is a final & A is non final ∴ so on

(F, A) both non final ∴ unmark in first iteration but might be marked in later iteration or next step.

do serially for all pairs in table

Date: Page:

step 3: (1st iteration)

for (B, A) pair

$\delta(B,0) = A$ } pair if ~~A is marked~~ if not marked do nothing
$\delta(A,0) = B$ }

$\delta(A,1) = C$ } unmarked do nothing
$\delta(B,1) = D$ }

for (D, C) pair

$\delta(D,0) = E$ } do nothing
$\delta(C,0) = E$ }

$\delta(D,1) = F$ } do nothing
$\delta(C,1) = F$ }

for (E, C) pair

$\delta(E,0) = E$ } do nothing
$\delta(C,0) = E$ }

$\delta(E,1) = F$ } not present on table so do nothing
$\delta(C,1) = F$ }

for (E, D) pair

$\delta(E,0) = E$ }
$\delta(D,0) = E$ }

$\delta(E,1) = F$ }
$\delta(D,1) = F$ }

for (F, A) pair

$\delta(F,0) = F$ } not marked previously
$\delta(A,0) = B$ }

$\delta(F,1) = F$ } check
$\delta(A,1) = C$ } yes (marked)

now mark F, A also

for (F, B) pair

$\delta(F,0) = F$ } marked
$\delta(B,0) = A$ } so mark (F, B) also

$\delta(F,1) = F$ } marked
$\delta(B,1) = D$ } :)

Date: Page:

- No more new markings, so end the process.

⇒ 2nd iteration (do yourself)

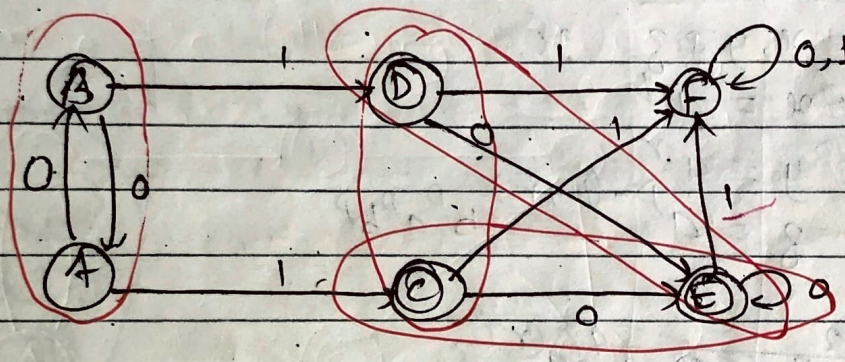
also in next iteration

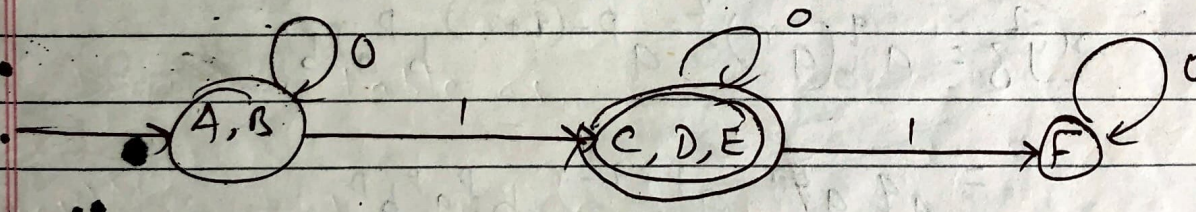
Step 4:

Combine all unmarked pair & make single state

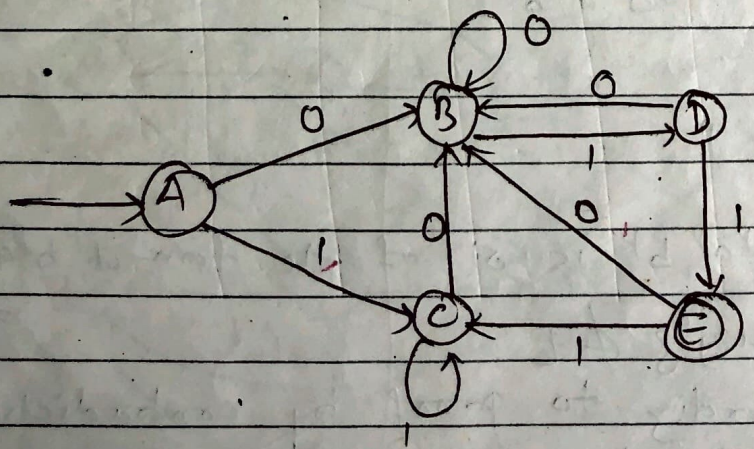
(A,B) (D,C) (E,C) (E,D) → unmarked pair

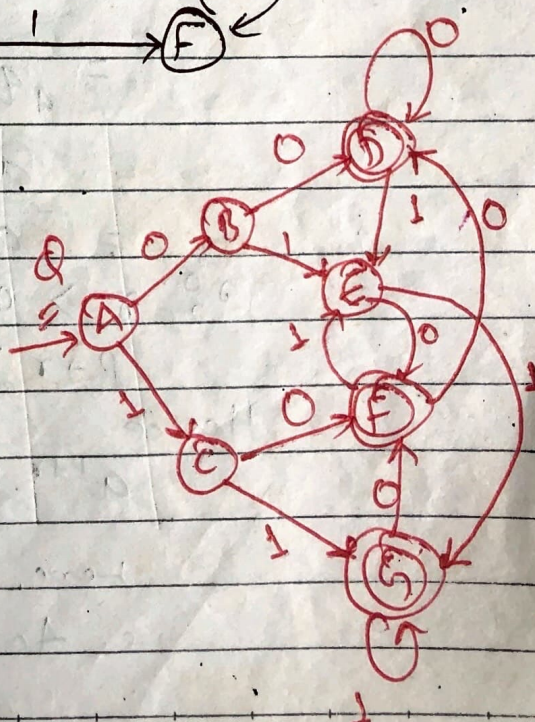
Original DFA is,

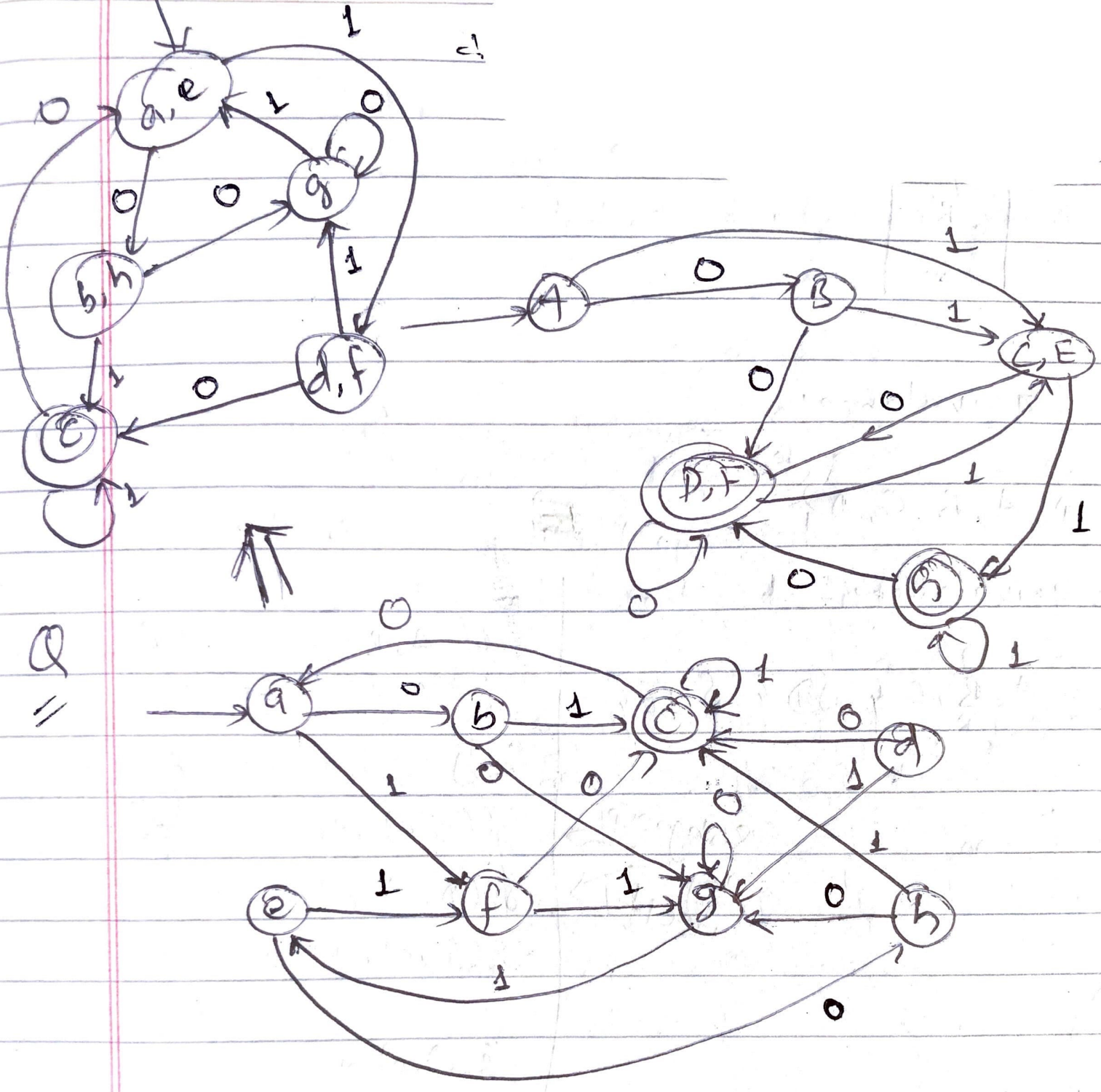




final minimized DFA.






a, e
g
b, h
d, f
c
A
B
C, E
D, F
G
a
b
c
d
e
f
g
h