Question | **Is more, more or less**

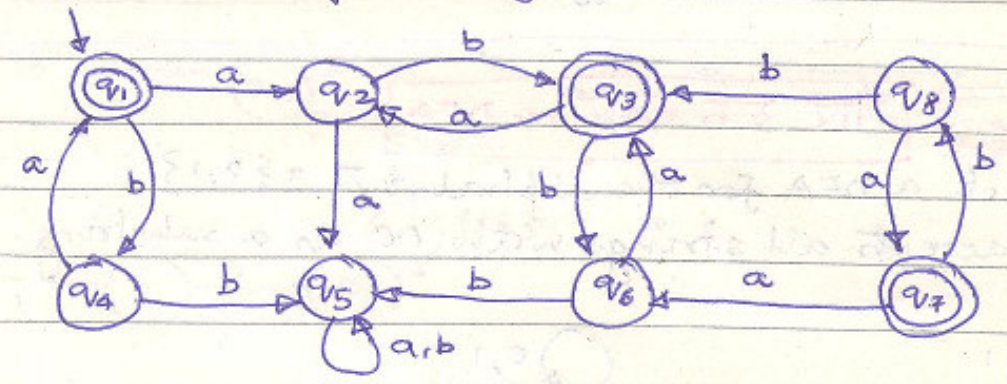
Consider the DFA $M = \{Q, \Sigma, \{q_1\}, \delta, F\}$ where all symbols have their 'usual' meanings.

$Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$

$F = \{q_1, q_3, q_7\}$ and δ is given by the table to the right. Using this info, construct an equivalent minimum-state DFA.

State	a	b
q_1	q_2	q_4
q_2	q_5	q_3
q_3	q_2	q_6
q_4	q_1	q_5
q_5	q_5	q_5
q_6	q_3	q_5
q_7	q_6	q_8
q_8	q_7	q_3

'Drawn nicely', the given DFA looks like:



Following the k-equivalence method,

$P_0 = \{\{q_1, q_3, q_7\}, \{q_2, q_4, q_5, q_6, q_8\}\}$

c_0 c_1

$x=a$	c_1	c_1	c_1	c_1	c_0	c_1	c_0	c_0
$x=b$	c_1	c_1	c_1	c_0	c_1	c_1	c_1	c_0
	c_0			c_2	c_3	c_4		c_5

$P_1 = \{\{q_1, q_3, q_7\}, \{q_2\}, \{q_4, q_6\}, \{q_5\}, \{q_8\}\}$

c_0 c_2 c_3 c_4 c_5

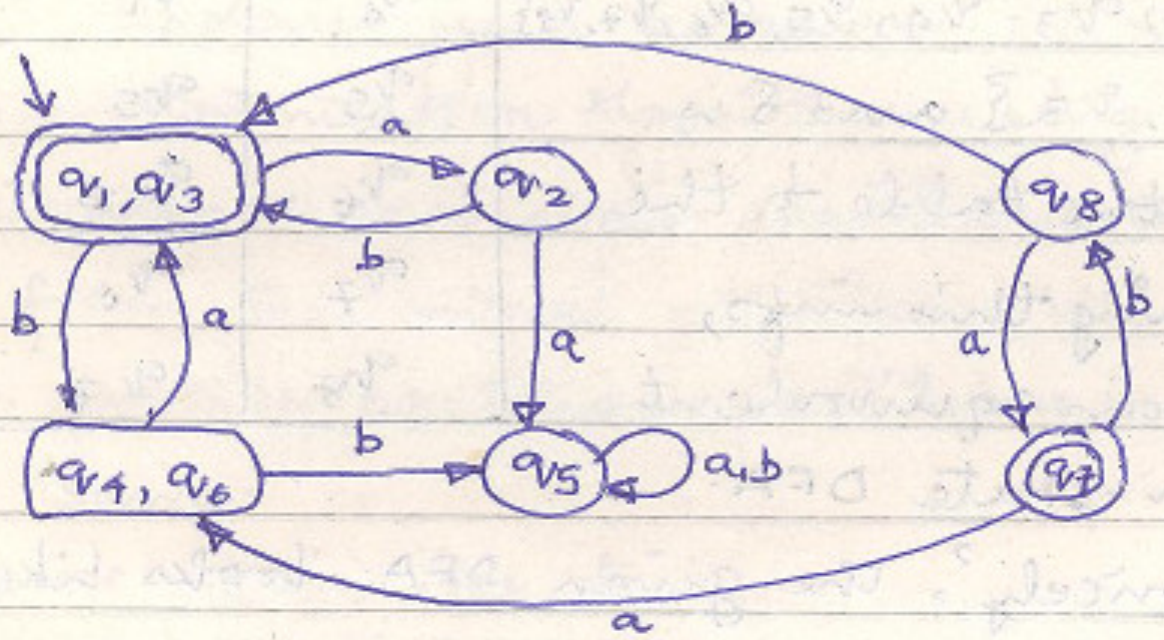
$x=a$	c_2	c_2	c_3	c_0	c_0
$x=b$	c_3	c_3	c_5	c_4	c_4
	c_6		c_7	c_3	

$P_2 = \{\{q_1, q_5\}, \{q_7\}, \{q_2\}, \{q_4, q_6\}, \{q_5\}, \{q_8\}\}$

C_6 C_7 C_2 C_3 C_4 C_5

$x=a$	C_2	C_2	C_6	C_6
$x=b$	C_3	C_3	C_4	C_4
	C_6		C_3	

$P_3 = P_2 \Longrightarrow$ stop!



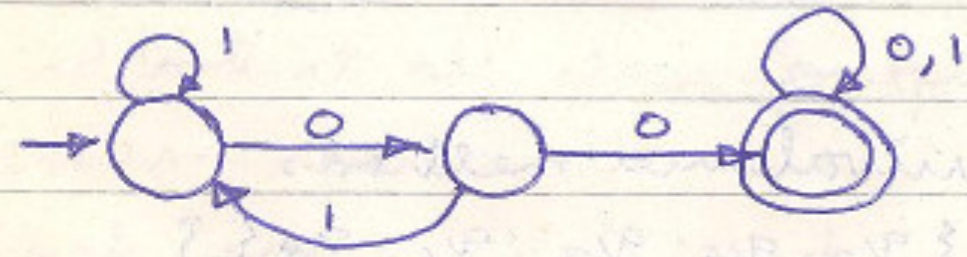
```
ALGORITHM FSM_matcher
q := 0; compute_δ();
FOR i := 1 TO n DO /*O(n)*/
   q := δ(q, T[i]);
   IF q == m THEN print shift=i−m;

ALGORITHM compute_δ()
FOR q := 0 TO m DO /*O(m)*/
FOREACH a ∈ Σ DO /*O(|Σ|)*/
   k := min(m+1, q+2); /*no jump-off*/
   REPEAT k := k−1; /*≤ O(m) times*/
   UNTIL P[1..k] is a suffix of P[1..q]a
   /*≤ m character comparisons*/
```

- FSM: $\mathcal{O}(n+m^3|\Sigma|)$

Question **फिर एक द.फ.आ... DFA**

construct a DFA for the alphabet $\Sigma = \{0, 1\}$ which accepts all strings with 00 as a substring.



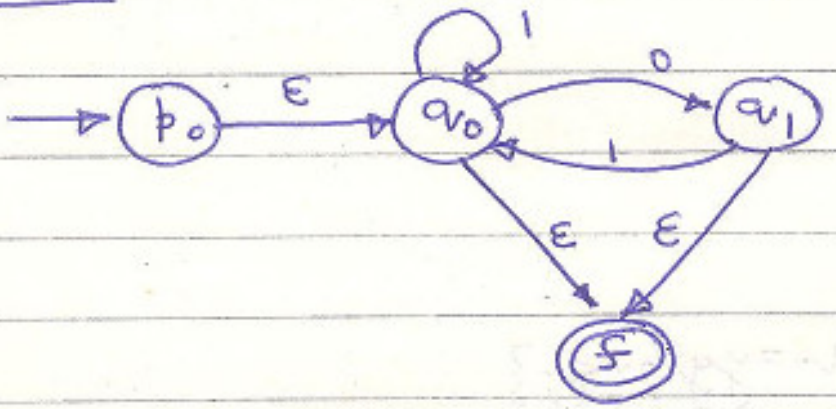
Question | Even the Introvert must engage in Regular Expression

Consider a Language over $\{0, 1\}$: all strings with 00 as a substring. construct a regular expression for the complement of this language.

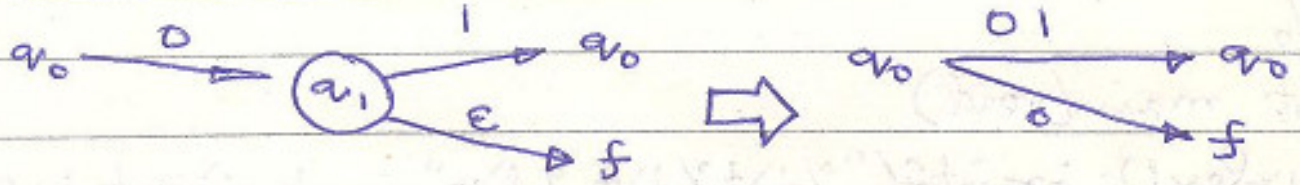
It is easy to go via a DFA, and the state minimisation method.

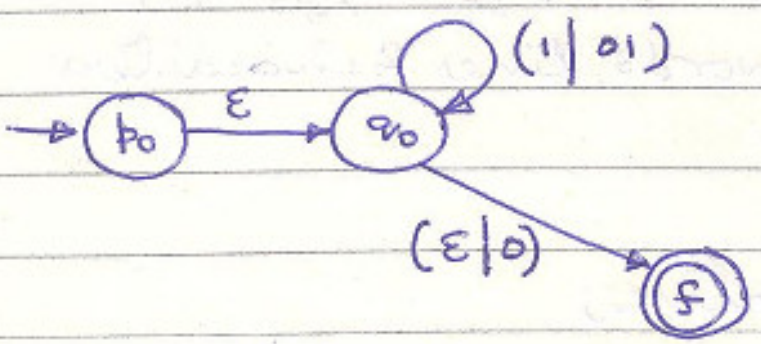
Step 1: Eliminate trap states.

Step 2: New start state, new final state

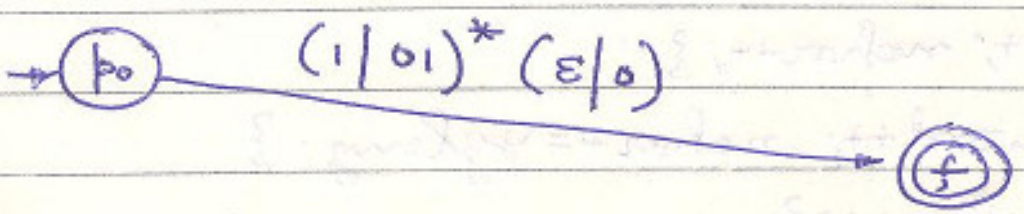


Eliminate q_1:





Eliminate q_0



Question | one for the Smart A-lex.../*No Comments*/

What precisely does the following lex code print out, given a few lines of text input, terminated by a Control-D ? There will be no marks for an imprecise answer. It is a digital world, you know!

```
%{
int i,j,k;
%}
%%
\n {k++; i++;}
[^ \n\t] {j++; i+=yyleng;}
.        {i++;}
%%
int main(void)
{yylex(); printf("%d\t%d\t%d\n",j, k, i); return 0;}
```

This counts the no. of words, lines & characters.

Better written code:-

```
%{
int nchar, nword, nline;
%}
%%
\n {nline++; nchar++;}
[^ \n\t] {nword++; nchar+=yyleng;}
.        {nchar++;}
%%
int main(void)
{yylex(); printf("%d\t%d\t%d\n", nchar, nword,
                                 nline); return 0;}
```