COMP335

Assignment 2

2024-02-25

Alawadi Mustafa

1. $\{w \in \{a,b,c\}: n_a(w) \geq 1$ and $n_b(w) \geq 2\}$

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

b) $e = a(aa)^*(bb)^*(cc)^* + (aa)^*b(bb)^*(cc)^* + (aa)^*(bb)^*c(cc)^* +$
$a(aa)^*b(bb)^*c(cc)^*$

2. $R = 0^* + 1^*$, $S = 01^* + 10^* + 1^*0 + (0^*1)^*$

a) $L(R) \setminus L(S)$ A string in $L(R)$ but not $L(S)$

000

b) $L(S) \setminus L(R)$ A string in $L(S)$ but not $L(R)$

01, 011, 10101, etc..

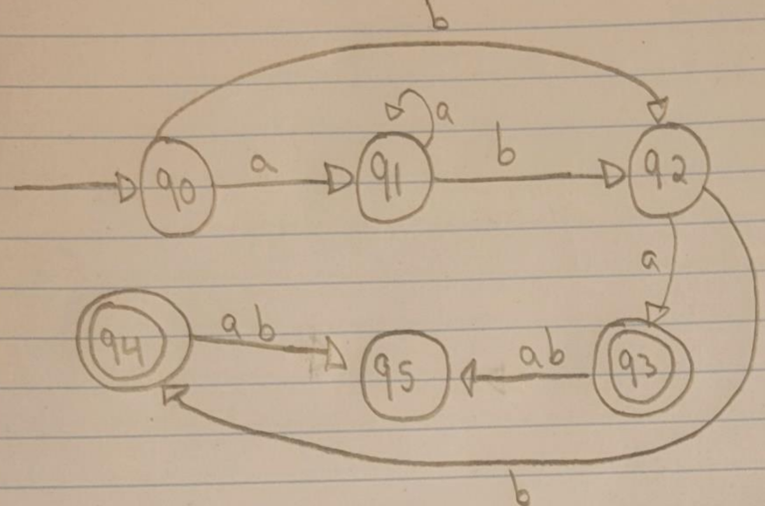
c) $L(R) \cap L(S)$ Both strings must be accepted

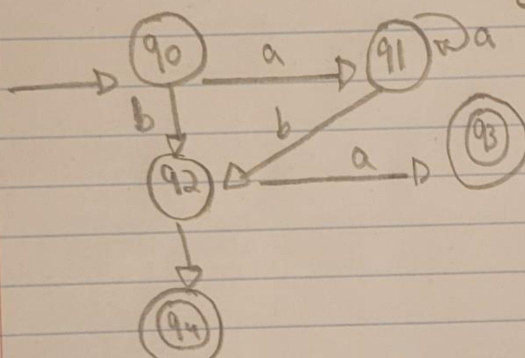
01

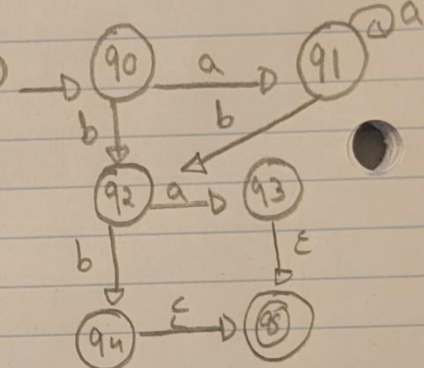
D) $\overline{L(R) \cup L(S)}$ A string not in either $L(R)$ or $L(S)$

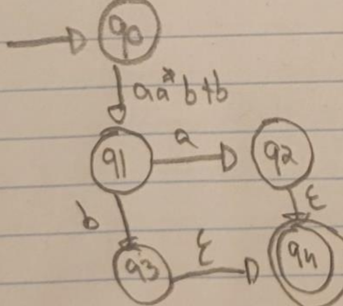
10010

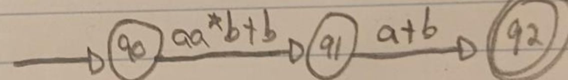
3. DFA A

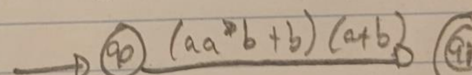


1)


2)


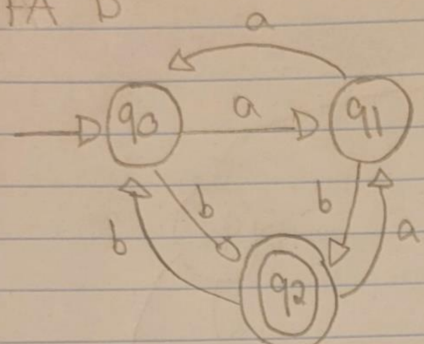
3)


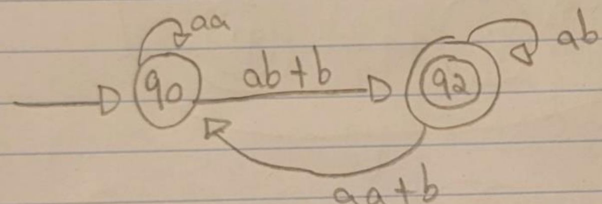
4)


5)


Thus, the regular expression is $(aa^*b+b)(a+b)$

DFA B

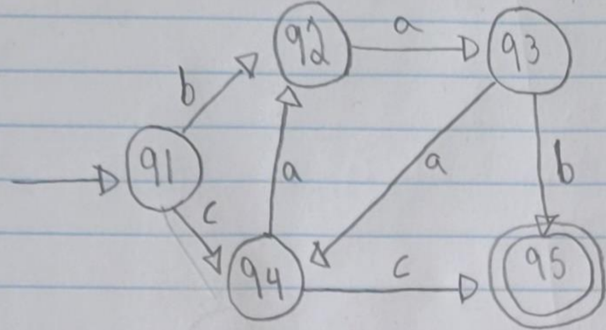




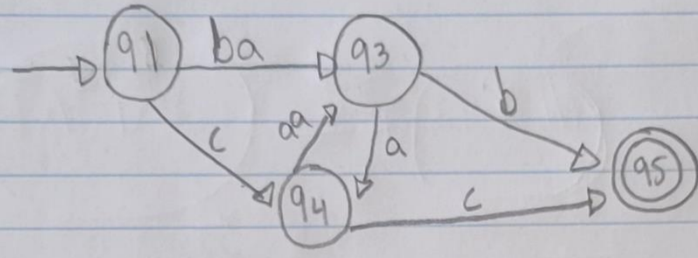
Thus, the regular expression is :

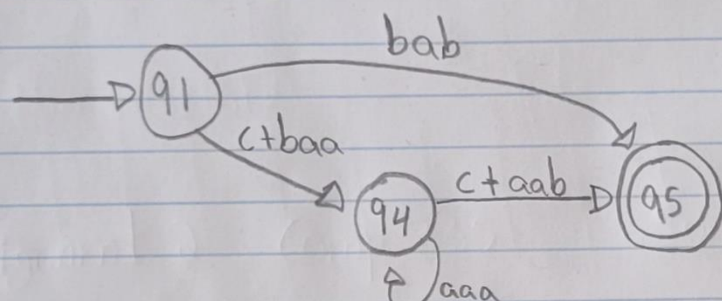
$$[aa + (ab+b)(ab)^*(aa+b)]^*(ab+b)(ab)^*$$

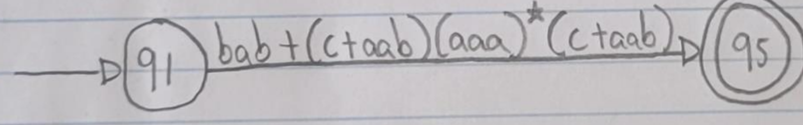
DFA C : Eliminating the trap state we get



then,



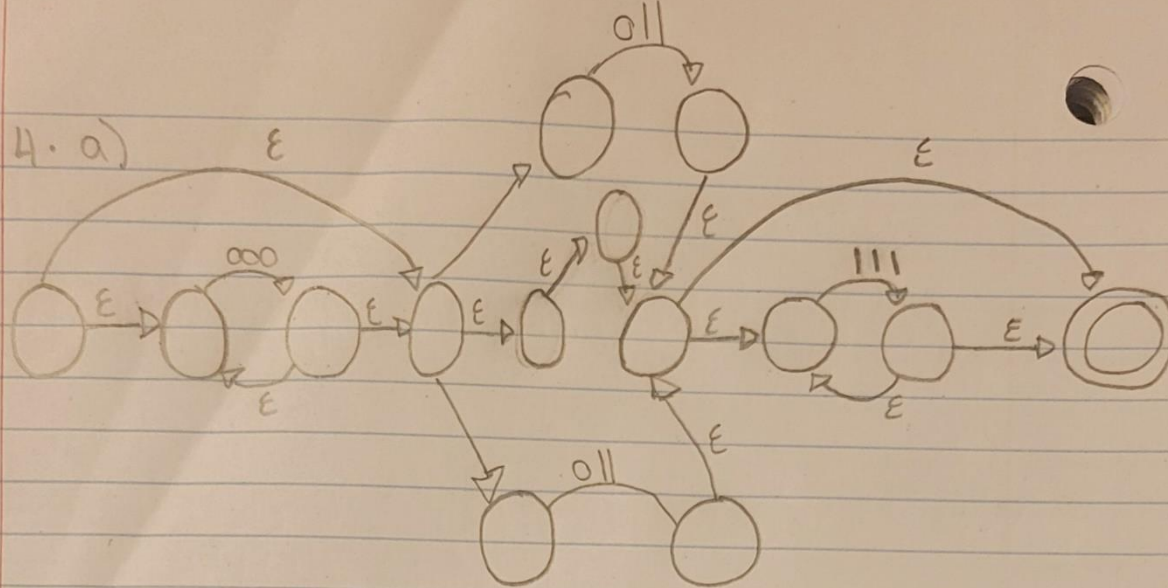




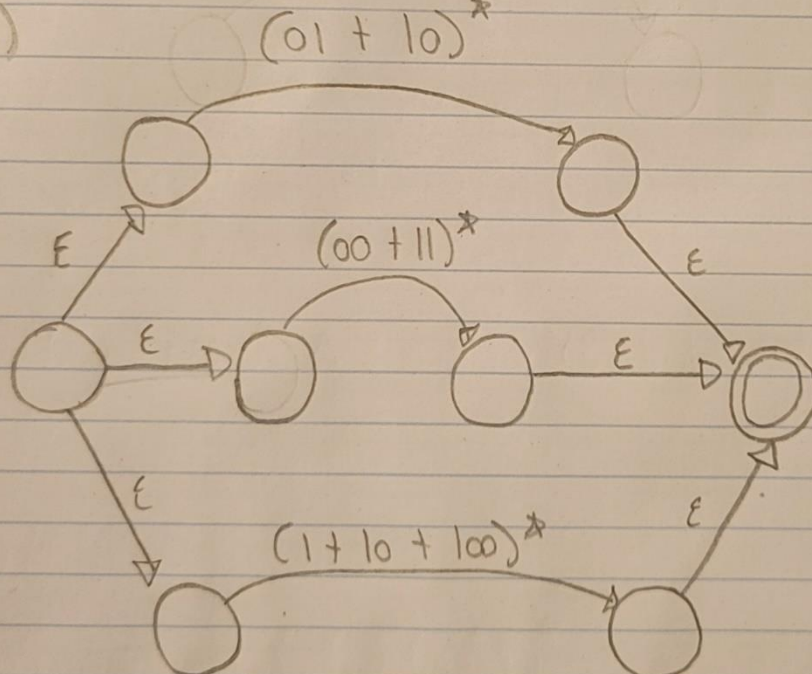
Therefore, we get the regular expression

$$= bab + (c+aab)(aaa)^*(c+aab)$$

4. a)



b)



5. a) $R(S+T) = RS + RT$

If we consider $R = \{0,1\}^*$, $S = \{0\}^*$, and $T = \{1\}^*$

Then the left side generates strings of 0's and 1's then followed by either 0 or 1. The second statement generates strings by concatinating strings from R with strings from S, or strings from R with T.

Therefore both statements generate the same strings because any strings in the first statement can be split in RS or RT.

Thus, the statements hold true.

b) $L((a+b)^*) = L((a^*b^*)^*)$, where $a, b \in \Sigma$

If $L \subseteq M$, then $L^* \subseteq M^*$, we note $(a+b) \subseteq a^*b^*$

Thus, this gives us $(a+b)^* \subseteq (a^*b^*)^*$

$$L((a+b)^*) \subseteq L((a^*b^*)^*)$$

If $(L^*)^* = L^*$

$$(a^*b^*) \subseteq (a+b)^*$$

So $(a^*b^*)^* \subseteq ((a+b)^*)^* = (a+b)^*$

Thus, $L((a^*b^*)^*) \subseteq L((a+b)^*)$

C) This statement is incorrect because if we consider the left side, it represents the closure of the union of the languages R and S which means we can concatinate 0 or more strings from either R or S. However, on the right side the statement represents the union of closures of R and S but individually. Therefore, a counterexample is if we let $R = \{a\}$ and $S = \{b\}$. Then the string "ab" is in the first statement $(R+S)^*$, however, it can't be in $R^* + S^*$ because this string is not in either R^* or S^*. Finally, the statement is disproved.

D) This statement is incorrect because the left side represents strings that can be formed by concatinating 0 or more strings from either RS or R then adding RS. However, on the right side, the statement represents strings made by concatinating 0 or more strings of RR then adding S. If we let $R = \{a\}$ and $S = \{b\}$ then the string "abb" belongs to the left side but not on right.

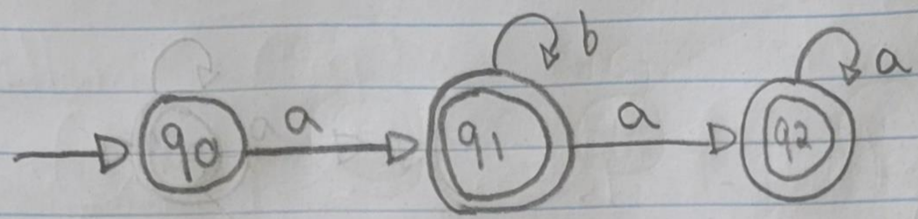
6. a) Using the proof by contradiction, we start by assuming that L is regular. By Pumping Lemma, the length is p, thus we want a string s in L for a length of p or greater so we can pump lemma. If we pick $s = 0^p 1 0^p 1$, then it's still in L and the substrings 01 and 10 are not overlapping. If we divide s in 3 parts, we get $s = xyz$, $|xy| \leq p$ and $|y| \geq 1$. We know that x and y can only have 0's because the first p symbols are 0's. If we pump y, the new string will be $xy^2z = xyyz$. Therefore, in this substring the y has been repeated and we have more 0's before we first see 01. So xy^2z has a number of non-overlapping occurences that isn't equal no more. Thus, the contradiction proved that the language is not regular.

b) Using the proof by contradiction we will assume that L is regular. Starting by applying the pumping Lemma to any string in L that is at least as long as p, which is the pumping length. Now, let $s = a^{2^n p}$ and that's a string in L. Then we split s in 3 parts, $s = xyz$, $|xy| \leq p$ and $|y| \geq 1$. If we start by pumping y, then the length of string xy^iz will not be anymore a power of 2 Since pumping y might add or remove a number of a's that's not a power of 2. Because the resulting string xy^iz is not in L, there is a contradiction.

Therefore, the contradiction proved that the language is not regular.

c) Using the proof by contradiction, we assume the language is regular. To apply the Pumping Lemma, consider p the pumping length and string $s = a^p b^p a^p$ which is in the language because it has an equal number of a's. Then s can be split in 3 parts, xyz, $|xy| \leq p$ and $|y| \geq 1$. If we consider y to have only b's, then pumping y will lead to a string where the number of b's isn't equal to number of a's. Therefore, by contradiction we have proved that the language is not regular.

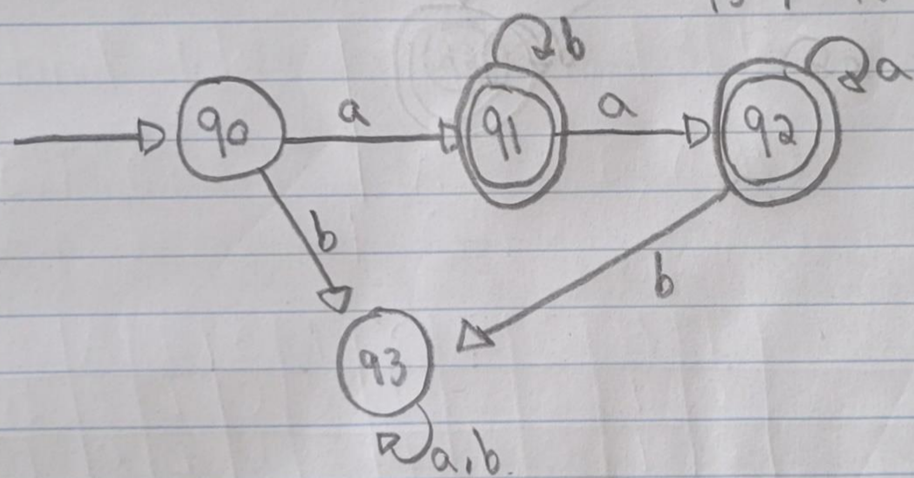
7. a) $L(ab^*a^*)$



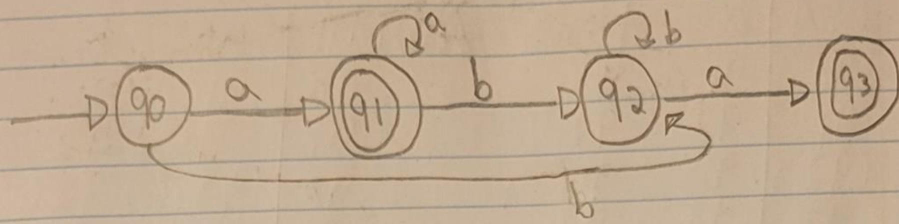
Converting to DFA we get

NFA	a	b
q_0	q_1	ϕ
q_1	q_2	q_1
q_2	q_2	ϕ

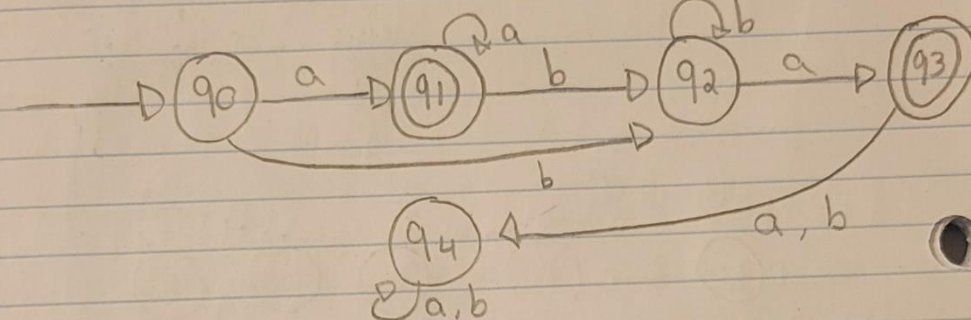
DFA	a	b
q_0	q_1	q_3
q_1	q_2	q_1
q_2	q_2	q_3
q_3	q_3	q_3



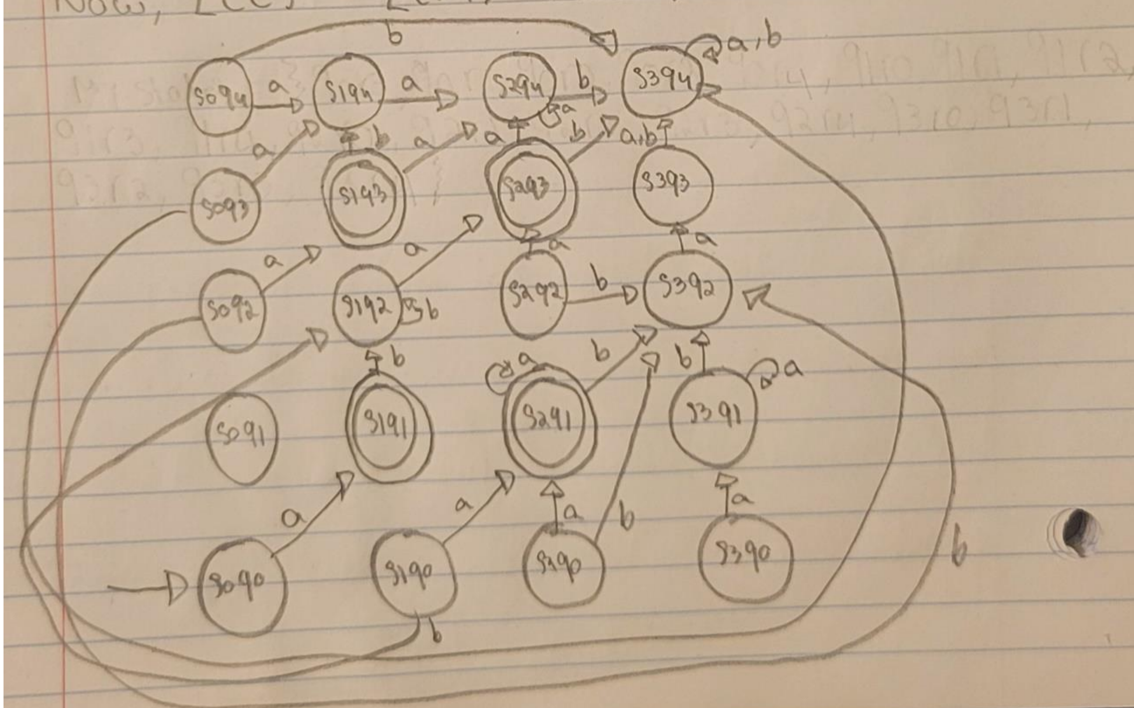
$L(a^*b^*a)$



Converting to a DFA we get:



Now, $L(C) = L(A) \cap L(B)$

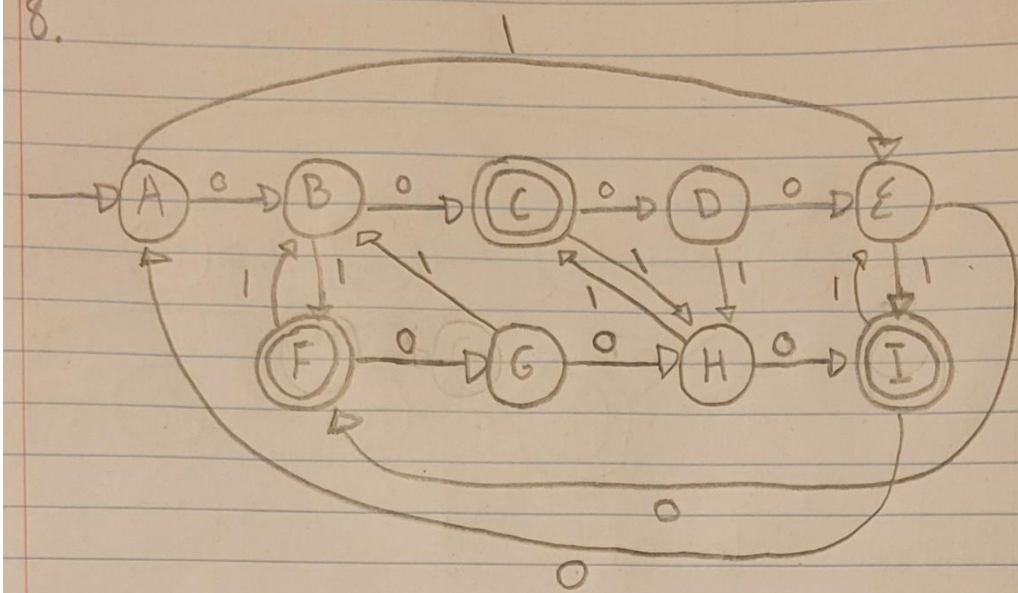


b) i) $h(21120) = ba\ bb\ ba\ a$

ii) $h(L(0+1^*2)) = a + b^*ba$

iii) $h^{-1}(L) = L(2)$

8.



Minimum State DFA:

Final state	non-Final State
{C, F, G}	{A, B, D, E, G, H}
Group 1	Group 2

	A	B	C	D	E	F	G	H	I
0	2	1	2	2	1	2	2	1	2
1	2	1	2	2	1	2	2	1	2

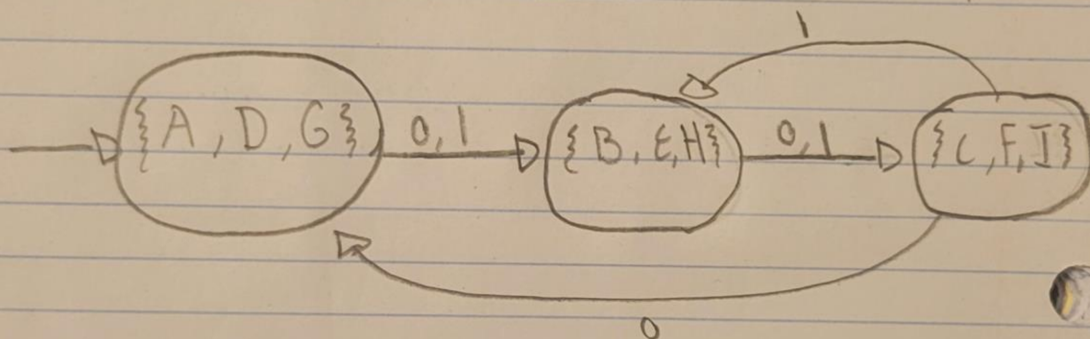
The ones equal are {A, C, D, F, G, I} and {B, E, H}
Group 3 Group 4

	A	B	C	D	E	F	G	H	I
0	4	3	3	4	3	3	4	3	3
1	4	3	4	4	3	4	4	3	4

The ones equal are {A, D, G}, {B, E, H}, {C, F, I}
Group 5 Group 6 Group 7

Thus the table is

B	X							
C	X	X						
D	O	X	X					
E	X	O	X	X				
F	X	X	O	X	X			
G	O	X	X	O	X	X		
H	X	O	X	X	O	X	X	
I	X	X	O	X	X	O	X	X
	A	B	C	D	E	F	G	H



∴ Minimal DFA is 3 states