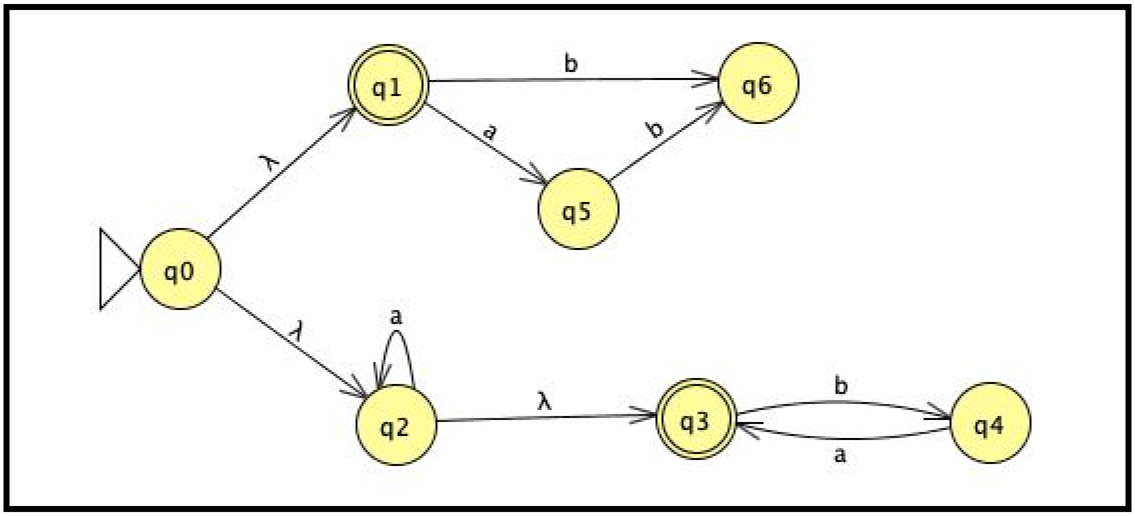
UE18CS254 - Theory of Computation

Assignment 1

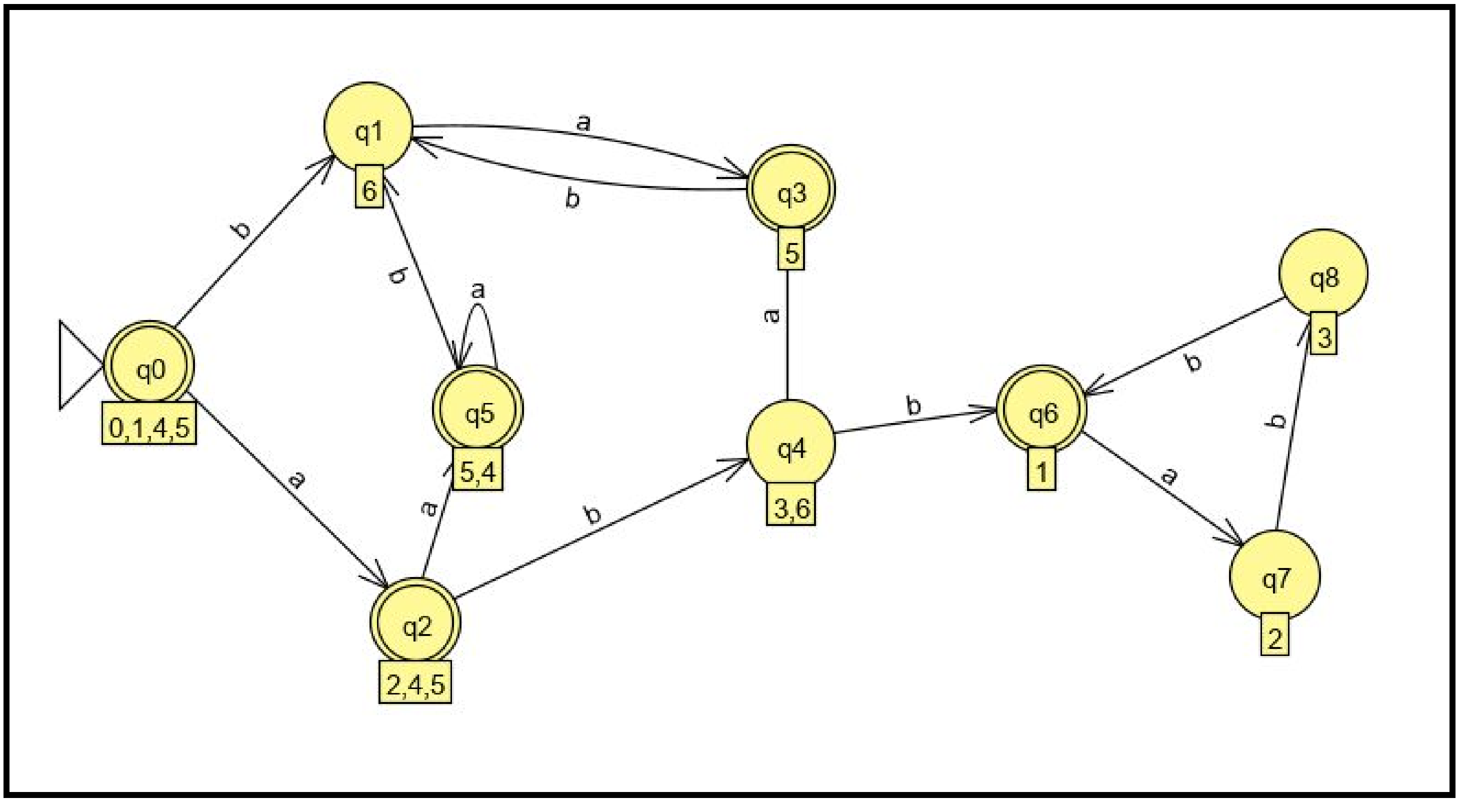
~ Tejas Srinivasan
Semester - 4 'A' section
Roll No - 5
PES1201800110

1. Given NFA -



State transition table -

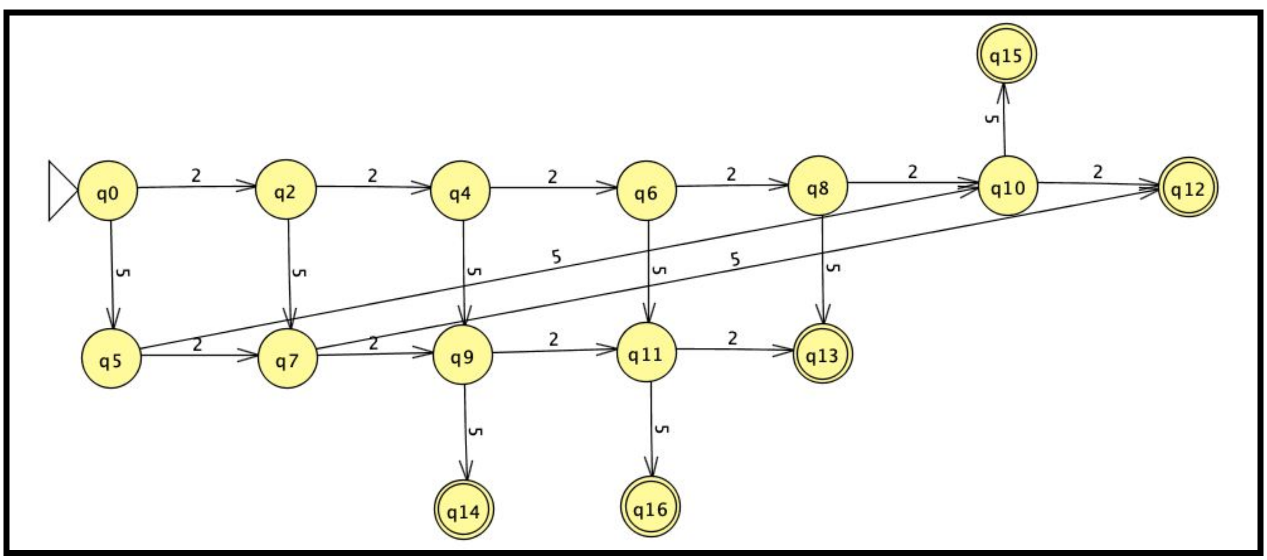
State	a	b
->*{q0, q1, q4, q5} q0	*{q3, q4, q5}	{q6}
*{q2, q4, q5} q1	*{q4, q5}	{q6, q4}
{q6} q2	*{q5}	{ }
*{q4, q5} q3	*{q4, q5}	{ }
{q6, q3} q4	*{q5}	*{q1}
*{q5} q5	{ }	{q6}
*{q1} q6	{q2}	{ }
{q2} q7	{ }	{q3}
{q3} q8	{ }	*{q1}



Converted DFA (from NFA). The labels of the states in DFA indicate the equivalent states of the NFA (i.e in q0 the label 0,1,4,5 indicates the NFA states).

2. 15 states are required in the automaton. (q0 to q16 and q1 and q3 do not exist since the vending machine does not accept 1 rupee coins)

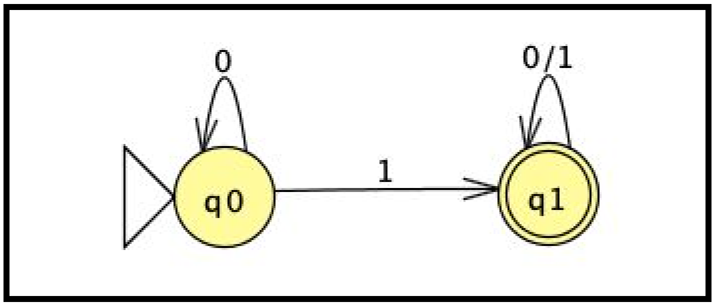
The finite automaton is given below -



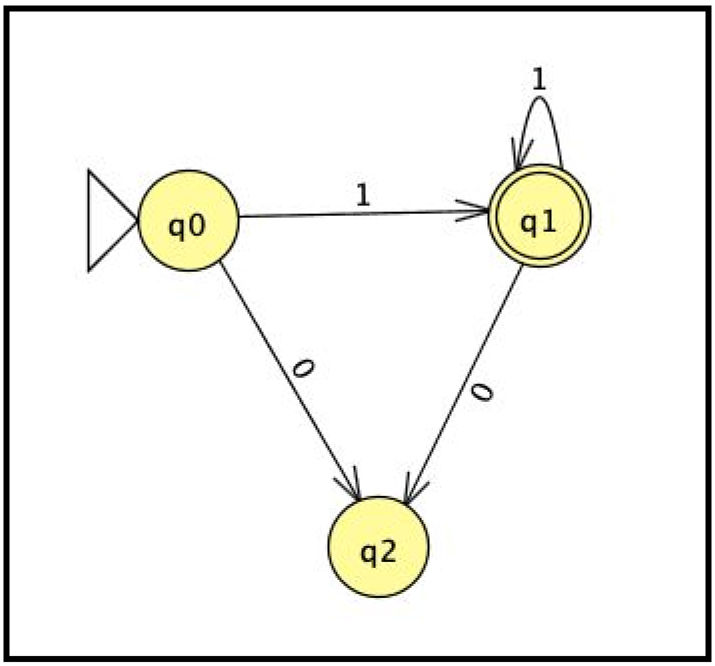
3. i) If the given transition would be introduced, it would result in a self loop over qs for all symbols in the alphabet in the language L(M).

ii) No, the automata would not remain a **DFA** anymore as the transition will now exist twice in the initial state q_s, once as a self loop over q_s and once as a transition from qs to the next state. Hence it is a **NFA.**

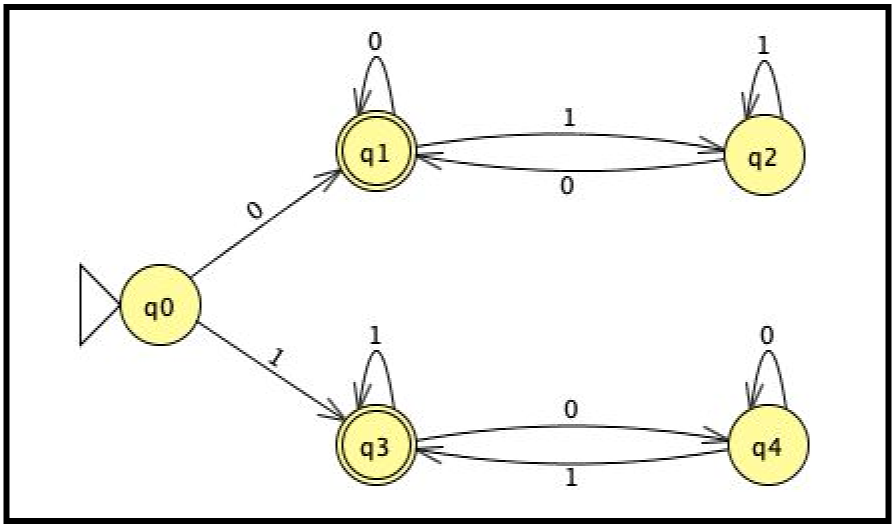
4. i) Automaton mimics OR operation -



ii) Automaton mimics AND operation -

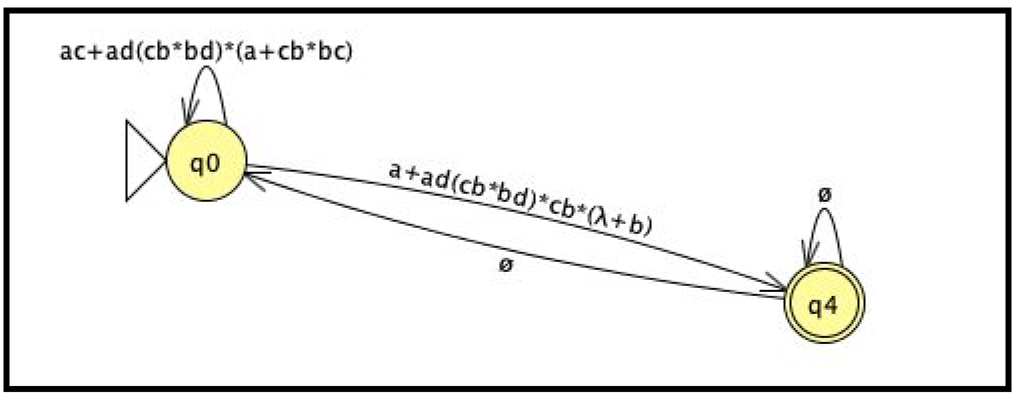


5.



Regex - $(1+0^*1^+)^* + (0+1^*0^+)^*$

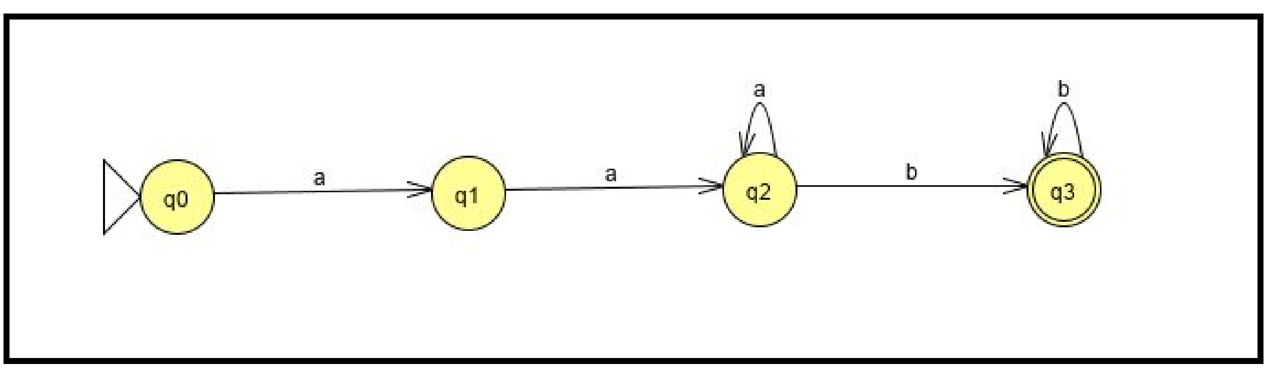
6. The automaton given has two final states. On converting it to have a single final state and after adding transitions and removing non final states we get -



The regular expression obtained is -

$(ac+ad(cb^*bd)^*(a+cb^*bc))^*(a+ad(cb^*bd)^*cb^*(\lambda+b))$

7. a)



q1	x		
q2	x	x	
q3*	x	x	x
States	q0	q1	q2

Final states – q3

Non-Final States – q1, q2

Every pair between them is distinguishable

States q2 and q0 are distinguishable since:

transition (q2, b) -> q3(final state) and transition (q0, b)-> {}

States q2 and q1 are distinguishable since:

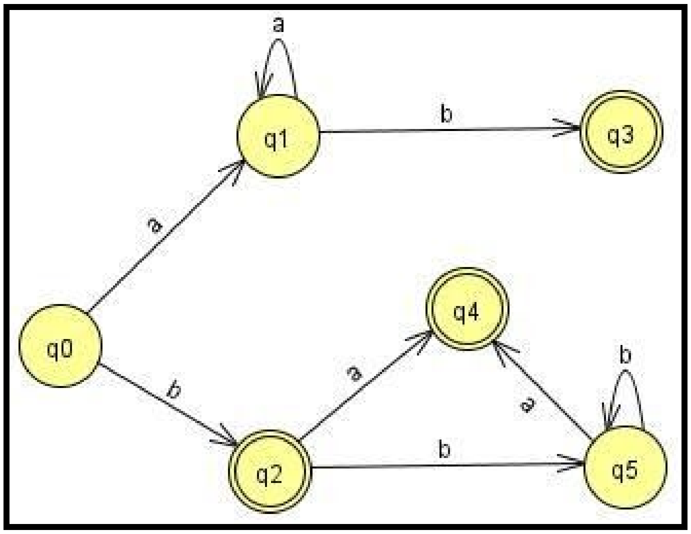
transition (q2, b) -> q3(final state) and transition (q1, b)-> {}

States q1 and q0 are distinguishable since:

transition (q1, a)->q2 and transition (q0, a)->q1 then after the new acquired states we realize that q2 will reach final state while q1 wont.

Hence the above DFA is the minimal form DFA.

b)



q1	X				
q2*	X	X			
q3*	X	X	X		
q4*	X	X	X	X	
q5	X	X	X	X	X
States	q0	q1	q2*	q3*	q4*

Final States – q2, q3, q4

Non-Final States – q0, q1, q5

Every pair between them is distinguishable

States q5 and q0 are distinguishable since:

q5(a) -> q4(final state) and q0(a)->q1(non-final state)

Similarly, for q5 and q1.

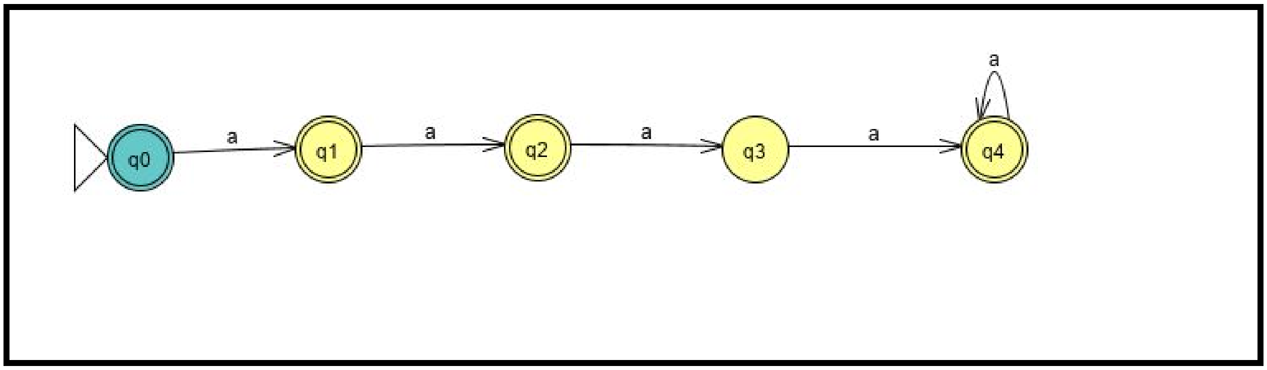
States q4 and a2 are distinguishable since transition (q4, b) -> {} and

transition (q2, b)->q5(non-final state).

Similarly, for all the other pairs.

Hence the above DFA is the minimal form DFA.

c)



q1*	X			
q2*	X	X		
q3	X	X	X	
q4*	X	X	X	X
States	q0*	q1*	q2*	q3

Non-Final States – q3

Final States – q0, q1, q2, q4

Every pair between them is distinguishable

State transition (q4, a) -> q4 and transition (q0, a) -> q1 and again

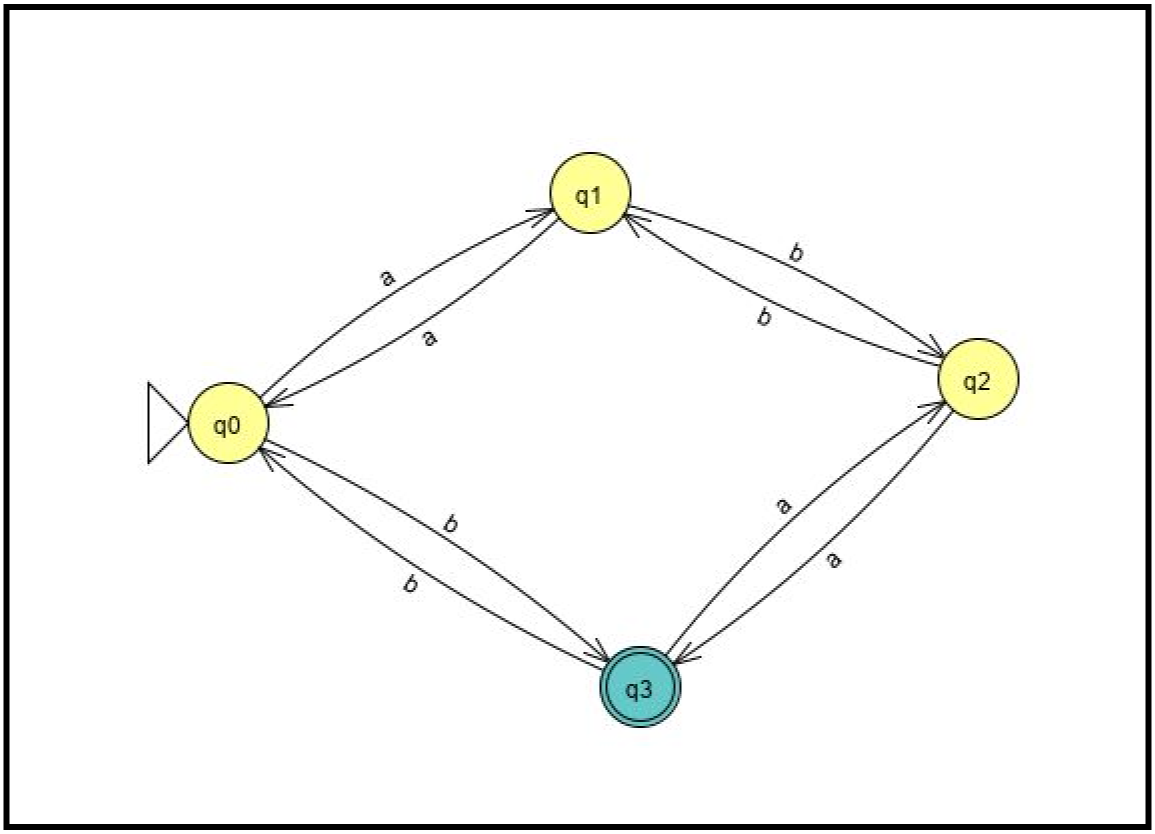
transition (q4, a) -> q4 and transition (q0, a) -> q2 and again

transition (q4, a) -> q4(final state) and transition (q0, a) -> q3(non-final state) which is a non-final state. Hence, they both are distinguishable.

Similarly, if we follow the similar pattern for rest of the state elements transitions, we see that they are distinguishable.

Hence the above DFA is the minimal form DFA.

8. The given DFA is -



The regular expression is -

((aa+ab(bb)*ba)*(b+ab(bb)*a)(a(bb)*a)*(b+a(bb)*ba))*(aa+ab(bb)*ba)*(b+ab(bb)*a)(a(bb)*a)*

9. To show that the language of binary strings of even length having the same number of 0s in the two halves is not regular -

For simplicity sake, let us consider the expression to be w = $0^m1| \ 0^m1$ (there can be a different number of 1's in the two halves as well but here it has been assumed to be the same).
Using the Pumping Lemma concept, we can further split this as -

0^{m-1} 0 1 0^m 1
x y z

On pumping the value of y from the middle -

0^{m-1} 00 1 0^m 1

0^{m+1} 1. 0^m 1

Since, there is no simple repeating pattern or multiple correlated patterns, the language is proved not regular by **Pumping Lemma.**

10. This concept is based on the adversarial game of Pumping Lemma to prove that the language is not regular.

i) The adversary would select $(ab)^{m-1}$ as x,y as ab and z as lambda.

ii) The adversary would select x as $a^{m/2-1}$, y as a and z as b.