Name: Aytaç SEKMEN

Student ID: 2575983

QUESTION 1:

A) (a(b+c)*a+b+aa)(a+b)*

So first box is: (b+c)

Second box: b

Third box: aa

Fourth box: (a+b)*

B)

A=0

B=1

C=0+1

D=2

E=1

F=(0+2)

QUESTION 2:

A) I can actually call that algorithm “state elimination algorithm”(maybe that is not the correct name, I’m not sure). Thanks to that algorithm we can convert NFA’s and DFA’s to corresponding regular expressions, to find their corresponding languages.

B) I think we should somewhat consider the output transitions. At each state while we are following the input transitions we should somewhat build the output string.

Also, since there is no final state I think we can construct our regular expressions as how we want them to be. For example we can achieve a regex which has the last character as “C” or a regular expression which represents the output strings which ends with “B”. Those will give us different regex. These modifications determine the final states, I think.

C) Let's begin from the initial state. There is a transition ((q0,a/A)q1) which actually cause a loop and give us the regular expression(regex) A* in terms of output alphabet. Then we can go to state of q1 with transition ((q0, b/B), q1). Now we can do 2 thing: go q2 or go back to q0 and come back again. Because of that before going to q2 I should put the regex (AA*B)* corresponding to the loop between q1 and q0. Since we are at q1 again. Now we can go to q2 with the transition ((q1, b/B), q2). Now there is loop in q2 so I should add that loop((q2, b/B), q2) , and my regex become: A*B(AA*B)*BB*. After that when we go to q0 with the transition ((q2, a/C),q0), now we get the output strings that end with "C". But since there should be at least one repeat of it i should put it and then concatanate it with its kleen star. So the resultant regex is:

(A*B(AA*B)*BB*C) (A*B(AA*B)*BB*C)*

QUESTION 3:

Hocam I should actually say that this question and third question confused me in many way since we didn't cover tranducers in lesson and this is the first time I see such questions. So I should say that I did this homework accordingly to my understanding.

I decided to look at what output strings can mealy machine produce and what strings can N2 and N3 accept. Because in question's statement it says that "Draw the DFA representing the whole system.". Firstly I thought of transforming P to DFA and transforming 2 NFA's to DFA and the whole system combined with the non deterministic gate to DFA. But I realized that this method will cost so much effort, and it is so much complicated. So I decided to find a *"cunnigly selected strategy"* as u mentioned in the discussion forum.

- First of all to decide the relation between them, I analyzed the NFA's, then I realized that upper part of NFA2(q3, q4) doesnt give anything and CC part of NFA3 is not applicable to the mealy machine. Because we can not get any C followed by another C in an output string from Mealy Machine. So I removed that parts.

- Also I realize that I accept the empty string thanks to N3, and I checked that I can find empty string from Mealy machine since there is kleene star in both part of Union operation. These two images are my analysis for N2 and N3:

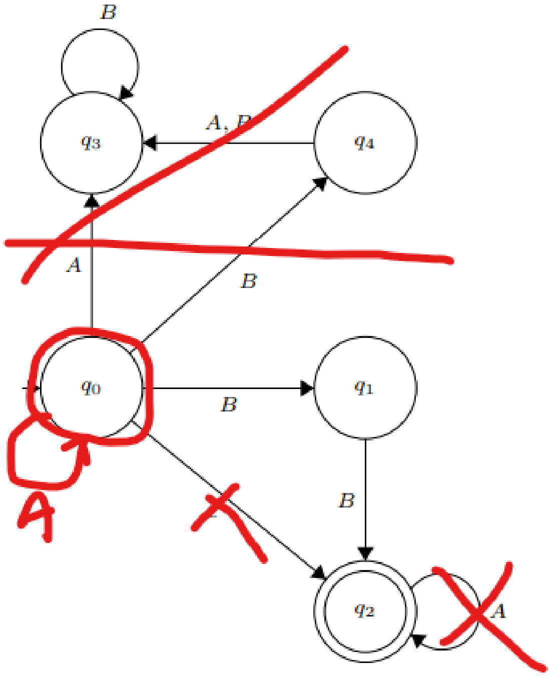


Figure 5. N2

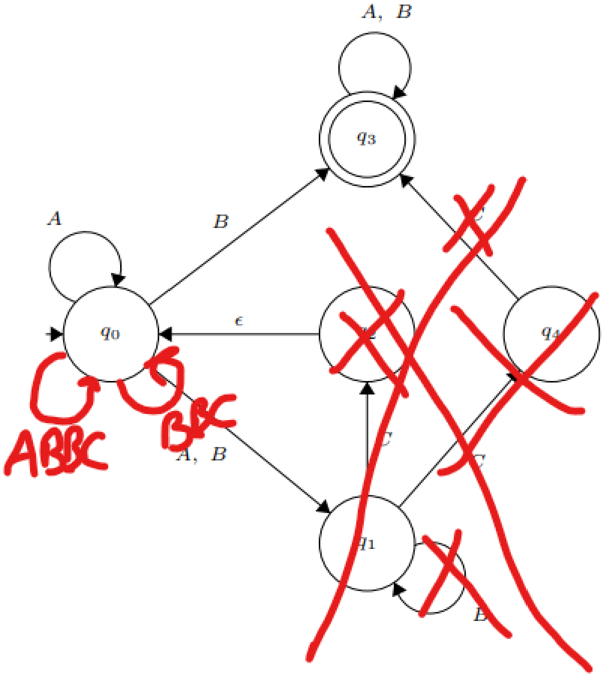
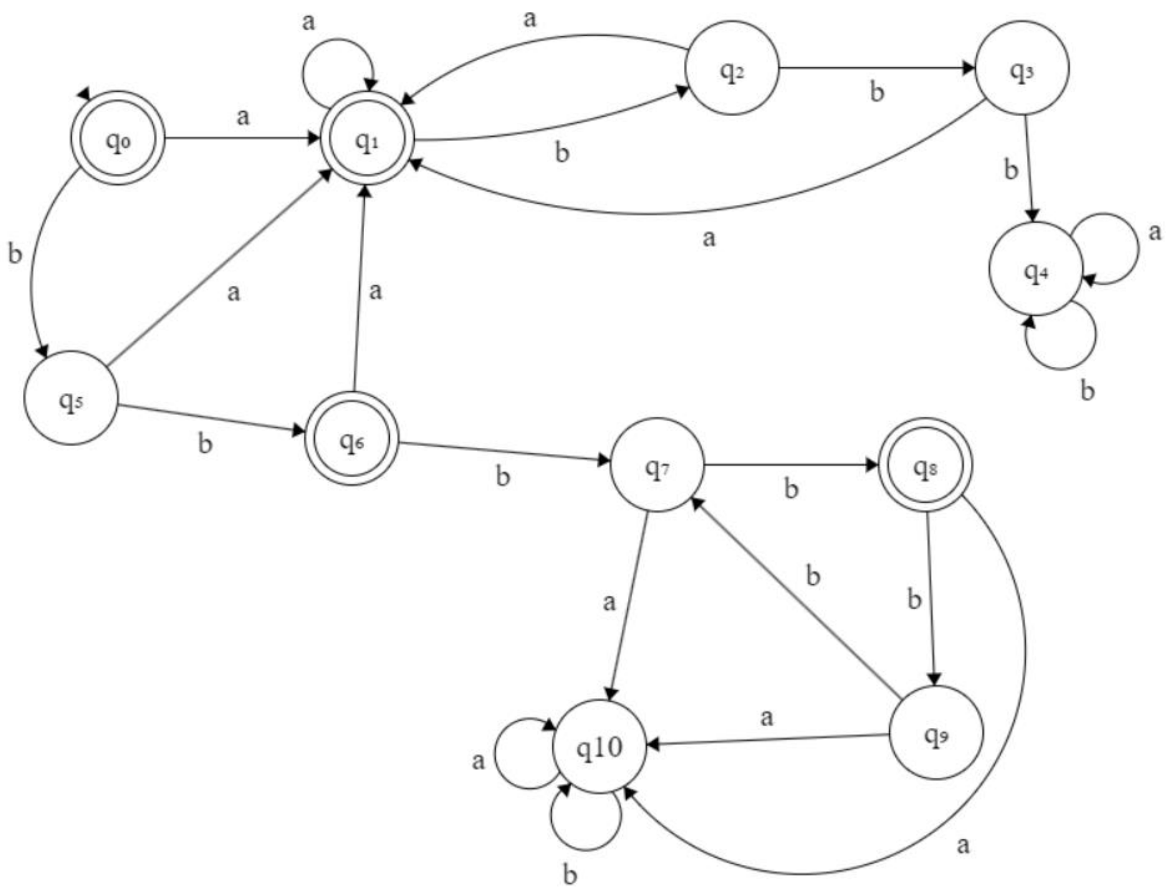


Figure 6. N_3

- Also i realized that if I read input of a or ba or bba, I can not read bb following these inputs because of the structure of Mealy Machine. There is 2 group: 1 group is a*, (ba)*, (bba)* and second group is (bb)* So in my DFA, I should somewhat prevent that happening by using trap state. For example in my drawings, q10 and q4 are trap states.

Final states are: q0,q1,q6,q8

Initial state: q0



Note: Since I wrote my homework in Word my image is not searchable.