

CENG 280

Formal Languages and Abstract Machines

Spring 2022-2023

Homework 2

Name Surname: Doruk Berke Yurtsizoglu
Student ID: 2522225

Answer for Q1

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

b.

A=: 0

B=: 1

C=: 0,1

D=: 2

E=: 1

F=: 0,2

Answer for Q2

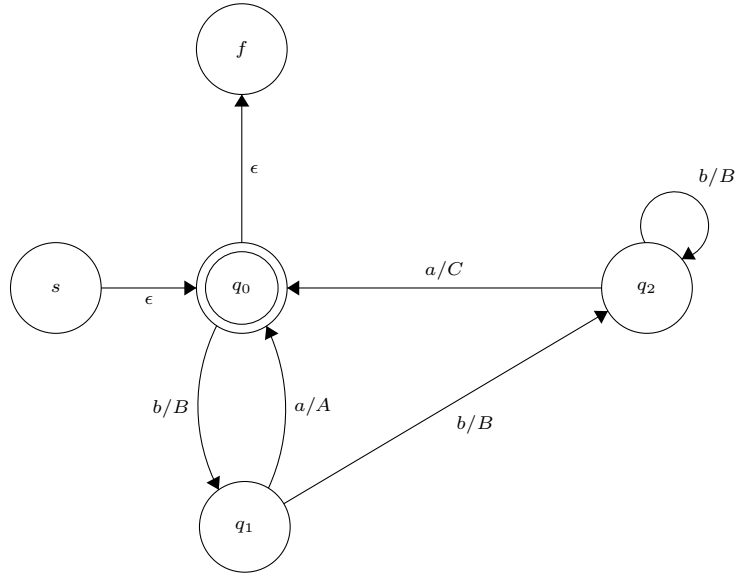
a. We can use state elimination to compute the output language of a Mealy Machine.

b. In order to use the state elimination technic, we must change the process of it a little bit.

- We must trace both outputs and of inputs, if we want to compute the output language of the machine.
- If we look closely to the machine, we can see that the machine doesn't have any final states. However, to apply state elimination to a machine, there must be final states. We can solve this problem by declaring all the states in the machine as final states since the output can end in any state of the machine.

After applying these modifications, we can easily apply state elimination to Mealy Machines.

c. In order to find the regular expression for strings that end with 'C', we will declare q_0 as the final state because that state is the only state which gives a 'C' output. After doing this we will simply apply state elimination.



$$R(q_1, q_0, q_2) = bb^*a/BB^*C$$

$$R(q_0, q_0, q_1) = b/B(a/A \cup bb^*a/BB^*C)$$

$$R(s, f, q_0) = (a/A \cup b/B(a/A \cup bb^*a/BB^*C))^*$$



As can be seen from the result, the regular expression of the machine doesn't have to end with 'C'. However, we solve this problem by applying a simple trick. The thing is, if a word ends with 'C', then there must be BB^*C part. The different ways that we can use this part in a word is $A^*(BA^*)(BB^*C)^*$. What we did is we concatenated the regular expression, and added kleene star to the elements that are connected with a union in order to find all the strings that ends with 'C'.

Answer for Q3

In order to find the DFA representation of the whole system, there are some important points we must consider:

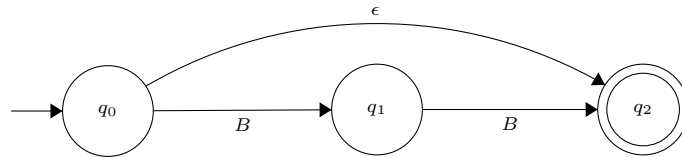
- Although the NFAs take the outputs (A,B,C) of the Mealy Machine; after we apply the DFA conversion to the whole system, the automata will still take the inputs in format (a,b). So, we must construct the DFA with the alphabet (a,b) not (A,B,C).
- We can modify the NFAs N_2 and N_3 in order to make the DFA conversion more applicable. We will analyze the regular expression of the Mealy Machine part by part to achieve our goal. An example for this modification is, in the NFA N_3 there is a part which accepts strings that has "CC":



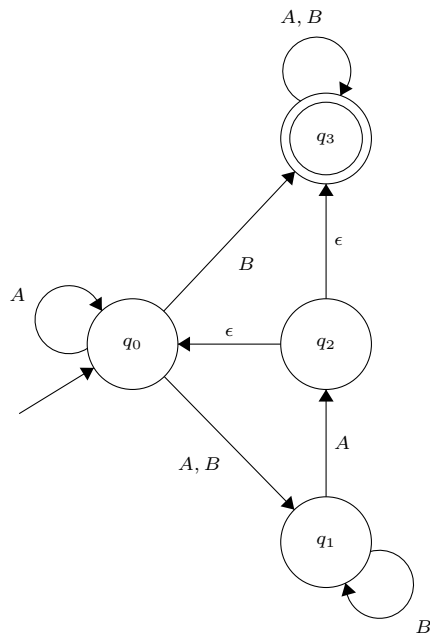
However, we can't give a string like that to N_3 because Mealy Machines doesn't give outputs in that form. So we simply delete that part and so on..

After all of the modifications N_2 and N_3 will look like:

– N_2

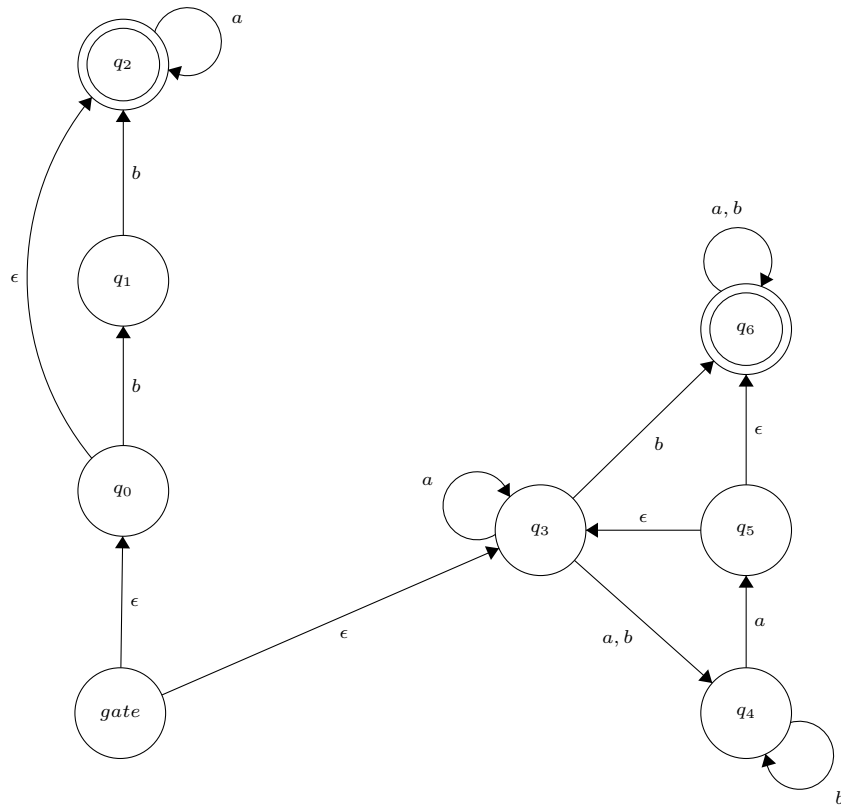


– N_3

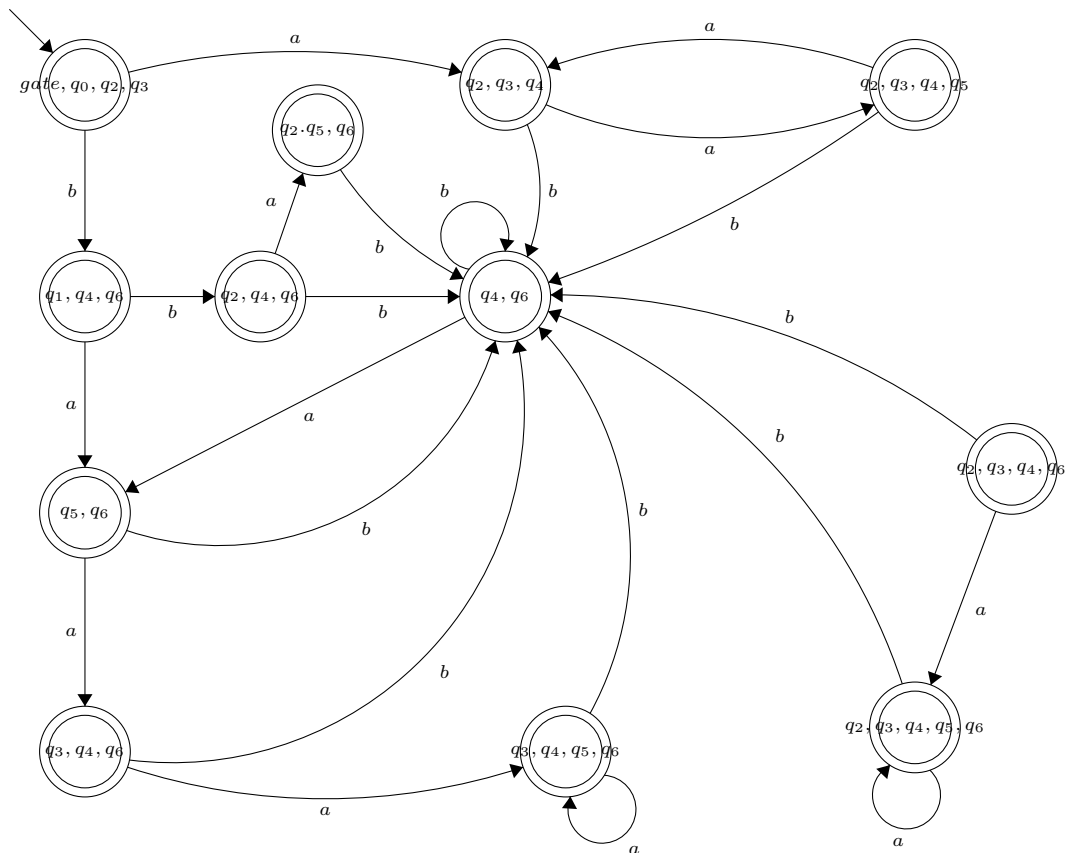


–So that we have the modified versions of the NFAs, now we will change the transition types from outputs of the Mealy Machine to inputs that Mealy Machine takes (a/A,b/B,a/C), and combine both N_2 and N_3 with the non-deterministic gate.

Now, our system will look like this:



– Now, we will apply the NFA to DFA algorithm to the system above. The final DFA will look like:



–By applying these steps, we converted the system to a DFA without worrying about Mealy Machine.