Middle East Technical University Department of Computer Engineering

CENG 280

Formal Languages and Abstract Machines

Spring 2022-2023

Homework 2

Name Surname: Yusuf Sami Lök
Student ID: 2521748

Answer for Q1

a. The expression is below.

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

1. Box: $b+c$
2. Box: aa
3. Box: b
4. Box: $(a+b)^*$

b.
A=: 0
B=: 1
C=: (0+1)
D=: 2
E=: 1
F=: (0+2)

Answer for Q2

a. The algorithm is State Elimination Algorithm. We can modify this algorithm to create a Mealy Machine.

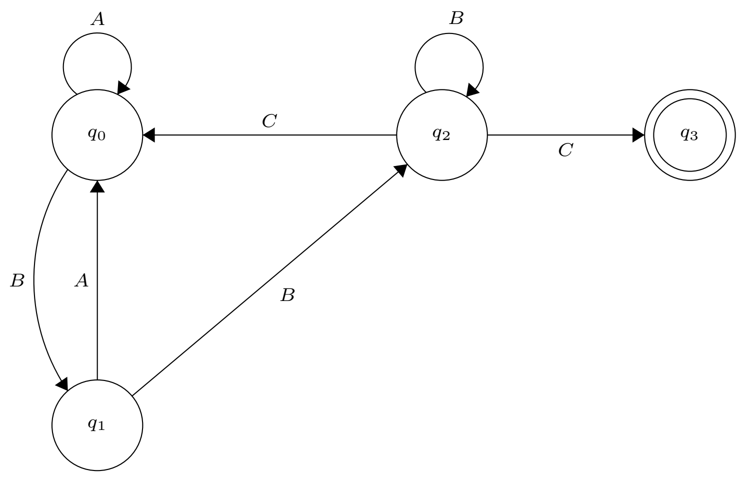
b. First of all, we need to add one additional state for each state, and connect them with empty string(from former state to new state), and make these additional states final states. q_0 is the starting state. We also have to convert to input symbols to output symbol with the function λ tabulated below:

q_{from}	input	q_{to}	$\lambda(q_{first}$,input,$q_{second})$
q_0	a	q_0	A
q_0	b	q_1	B
q_1	a	q_0	A
q_1	b	q_2	B
q_2	b	q_2	B
q_2	a	q_0	C

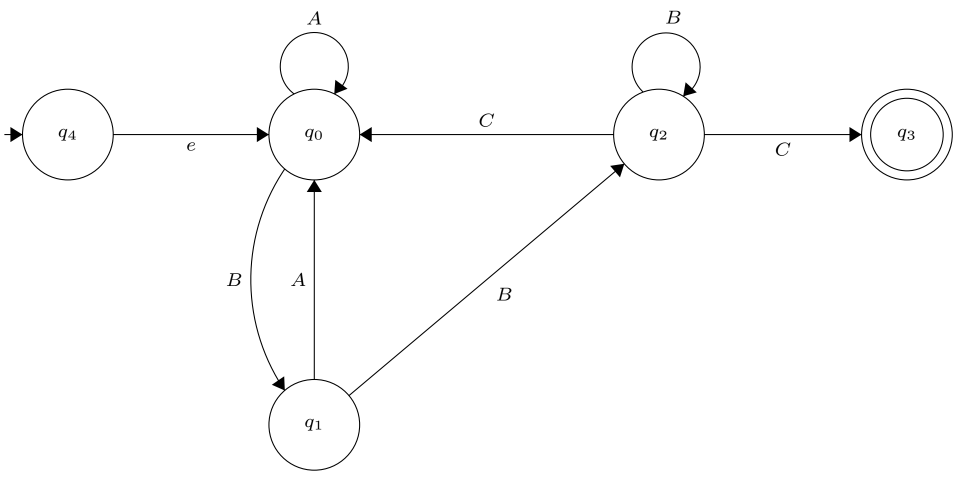
Finally, the rest of the algorithm should be the same.

c. Since we have to find the set of output strings of M1 which are ending with only the letter "C", we can use the previous method but with a different version. Instead of adding one additional final state to all existing states, we should add one final state to q_2 with a transition "a/C". We are doing this because we only want to find the strings with ending "C" letter, and the only state that can create the letter C is state q_2.

1. Adding final state with a transition C from q_2 to new final state q_3.

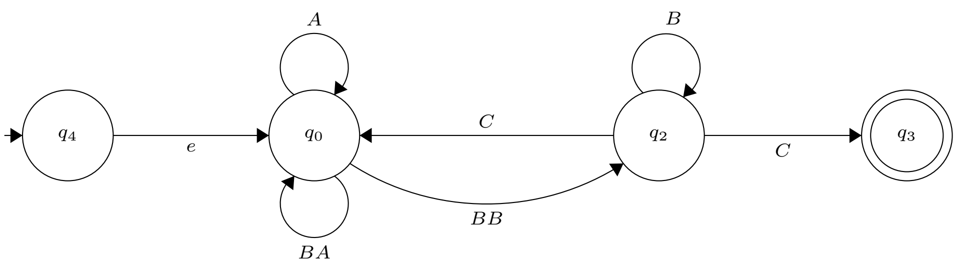


2. Because q_0 has transitions in, we add a new starting state with a transition with empty string from q_4(new starting state) to q_0.

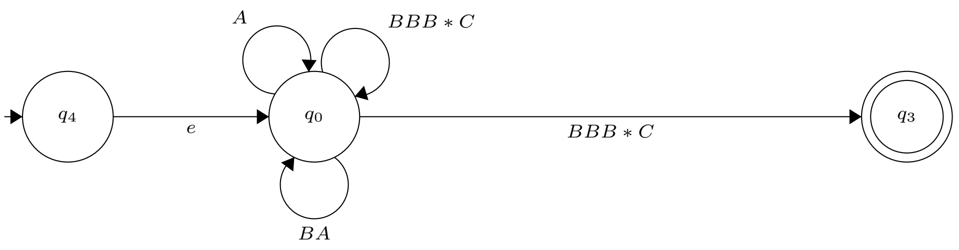


3. Now, we can eliminate one by one the states which are between the starting and final states.

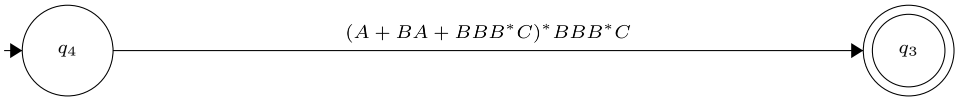
removing q_1



removing q_2



removing q_0



Therefore, the set of output strings of M1 which are ending with "C" with regular expression is $(A+BA+BBB^*C)^*BBB^*C$.

Answer for Q3

First of all, we need to find the intersection of the output P, and accepting strings from NFA's N_2 and N_3. To simplify this, we can process by removing situations that cannot be reached from the output of the P machine using NFAs.

Since the inputs only contain 'a' and 'b', the letters in the system should be 'a' and 'b' as well. In the N2 and N3 machines, we can use 'a' for 'A', 'b' for 'B', and 'bba' for 'BBC'.
N2 machine: We can ignore the states q_3 and q_4 for N2 because there is no way to reach final states from those states. For the states q_1 and q_2, we can accept the following expressions using the regular expression in the output from the P:
–(bb) : $q_0 - q_1 - q_2$ with b
–$a*$: $q_0 - q_2$ with empty string, and self loop with a.(it includes the empty string so it is not necessary to include.

Hence, it is

$$(a^* + bb).$$

N3 machine: We can again ignore the state q_4 because there is no way to create "CC" string from the machine P in order to reach final state. For states q_2 and q_1, we need to consider which outputs coming from P we will accept by thinking as if we have a self-loop on q_0. The acceptable outputs that can be inferred from this section are as follows:

–$(bba)*$ from $q_0 - q_1$ with b, then q_1 self loop with one b, then q_2 with a, finally q_0 with empty string, there is one way more but finding one way is enough, and they are equal. Since we reached the starting state q_0, we can traverse this path zero or more times, so it is $(bba)^*$.

–a^* : self loop with a.

–$ba(a+ba)^*$ $q_0 - q_3$ with a and then one a(because there is no b in the output list but we have ba), and then $(a+ba)^*$ from self loops a and b. Therefore, we have $ba(a+ba)^*$.

If we combine these, we have:

$$((bba)+a)^*ba(a+ba)^*.$$

The other way is: $bb(bb)^*$ $q_0 - q_3$ with b and then with self loop $b(bb)*$. **Hence**, N1 machine, and P machine accepts this regular expression:

$$((bba)+a)^*ba(a+ba)^* + bb(bb)^*$$

The combined form of these three machines as a regular expression is as follows:

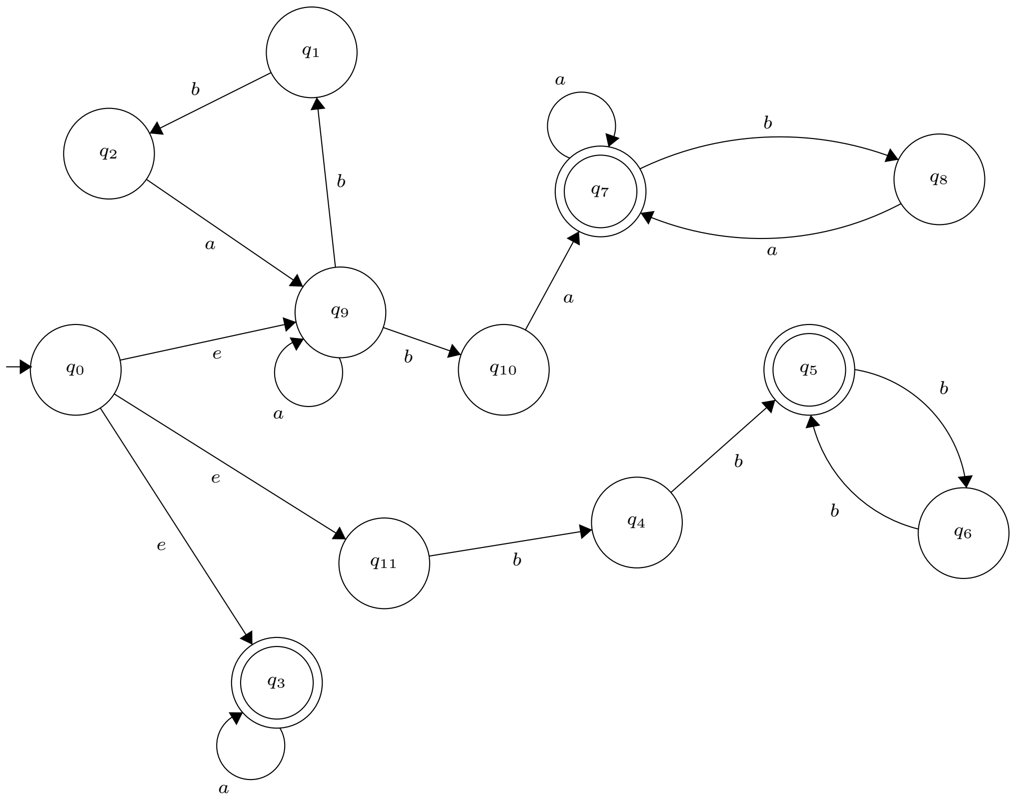
$$(bba+a)^*ba(a+ba)^* + bb(bb)^* + (a^* + bb)$$

We can rewrite this as:

$$(bba+a)^*ba(a+ba)^* + bb(bb)^* + (a^*)$$

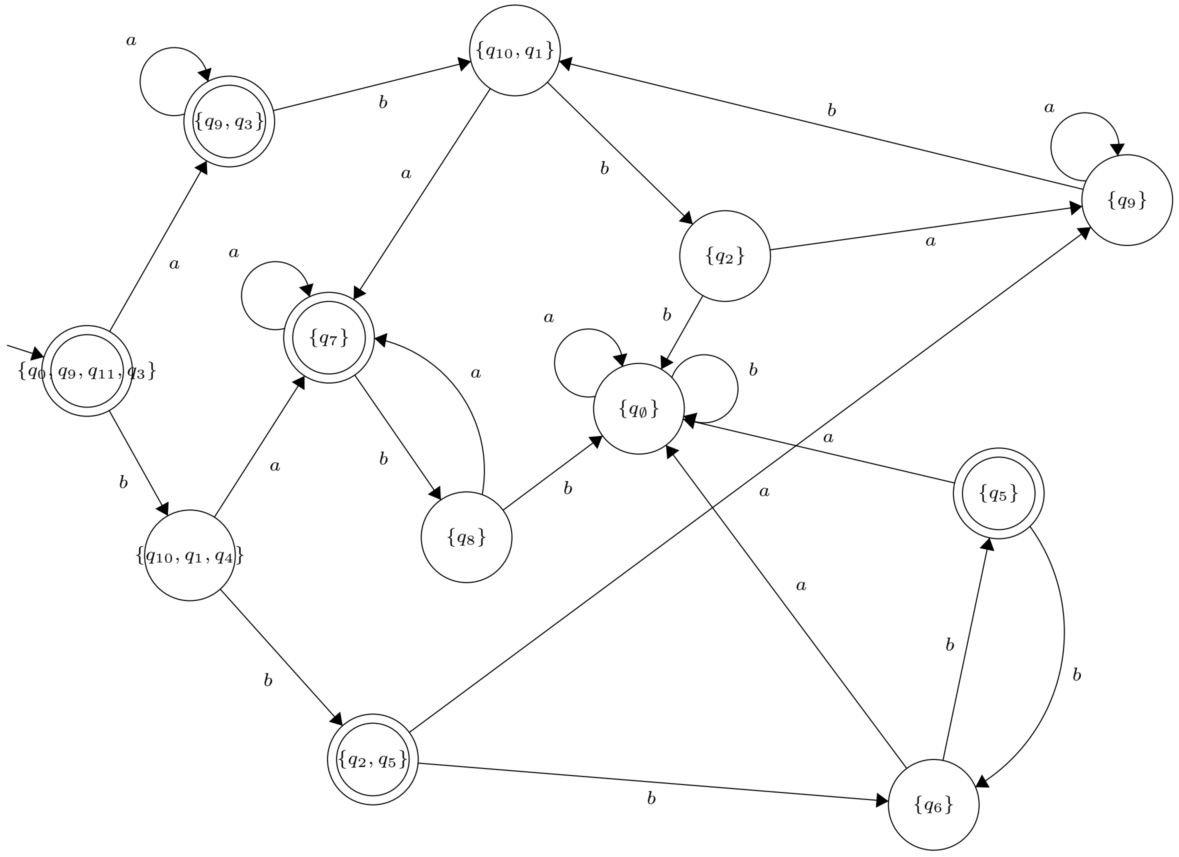
because there is already a bb from $bb(bb)^*$

Let's convert this regular expression we created from these three machines into an NFA.



Finally, we have to just convert this NFA to DFA with the NFA to DFA algorithm. Here is the DFA which is equal to NFA above, and it represents the whole system: $\delta(\{q_0, q_9, q_{11}, q_3\}, a) = \{q_9, q_3\}$

$\delta(\{q_0, q_9, q_{11}, q_3\}, b) = \{q_{10}, q_1, q_4\}$
$\delta(\{q_9, q_3\}, a) = \{q_9, q_3\}$
$\delta(\{q_9, q_3\}, b) = \{q_{10}, q_1\}$
$\delta(\{q_{10}, q_1, q_4\}, a) = \{q_7\}$
$\delta(\{q_{10}, q_1, q_4\}, a) = \{q_2, q_5\}$
$\delta(\{q_{10}, q_1\}, a) = \{q_7\}$
$\delta(\{q_{10}, q_1\}, b) = \{q_2\}$
$\delta(\{q_7\}, a) = \{q_7\}$
$\delta(\{q_7\}, b) = \{q_8\}$
$\delta(\{q_2, q_5\}, a) = \{q_9\}$
$\delta(\{q_2, q_5\}, b) = \{q_6\}$
$\delta(\{q_2\}, a) = \{q_9\}$
$\delta(\{q_2\}, b) = \{\}$
$\delta(\{q_8\}, a) = \{q_7\}$
$\delta(\{q_8\}, b) = \{\}$
$\delta(\{q_9\}, a) = \{q_9\}$
$\delta(\{q_9\}, b) = \{q_1, q_{10}\}$
$\delta(\{q_6\}, a) = \{\}$
$\delta(\{q_6\}, b) = \{q_5\}$
$\delta(\{q_5\}, a) = \{\}$
$\delta(\{q_5\}, b) = \{q_6\}$



It is the DFA which represents the whole system (P,N1,N2).