Middle East Technical University | Department of Computer Engineering

CENG 280

Formal Languages and Abstract Machines

Spring 2022-2023

Homework 2

Solutions

Answer for Q1 (40pts)

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

b.

- $A \rightarrow 0$
- $B \rightarrow 1$
- $C \rightarrow 0+1$
- $D \rightarrow 2$
- $E \rightarrow 1$
- $F \rightarrow 0+2$

Answer for Q2 (35pts)

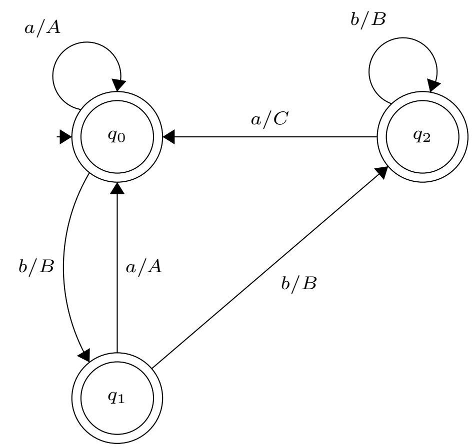
a. State elimination algorithm (the algorithm that converts a given NFA to the corresponding regular expression)

b.

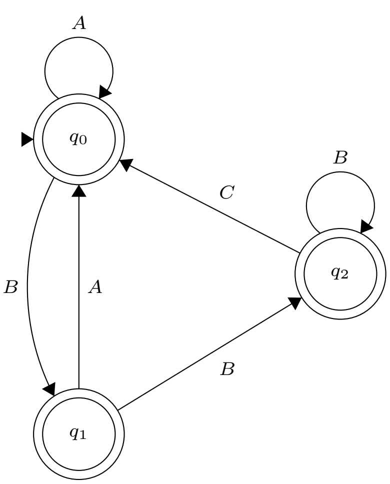
1. Added Step: At the very beginning, mark all the states as final states.
   This is required since a Mealy Machines generates output for any given string (over its alphabet), without requiring the input reaching a specific state.
2. Modification: While eliminating states, construct partial regular expression according to output symbols, instead of input symbols. (i.e. remove input symbols and continue according to output symbols).

Remark: Since the output language is defined as the set of strings that can be outputted by a Mealy Machine, it is input independent.

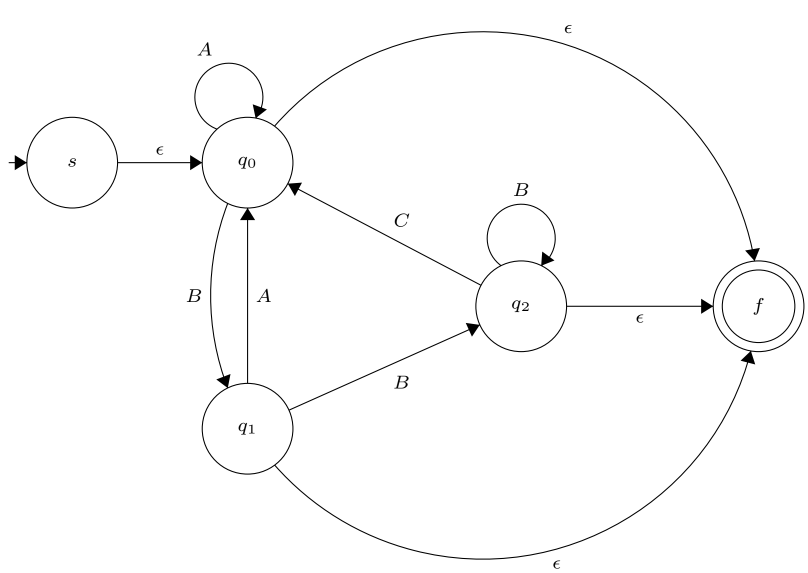
c. Step 1: Mark all states as final states:



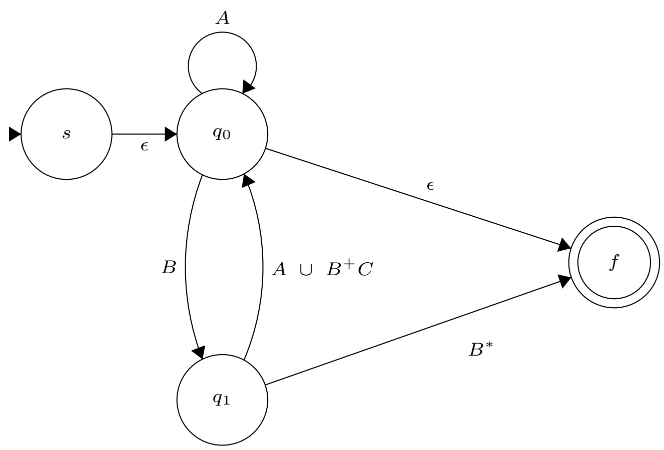
Step 2 : Remove input symbols



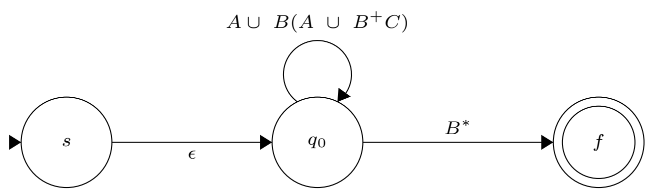
Step 3: Introduce dummy initial and dummy final states:



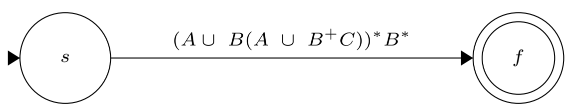
Step 4: Remove q_2:



Step 5: Remove q_1:



Step 6: Remove q_0:



The output language of M_1 is $L_{out}(M_1) = (A \cup B(A \cup B^+C))^*B^*$.

Since the question asks the subset of that language which is consists of strings that are ending with C, the correct answer is:

$$(A \cup B(A \cup B^+C))^*B(B^+C)$$

Remark: A simpler solution is possible by marking only q_0 as the final state. This is the slight modification on the algorithm proposed as the solution of part-b, which is mentioned at the hint. This is doable for this question since it looks for the strings that are ending with C, which can only be generated when the last transition outputs C; and in M_1, the only transition outputting C ends up at q_0.

Answer for Q3 (25pts)

Let us call the given extended regular expression e_P.
Let $e_P = (e_0 \cup e_1 \cup e_2)^* \cup e_3$ where
$e_0 = (a/A)$
$e_1 = (b/B)(a/A)$
$e_2 = (b/B)(b/B)(a/C)$
$e_3 = [(b/B)(b/B)]$

The input language e_0 generates is: $L_{in}(e_0) = \{a\}$
The output language e_0 generates is: $L_{out}(e_0) = \{A\}$

The input language e_1 generates is: $L_{in}(e_1) = \{ba\}$
The output language e_1 generates is: $L_{out}(e_1) = \{BA\}$

The input language e_2 generates is: $L_{in}(e_2) = \{bba\}$
The output language e_2 generates is: $L_{out}(e_2) = \{BBC\}$

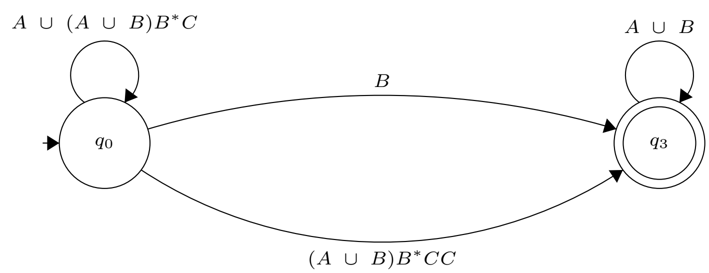
The input language e_3 generates is: $L_{in}(e_3) = L((bb)^*)$
The output language e_3 generates is: $L_{out}(e_3) = L((BB)^*)$

Then:
The input language e_P generates is: $L_{in}(e_P) = \{L_{in}(e_0) \cup L_{in}(e_1) \cup L_{in}(e_2)\}^* \cup L_{in}(e_3)$
The output language e_P generates is: $L_{out}(e_P) = \{L_{out}(e_0) \cup L_{out}(e_1) \cup L_{out}(e_2)\}^* \cup L_{out}(e_3)$

For the later use, we can also represent e_P as $e_{P1} \cup e_{p_2}$ where $e_{P1} = (e_0 \cup e_1 \cup e_2)^*$ and $e_{P2} = e_3$
In that case, $L_{in}(e_P) = L_{in}(e_{P1}) \cup L_{in}(e_{P1})$ and $L_{out}(e_P) = L_{out}(e_{P1}) \cup L_{out}(e_{P1})$

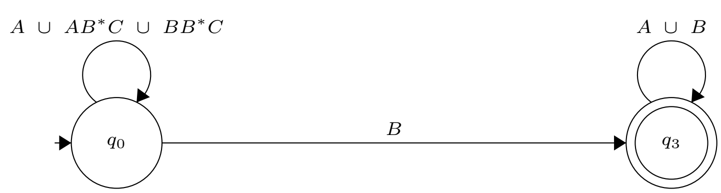
We know that the strings that are inputted to N_2 and N_3 are elements of $L_{out}(e_P)$. We should check which ones among those strings are accepted by at least one of N_2 and N_3.

N_3 can be simplified as:



Note that $(A \cup B)B^*CC)$ (lower arrow going from q_0 to q_3) is non-functional (i.e. it will never be used) in the given system since it includes CC substring which is not a substring of any string in $L_{out}(e_P)$.

Then, the functional part of N_3, say N_3', can be drawn as:



It is easily seen that N_3' accepts all strings of $L(e_{P2})$. Therefore;
Result 1: $L_{in}(e_3)$, which is equal to $L_{in}(e_{P2})$, is accepted by the overall system.

When it comes to $L(e_{P1})$, let us analyse the behaviour of components of e_{P1} on N_3':

When N_3' is at the state q_0, if
- A is fed, machine stays at q_0.

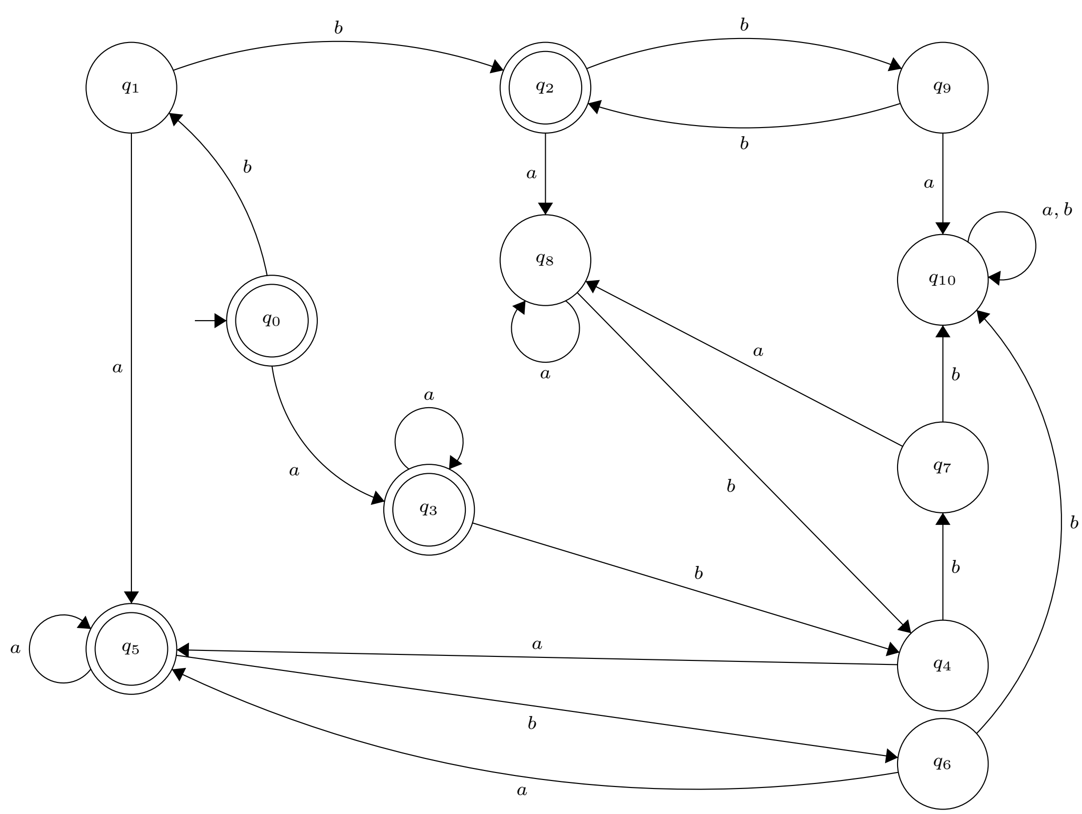
- BA is fed, machine ends up at q_3.
- BBC is fed, machine stays at q_0.

When $N_3^{'}$ is at the state q_3, if
- A is fed, machine stays at q_3.
- BA is fed, machine stays at q_3.
- BBC is fed, machine cannot proceed.

Since BBC cannot proceed to and from q_3, strings ending with BBC are not accepted by $N_3^{'}$, however strings including any number of BBC substring (but not at the end) are accepted. When it comes to A and BA, we see that the machine stays at q_0 when fed with A. Further, the only component of e_{P1} that make $N_3^{'}$ to reach its final state q_3 is BA. Lastly, once the machine is at q_3, it continues to accept any number of A and BA. To sum up, the sublanguage of $L(e_{P1})$ that is accepted by $N_3^{'}$ (and therefore by N_3) is $(A \cup BBC)^*(BA)(A \cup BA)*$ or equivalently, $(L_{out}(e_0) \cup L_{out}(e_2))^* L_{out}(e_1)(L_{out}(e_0) \cup L_{out}(e_1))^*$. This concludes that the subset of $L_{in}(e_{P1})$ accepted by N_3 is $(L_{in}(e_0) \cup L_{in}(e_2))^* L_{in}(e_1)(L_{in}(e_0) \cup L_{in}(e_1))^*$, or equivalently:
Result 2: The subset of $L_{in}(e_{P1})$ that is accepted by N_3 is $(a \cup bba)^* ba(a \cup ba)^*$.

On the other hand, $L(N_2) = BBA^* \cup A^*$; and its intersection with $L_{out}(e_{P1})$ is A^*, which means the subset of $L_{in}(e_{P1})$ accepted by N_2 is $(L_{in}(e_0))^*$. (L_{P2} is not checked since Result 1 already concludes it is accepted by the system.)
Result 3.: The subset of $L_{in}(e_{P1})$ that is accepted by N_2 is a^*.

To sum up, from Result 1, Result 2, Result 3 and since the $L(e_P) = L(e_{P1}) \cup L(e_{P2})$, the language accepted by the overall system is $L_s = a^* \cup (bb)^* \cup (a \cup bba)^* ba(a \cup ba)^*$. Hence, the following DFA D_1 with $L(D_1) = L_s$ represents the whole system:



DFA $\mathbf{D_1}$