

# CENG 280

## Formal Languages and Abstract Machines

Spring 2022-2023

### Homework 2

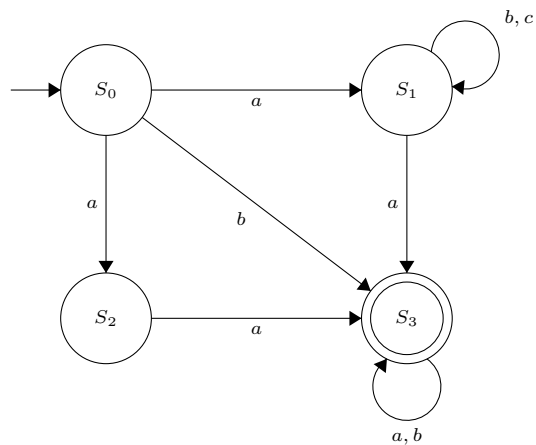
---

## Regulations

1. The homework is due by **23:59 on April 9th, 2023**. Late submission is not allowed.
2. Submissions will be via ODTUClass, do not send your homework via e-mail.
3. You can use any typesetting tool (LaTeX, Word, etc.) while writing the homework. However, you must upload the homework as a **searchable pdf file**. Other formats (e.g handwriting etc.) will **not** be considered for grading.
4. Name the PDF file you submit as **HW2\_yourStudentID.pdf** (e.g. HW2\_1234567). Submissions violating the naming convention will be penalized.
5. Write your name and student ID number to the top of your solution sheets. A grade reduction will be applied to the solution sheets without a name and ID on them.
6. Send an e-mail to **both garipler@metu.edu.tr** and **mduymus@ceng.metu.edu.tr** if you need to get in contact.
7. Please give as neat and as brief answers as possible.
8. **This is an individual homework, which means you have to answer the questions on your own. Any contrary case including but not limited to getting help from automated tools, sharing your answers with each other, extensive collaboration etc. will be considered as cheating and university regulations about cheating will be applied.**

## Question 1 (40pts)

a.



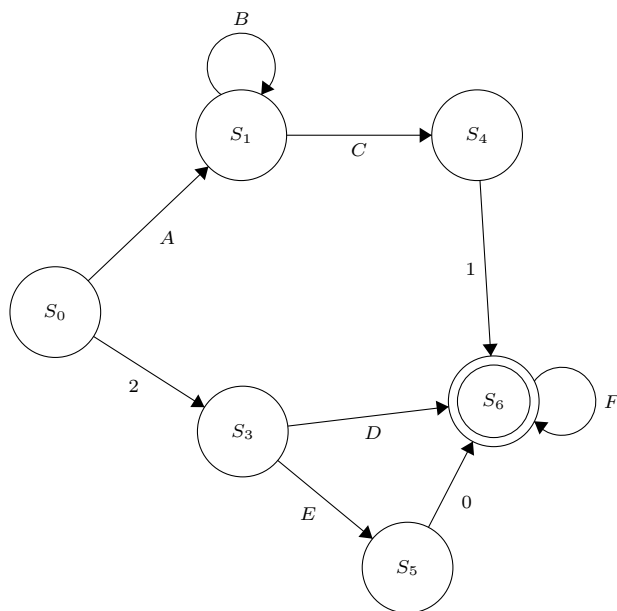
**Figure 1.** NFA  $N_1$

For the NFA given above ( $N_1$ ) fill the boxes in the following regular expression using  $\Sigma = \{a, b, c\}$  and parentheses or symbols such as  $+$  or Kleene Star;

$$(a(\square)^*a + \square + \square)\square$$

Each box is worth 4 points.

b.



**Figure 2.** NFA  $N'_1$

The NFA  $N'_1$  represents the following regular expression;

$$((01^*(0+1)1) + 2(2+10))(0+2)^*$$

What are the values of  $A$  to  $F$ ? Each one is worth 4 points.

## Question 2 (35pts)

**Definition.** Finite state transducers are DFA-like machines which translate the input strings into output strings. A Mealy Machine is a deterministic finite state transducer defined as such:

A Mealy Machine  $M = (Q, s, \Sigma, O, \delta, \lambda)$  is a 6-tuple where

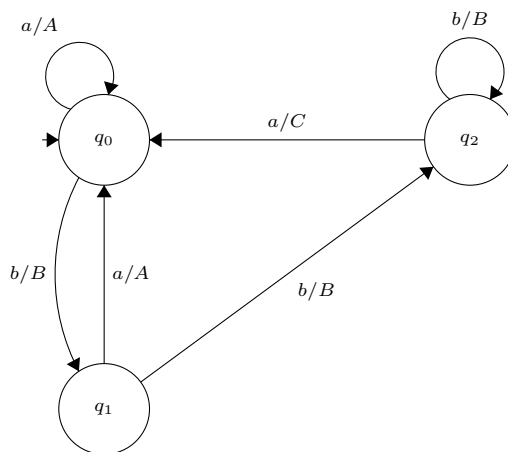
- $Q$  is a finite set of states
- $s$  is the starting state
- $\Sigma$  is the input alphabet
- $O$  is the output alphabet
- $\delta : Q \times \Sigma \rightarrow Q$  is the transition function which maps each (state, input symbol) pair to a state.
- $\lambda : Q \times \Sigma \rightarrow O$  is the output function which maps each (state, input symbol) pair to an output symbol.

In contrast to DFAs, Mealy Machines do not have final states; thus, they do not retain the concept of accepting or rejecting a string. What they do is reading input from the input tape and writing the output to the output tape. On the other hand they have output alphabets and output functions. The output function at first sight resembles the transition function. It takes the same input as the transition function but returns the output symbol to be emitted at the transition.

To illustrate, a simple Mealy Machine is given below:

$M_1 = (Q, s, \Sigma, O, \delta, \lambda)$  where  $\Sigma = \{a, b\}$ ,  $O = \{A, B, C\}$  and  $Q, s, \delta, \lambda$  are as given at the graphical representation at **Figure3**.

**Remark.**  $x/Y$  denotes the input symbol  $x$  is read and the output symbol  $Y$  is outputted.



**Figure 3.** Mealy Machine  $M_1$

**a.** Notice that the set of output strings that can be produced by a Mealy Machine (i.e. output language of a Mealy Machine) is a regular language. In fact, an algorithm to compute the output language of a given Mealy Machine can easily be devised by slightly modifying one of the algorithms you have already learned (in this course). State which algorithm is it.

**Remark:** Your answer may be in the form "the algorithm which is used for this" or "the algorithm which does that" or the exact name of the algorithm.

**b.** State the modification(s) that must be made on the algorithm. Your answer does not have to be very precise, but must somehow express the required modification(s). Also **very briefly** explain your reasoning. Answers in the form "Such a step must be added in order to ...", "Instead of doing this on that, doing this on that is required, since...". will also be accepted. If multiple modifications/additions are required, present each of them as different bullet points.

**Remark:** The original algorithm may include steps that are redundant in this context. If so, you do not have to write anything about such steps. Just assume they do not exist and focus only on the step(s) that must be modified.

**Hint:** Only a few modifications/additions are sufficient.

**c.** Applying the method you proposed at part-b, find the set of output strings of  $M_1$  which are ending with "C" and express it as a regular expression ( $M_1$  is the Mealy machine given at the Figure3.). Show steps of your solution.

**Hint:** Do not attempt to find the output language and then to truncate it. Directly finding the requested set is much easier. A slight simplification on the method you proposed at part-b may be required and will be accepted (or may not be required, depending on the method you proposed).

## Question 3 (25pts)

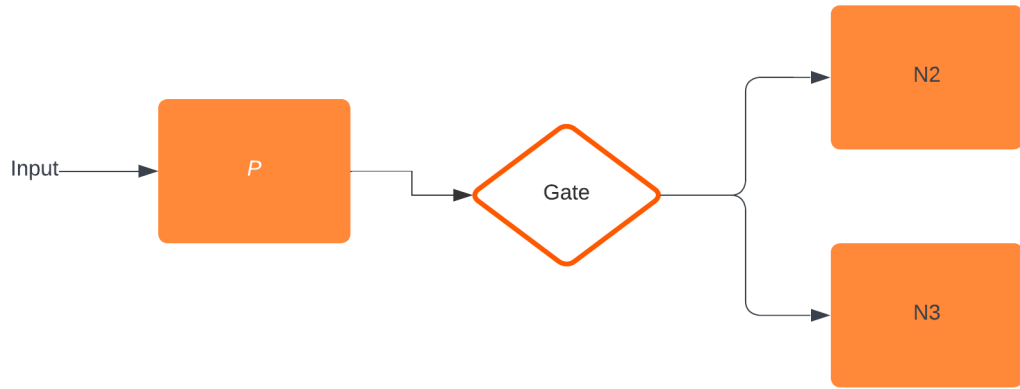
**Definition.** Let us extend the definition of regular expression such that it does not only represent input strings but also represent output strings at the same time. That is possible by constituting regular expressions from input/output symbol pairs instead of only input alphabet symbols. To illustrate,  $(a/A)[(b/C)]^*$  denotes a system producing the set of output strings  $AC^*$  while recognizing the set input strings  $ab^*$ ;  $[(a/A) \cup (b/B)(a/C)]^*$  denotes a system producing the set of output strings  $(A \cup BC)^*$  while recognizing the set of input strings  $(a \cup ba)^*$ . I/O pairs are shown in parentheses so as to determine where the former output ends and where the latter input starts.

Notably, that extended form of regular expressions enable us to represent the relation between input and output; which cannot always be captured by providing two separate regular expressions for the sets of input string and output strings.

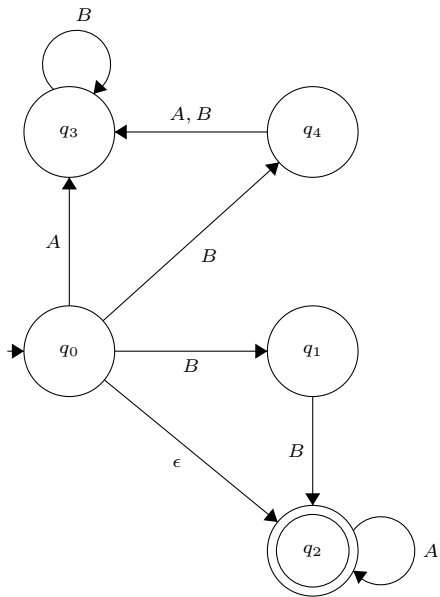
You are given a system consisting of a black-box program  $P$ , a non-deterministic gate and two NFAs  $N_2$  and  $N_3$ . The program  $P$  takes the input from the outside world, processes it to produce an output, and transmits the output to the non-deterministic gate. The gate selects one of the two NFAs non-deterministically and inputs the output of  $P$  to the selected NFA.

The NFAs  $N_2$  and  $N_3$  are as given at Figure5 and Figure6 respectively. Moreover, it is known that, the input/output relation of the program  $P$  can be represented by a Mealy Machine; further, the I/O behaviour of that Mealy Machine can be expressed as

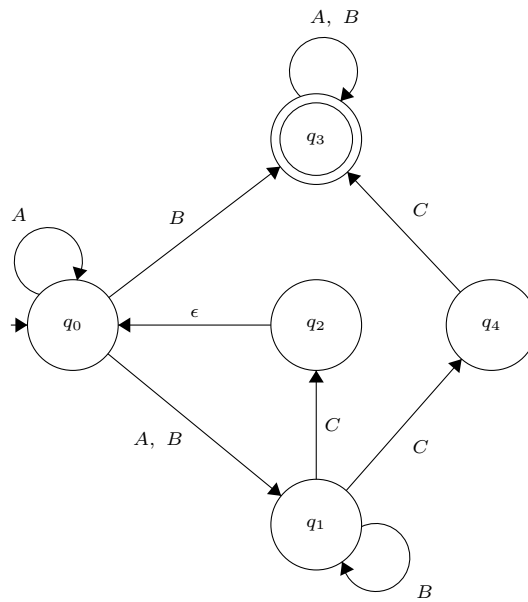
$$[(a/A) \cup (b/B)(a/A) \cup (b/B)(b/B)(a/C)]^* \cup [(b/B)(b/B)]^*$$



**Figure 4.** Described System



**Figure 5.**  $N_2$



**Figure 6.**  $N_3$

Draw the DFA representing the whole system. Show and very briefly explain each step of your solution.

**Hint1:** Treat the non-deterministic gate as if it is a simple NFA (but you do not have to explicitly model it as an NFA).

**Hint2:** The solution might be simpler than you think!