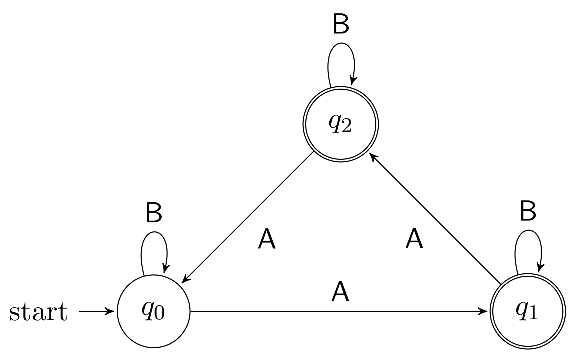
Homework 2 – Due Thursday, September 22, 2022 at 11:59 PM

Reminder Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the course staff if asked. You must also identify your collaborators and write "Collaborators: none" if you worked by yourself. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

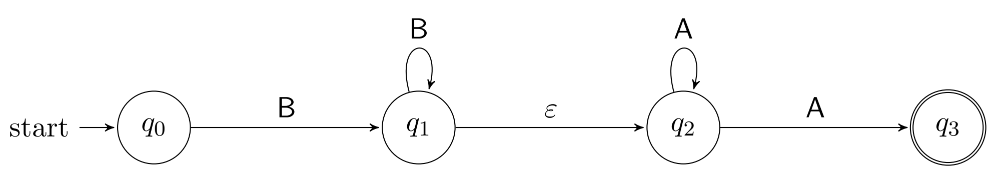
Problems There are 7 required problems and two bonus problems.

1. Consider the following state diagram of a DFA M using alphabet $\Sigma = \{\mathsf{A}, \mathsf{B}\}$.



   (a) What is the start state of M?
   (b) What is the set of accept states of M?
   (c) Give a formal description of the machine M (i.e., as a 5-tuple).
   (d) What is the language recognized by M? (Hint: It has a simple English description using modular arithmetic.)

2. Consider the following state diagram of an NFA N over alphabet $\{\mathsf{A}, \mathsf{B}\}$.



   (a) Give the formal description of N as a 5-tuple.
   (b) Consider running N on input BBA. Give examples (i) of a computation path that leads N to an accept state when run on this input, (ii) a computation path that leads N to a reject state, and (iii) a computation path that leads N to fail before it's read the entire input.
   (c) What is the language recognized by N?
   (d) Use the subset construction to convert N into a DFA recognizing the same language. Give the state diagram of this DFA – only include states that are reachable from the start state.

3. This problem will be autograded using AutomataTutor. Visit `http://automatatutor.com/` and click on "Sign Up." Make an account using your name and `@bu.edu` email address (it is important for recording grades that the information for your account match the information on the course list provided by the university). We'll provide more specific information about how to register for this course on Piazza.

   Give state diagrams of DFAs with as few states as you can recognizing the following languages. You may assume that the alphabet in each case is $\Sigma = \{0, 1\}$.

   (a) $L_1 = \{w \mid w \text{ begins with 1 and ends with 00}\}$.

   (b) $L_2 = \{w \mid w \text{ contains at most three 1's}\}$.

   (c) $L_3 = \{w \mid w \text{ contains the substring 010}\}$.

   Give state diagrams of NFAs with as few states as you can recognizing the following languages. You may assume that the alphabet in each case is $\Sigma = \{0, 1\}$.

   (d) $L_4 = \{xyz \mid \text{ string } x \text{ consists only of 0's, string } y \text{ consists only of 1's, and string } z \text{ consists only of 0's and contains at least one 0}\}$.

   (e) $L_5 = \{w \mid w \text{ contains substrings 100 and 10 which do not overlap}\}$.

4. Draw (and include in the PDF you submit to Gradescope) state diagrams of DFAs with as few states as you can recognizing the following languages. You may assume that the alphabet in each case is $\Sigma = \{0, 1\}$.

   (a) $L_6 = \{w \mid w \text{ is a string of the form } x_1y_1x_2y_2 \ldots x_ny_n \text{ for some integer } n \geq 0 \text{ such that } x_i, y_i \in \{0, 1\} \text{ and } x_i = y_i \text{ for all } i\}$.

   (b) $L_7 = \{w \mid w \text{ represents a binary number that is congruent to 1 modulo 3}\}$. In other words, this number minus 1 is divisible by 3. The number is presented starting from the most significant bit and can have leading 0s.

5. *Think about, but do not hand in:* A DFA or NFA can, in general, have zero, one, or many accept states. Show that every NFA can be converted into another NFA recognizing the same language, but which has exactly one accept state. (This is solved exercise 1.11 in Sipser if you'd like to check your solution.)

   To hand in: Prove that this is not true for DFAs. In particular, show that every DFA recognizing the language of problem 1(d) requires at least two accept states.

6. On Tuesday, we'll show that if M is a DFA that recognizes a language A, then we can obtain a DFA that recognizes $\overline{A}$ by swapping the accept and non-accept states in M.

   (a) Show, by giving an example, that if N is an NFA recognizing a language B, then we do not necessarily obtain an NFA recognizing $\overline{B}$ by swapping the accept and non-accept states in N.

   (b) Is the class of languages recognized by NFAs closed under complement? Explain your answer.

7. On Tuesday, we'll show that the class of regular languages is closed under the star operation. This problem will help you investigate this property.

   (a) Let $A = \{w \in \{0, 1\}^* \mid w \text{ ends with } 1\}$. Give the state diagram of a 2-state NFA N recognizing A.

   (b) Give a simple (English or set-builder) description of the language A^*.

(c) Consider the following **failed** attempt to construct an NFA recognizing A^*: Add an ε transition from every accept state of N to the start state, and make the start state an accept state. Draw the state diagram of this NFA, and call it N'.

(d) What is $L(N')$? Give an example of a string w such that $w \in L(N')$, but $w \notin A^*$.

(e) Give the state diagram of an NFA that *does* recognize A^*.

Bonus Problems

1. Show that for any natural number n, the language $MOD_n = \{w \mid w \text{ represents a binary number that is divisible by } n\}$ is regular. The number is presented starting from the most significant bit and can have leading 0's.

2. A coNFA is like an NFA, except it accepts an input w if and only if *every* possible state it could end up in when reading w is an accept state. (By contrast, an NFA accepts w iff *there exists* an accept state it could end up in when reading w.) Show that the class of languages recognized by coNFAs is exactly the regular languages.

3. In this problem, you will show that in the worst case, the subset construction uses a number of states that is optimal up to a factor of 2.

   (a) For a natural number k, let R_k be the language over alphabet $\{0, 1\}$ consisting of strings w such that the kth symbol from the right of w is a 0. Show that R_k is recognized by an NFA with $k + 1$ states.

   (b) Show that every DFA recognizing R_k requires at least 2^k states.