CS 332: *Theory of Computation* Prof. Mark Bun
Boston University February 4, 2025

Homework 3 – Due Tuesday, February 11 at 11:59 PM

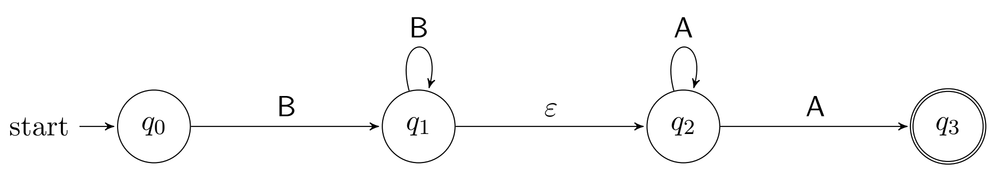
Reminder Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the course staff if asked. You must also identify your collaborators and write "Collaborators: none" if you worked by yourself. Getting solutions from outside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

Problems There are 7 required problems and one bonus problems.

1. *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 $\{w \in \{0,1\}^* \mid w \text{ contains at most one } 1\}$ requires at least two distinct accept states.

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) What is the language recognized by N?

   (c) 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. On Wednesday, 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 a counterexample, 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.

4. On Wednesday, 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^*.

5. (**Closure properties**)

(a) Given languages A, B, define the language $MIX(A, B)$ by

$$MIX(A, B) = \{x_1 y_1 x_2 y_2 \dots x_n y_n \mid n \geq 0, x_i \in A, y_i \in B\}.$$

Note that each x_i, y_i is a *string*. Show that the class of regular languages is closed under MIX. Hint: You don't need to construct an NFA recognizing $MIX(A, B)$ if you can find a way to express it in terms of other operations.

(b) Given a language A over alphabet Σ, define the language $TAIL(A) = \{y \in \Sigma^* \mid xy \in A \text{ for some } x \in \Sigma^*\}$. Show that the regular languages are closed under $TAIL$.

6. (**Regex to description**) Give plain English descriptions of the languages generated by each of the following regular expressions

(a) $(a \cup b)^* \cup c^*$

(b) $1(000)^*1$

(c) $a(ba)^*b$

(d) $\emptyset^*$

(e) $(\emptyset \cup \varepsilon)^*$

7. (**Description to regex**) Please log on to AutomataTutor to submit solutions for this question. Give regular expressions generating the following languages:

(a) $\{w \in \{0,1\}^* \mid w \text{ has exactly two 0's and at least one 1}\}$

(b) $\{w \in \{0,1\}^* \mid w \text{ is not the string } 01\}$

(c) $\{w \in \{0,1\}^* \mid \text{the number of 1's in } w \text{ is divisible by 3}\}$.

8. (**Bonus Problem**) 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.