CS 332: *Theory of Computation* — Prof. Mark Bun
Boston University — January 25, 2024

Homework 1 – Due Thursday, February 1, 2024 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. Collaboration is not allowed on problems marked "INDIVIDUAL."

Problems There are 6 required problems and one bonus problem. Problems 1-5 are to be turned in via Gradescope. Problem 6 will be autograded using AutomataTutor.

1. For each of the following languages, (i) give a plain English description of the language, (ii) describe the decision problem corresponding to that language (i.e., "Given a string ... determine whether ..."), and (iii) give two examples of strings in the language and two examples of strings that are in Σ^* but are outside the language.

   (a) $L_1 = \{0x1 \mid x \in \{0,1\}^*\}$ where the alphabet $\Sigma = \{0,1\}$

   (b) $L_2 = \{w \in \{a,b\}^* \mid |w| \geq 5\} \cap \overline{\{x\mathsf{abba}y \mid x,y \in \{\mathsf{a},\mathsf{b}\}^*\}}$ where the alphabet $\Sigma = \{a,b\}$

2. For each of the following languages, (i) describe the language using set-builder notation and union/intersection/complement/reverse/concatenation operations (the notation used in Problem 1), (ii) describe the decision problem corresponding to that language, and (iii) give two examples of strings in the language and two examples of strings that are in Σ^* but are outside the language.

   (a) $L_3 =$ the set of all strings over alphabet $\{\mathsf{a},\mathsf{b},\mathsf{c}\}$ that are palindromes (read the same forwards and backwards) or that end with c. (For example, bacab and cabbac are palindromes.)

   (b) $L_4 =$ the set of all strings over alphabet $\{1,2,\ldots,9\}$ whose length is divisible by s, where s is the first symbol of the string.

3. Which of the following statements are true or false, for all alphabets Σ? For each, provide either a proof or a counterexample.

   (a) **INDIVIDUAL:** For all strings $x,y,z \in \Sigma^*$, we have $|x^R y \circ \varepsilon \circ z| = |x| + |y| + |z|$. (Recall that $\circ$ denotes concatenation.)

   (b) For all languages $L_1, L_2 \subseteq \Sigma^*$, we have $(L_1 \cap L_2)^R = L_1^R \cap L_2^R$.

   (c) For every *finite* language $L \subseteq \Sigma^*$, we have $|L \circ L| = |L|^2$.

   (d) For all languages $L_1, L_2, L_3 \subseteq \Sigma^*$, we have $L_1 \cup (L_2 \circ L_3) = (L_1 \cup L_2) \circ (L_1 \cup L_3)$.

4. Let L be a language consisting of strings over an alphabet Σ. Define

   $$\text{Prefix}(L) = \{x \in \Sigma^* \mid xy \in L \text{ for some } y \in \Sigma^*\}.$$

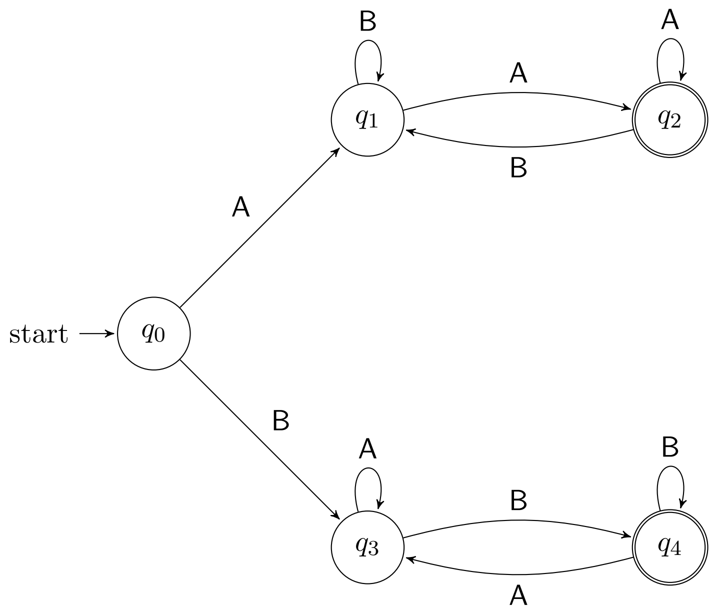
   That is, Prefix(L) consists of all (possibly empty) strings which appear as prefixes of strings in L.

(a) Let $A = \{00, 001, 1011\}$. What is Prefix(A)?

(b) Let $B = \{\mathsf{ab}^n \mid n \geq 0\}$. What is Prefix($B$)?

(c) Prove that if L is a finite language, then Prefix(L) is also finite.

5. 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 sequence of states does the machine go through on input ABBAB?

(e) Does the machine accept the string ABBAA?

(f) Give a simple description of language recognized by M. (Either plain English or set-builder notation is fine.)

6. 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 registrar). We'll provide more specific information about how to register for this tool 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) $L_1 = \{w \mid w \text{ contains the substring } 100\}$.

(b) $L_2 = \{w \mid w \text{ is any string except for } 1001\}$.

(c) $L_3 = \{w \mid \text{every even position of } w \text{ is } 0\}$.

7. **Bonus Problem.** In this problem, you'll explore how to formally define and analyze properties of strings. Let Σ be an alphabet. A *string* over alphabet Σ is defined recursively as follows. It

is either the empty string ε (base case) or takes the form ax where $a \in \Sigma$ and x is itself a string (recursive case). The *length* of the string x can then be defined as follows:

$$|x| = \begin{cases} 0 & \text{if } x = \varepsilon \\ 1 + |z| & \text{if } x = az \text{ for } a \in \Sigma \text{ and } z \in \Sigma^*. \end{cases}$$

Similarly, the concatenation of two strings x, y can be defined as

$$xy = \begin{cases} y & \text{if } x = \varepsilon \\ a(zy) & \text{if } x = az \text{ for } a \in \Sigma \text{ and } z \in \Sigma^*. \end{cases}$$

These definitions let us give inductive proofs of properties of strings. For instance, consider the following claim, which says that the length of the concatenation of two strings is the sum of the lengths of those strings.

Claim. For any two strings x, y, we have $|xy| = |x| + |y|$.

To prove this, let x and y be arbitrary strings. We will prove this by induction on the length $n = |x|$. As our base case, suppose $n = 0$. Then $x = \varepsilon$, so

$$\begin{aligned} |xy| &= |\varepsilon y| && \text{(assumption on } x\text{)} \\ &= |y| && \text{(definition of concatenation)} \\ &= |\varepsilon| + |y| && \text{(definition of length)} \\ &= |x| + |y| && \text{(assumption on } x\text{)} \end{aligned}$$

as we wanted. Now assume as our inductive hypothesis that the claim is true for length n; we want to show it is true for length $n + 1$. In this case, we have $x = az$ for some string z of length n. So

$$\begin{aligned} |xy| &= |a(zy)| && \text{(definition of concatenation)} \\ &= 1 + |zy| && \text{(definition of length)} \\ &= 1 + |z| + |y| && \text{(inductive hypothesis)} \\ &= |x| + |y| && \text{(definition of length).} \end{aligned}$$

(a) Given a string $x \in \{0, 1\}^*$, let $\text{sort}(x)$ denote the string obtained by sorting the characters of x so that all 0's appear before all 1's. For example, $\text{sort}(10110) = 00111$. Give a recursive definition of the sort function along the lines of what we did with length above.

(b) Give an inductive proof that $|\text{sort}(x)| = |x|$ for every string $x \in \{0, 1\}^*$.

(c) Prove that $\text{sort}(\text{sort}(x)) = \text{sort}(x)$ for every string $x \in \{0, 1\}^*$. Hint: Instead of trying to prove this directly by induction, it might be useful to introduce some auxiliary recursively-defined functions and prove statements about those.