CS/ECE 374 A ✧ Fall 2026

Homework 3

Due Tuesday, September 15, 2026 at 9pm Central Time

---

1. Prove that the following languages over the alphabet $\Sigma = \{0, 1\}$ are *not* regular.

   (a) $\left\{0^a 1^b 0^c 1^d \;\middle|\; a, b, c, d \geq 1 \textbf{ and } \text{if } a = d, \text{ then } b = c\right\}$

   (b) All strings in which every prefix has at least as many 1s as 0s. More formally:
   $$\{w \in \Sigma^* \mid \text{for all } x, y \text{ such that } w = xy,\ \#(1, x) \geq \#(0, x)\}$$

   (c) $\{xyx \mid x, y \in \Sigma^* \text{ and } |x| = |y|\}$

   (d) Even-length strings whose second half contains an even number of 1s. More formally:
   $$\{xy \in \Sigma^* \mid |x| = |y| \text{ and } \#(1, y) \text{ is even}\}$$

2. For each of the following languages over the alphabet $\Sigma = \{0, 1\}$, either show that the language is regular (for example, by describing and justifying a DFA, NFA, regular expression, or regularity preserving construction from other regular languages) or prove that the language is not regular.

   Recall w^R denotes the *reversal* of w, and a *run* of a string is defined as a *maximal non-empty* substring in which all symbols are equal.

   (a) All strings with strictly more runs of 1s than runs of 0s.

   (b) All strings with strictly more runs of 1s with length at least two than runs of 0s with length at least two.

   (c) All strings containing an equal but disjoint prefix and suffix of length at least 374. More formally:
   $$\{xyx \in \Sigma^* \mid |x| \geq 374\}$$

   (d) All strings containing with a prefix of length at least 374 and its reversal as a disjoint suffix. More formally:
   $$\left\{xyx^R \in \Sigma^* \;\middle|\; |x| \geq 374\right\}$$

   [Hint: Exactly two of these languages are regular.]

*3. ***Practice only. Do not submit solutions.***

A ***Moore machine*** is a variant of a finite-state automaton that produces output; Moore machines are sometimes called finite-state *transducers*. For purposes of this problem, a Moore machine formally consists of six components:

- A finite set Σ called the input alphabet
- A finite set Γ called the output alphabet
- A finite set Q whose elements are called states
- A start state $s \in Q$
- A transition function $\delta\colon Q \times \Sigma \to Q$
- An output function $\omega\colon Q \to \Gamma$

More intuitively, a Moore machine is a graph with a special start vertex, where every node (state) has one outgoing edge labeled with each symbol from the input alphabet, and each node (state) is additionally labeled with a symbol from the output alphabet.

The Moore machine reads an input string $w \in \Sigma^*$ one symbol at a time. For each symbol, the machine changes its state according to the transition function δ, and then outputs the symbol $\omega(q)$, where q is the new state. Formally, we recursively define a *transducer* function $\omega^*\colon Q \times \Sigma^* \to \Gamma^*$ as follows:

$$\omega^*(q,w) = \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ \omega(\delta(q,a)) \cdot \omega^*(\delta(q,a),x) & \text{if } w = ax \end{cases}$$

Given input string $w \in \Sigma^*$, the machine outputs the string $\omega^*(w,s) \in \Gamma^*$. The ***output language $L^\circ(M)$*** of a Moore machine M is the set of all strings that the machine can output:

$$L^\circ(M) := \{\omega^*(s,w) \mid w \in \Sigma^*\}$$

(a) Let M be an arbitrary Moore machine. Prove that $L^\circ(M)$ is a regular language.

(b) Let M be an arbitrary Moore machine whose input alphabet Σ and output alphabet Γ are identical. Prove that the language

$$L^=(M) = \{w \in \Sigma^* \mid w = \omega^*(s,w)\}$$

is regular. $L^=(M)$ consists of all strings w such that M outputs w when given input w; these are also called *fixed points* for the transducer function ω^*.

[Hint: These problems are easier than they look!]

Solved problems

4. For each of the following languages, either prove that the language is regular (by constructing an appropriate DFA, NFA, or regular expression) or prove that the language is not regular (using a fooling-set argument).

   Recall that a *palindrome* is a string that equals its own reversal: $w = w^R$. Every string of length 0 or 1 is a palindrome.

   (a) Strings in $(0 + 1)^*$ in which no prefix of length at least 2 is a palindrome.

   Solution: Regular: $\varepsilon + 01^* + 10^*$. Call this language L_a.

   Let w be an arbitrary non-empty string in $(0 + 1)^*$. Without loss of generality, assume $w = 0x$ for some string x. There are two cases to consider.

   - If x contains a 0, then we can write $w = 01^n0y$ for some integer n and some string y. The prefix 01^n0 is a palindrome of length at least 2. Thus, $w \notin L_a$.
   - Otherwise, $x \in 1^*$. Every non-empty prefix of w is equal to 01^n for some non-negative integer $n \leq |x|$. Every palindrome that starts with 0 also ends with 0, so the only palindrome prefixes of w are ε and 0, both of which have length less than 2. Thus, $w \in L_a$.

   We conclude that $0x \in L_a$ if and only if $x \in 1^*$. A similar argument implies that $1x \in L_a$ if and only if $x \in 0^*$. Finally, trivially, $\varepsilon \in L_a$. ■

   Rubric: 2½ points = ½ for “regular” + 1 for regular expression + 1 for justification. This is more detail than necessary for full credit.

   (b) Strings in $(0 + 1 + 2)^*$ in which no prefix of length at least 2 is a palindrome.

   Solution: Not regular. Call this language L_b.

   Consider the set $F = (012)^+$.

   Let x and y be arbitrary distinct strings in F.

   Then $x = (012)^i$ and $y = (012)^j$ for some positive integers $i \neq j$.

   Without loss of generality, assume $i < j$.

   Let z be the suffix $(210)^i$.

   - $xz = (012)^i(210)^i$ is a palindrome of length $6i \geq 2$, so $xz \notin L_b$.
   - $yz = (012)^j(210)^i$ has no palindrome prefixes except ε and 0, because $i < j$, so $yz \in L_b$.

   Thus, z is a distinguishing suffix for x and y.

   We conclude that F is a fooling set for L_b.

   Because F is infinite, L_b cannot be regular. ■

   Rubric: 2½ points = ½ for “not regular” + 2 for fooling set proof (standard rubric, scaled).

(c) Strings in $(\mathtt{0}+\mathtt{1})^*$ in which no prefix of length at least 3 is a palindrome.

Solution: Not regular. Call this language L_c.

Consider the set $F = (\mathtt{001101})^+$.

Let x and y be arbitrary distinct strings in F.

Then $x = (\mathtt{001101})^i$ and $y = (\mathtt{001101})^j$ for some positive integers $i \neq j$.

Without loss of generality, assume $i < j$.

Let z be the suffix $(\mathtt{101100})^i$.

- $xz = (\mathtt{001101})^i(\mathtt{101100})^i$ is a palindrome of length $12i \geq 2$, so $xz \notin L_b$.
- $yz = (\mathtt{001101})^j(\mathtt{101100})^i$ has no palindrome prefixes except ε and $\mathtt{0}$ and $\mathtt{00}$, because $i < j$, so $yz \in L_b$.

Thus, z is a distinguishing suffix for x and y.

We conclude that F is a fooling set for L_c.

Because F is infinite, L_c cannot be regular. ■

Rubric: 2½ points = ½ for "not regular" + 2 for fooling set proof (standard rubric, scaled).

(d) Strings in $(\mathtt{0}+\mathtt{1})^*$ in which no *substring* of length at least 3 is a palindrome.

Solution: Regular. Call this language L_d.

Every palindrome of length at least 3 contains a palindrome substring of length 3 or 4. Thus, the complement language $\overline{L_d}$ is described by the regular expression

$$(\mathtt{0}+\mathtt{1})^*(\mathtt{000}+\mathtt{010}+\mathtt{101}+\mathtt{111}+\mathtt{0110}+\mathtt{1001})(\mathtt{0}+\mathtt{1})^*$$

Thus, $\overline{L_d}$ is regular, so its complement L_d is also regular. ■

Solution: Regular. Call this language L_d.

In fact, L_d is *finite*! Appending either $\mathtt{0}$ or $\mathtt{1}$ to any of the underlined strings creates a palindrome suffix of length 3 or 4.

$$\varepsilon+\mathtt{0}+\mathtt{1}+\mathtt{00}+\mathtt{01}+\mathtt{10}+\mathtt{11}+\mathtt{001}+\underline{\mathtt{011}}+\underline{\mathtt{100}}+\mathtt{110}+\underline{\mathtt{0011}}+\underline{\mathtt{1100}}$$

■

Rubric: 2½ points = ½ for "regular" + 2 for proof:

- 1 for expression for $\overline{L_d}$ + 1 for applying closure
- 1 for regular expression + 1 for justification