

CS/ECE 374 A ♦ Fall 2026

Midterm 1 Problem 1 Solution

For each statement below, there are two boxes in the answer booklet labeled “Yes” and “No”. Check “Yes” if the statement is *always* true and “No” otherwise, and give a **brief** (at most one short sentence) explanation of your answer.

Read each statement *very* carefully; some of them are deliberately subtle!

- (a) The regular expression $(01 + 10)^*(0 + 1 + \varepsilon)$ represents a language with an infinite fooling set.

Yes

No

The language is regular, and no regular language has an infinite fooling set.

- (b) The regular expression 0^*1^* represents the language $\{0^n1^n \mid n \geq 0\}$.

Yes

No

This expression allows the two runs to have different lengths.

- (c) For any two languages L_1 and L_2 , if $L_1 \cap L_2$ is regular, then both L_1 and L_2 are regular.

Yes

No

Let $L_1 = \{0^n1^n \mid n \geq 0\}$ and $L_2 = \bar{L}_1 = \Sigma^* \setminus L_1$.

- (d) For any two languages L_1 and L_2 , if $L_1 \cap L_2$ is nonregular, then both L_1 and L_2 are nonregular.

Yes

No

Let $L_1 = \{0^m1^n \mid m, n \geq 0\}$ and $L_2 = \{w \in \{0, 1\}^* \mid \#(0, w) = \#(1, w)\}$.

- (e) For every regular language L , there is a DFA accepting L whose start state is *not* accepting.

Yes

No

The start state must accept if $\varepsilon \in L$.

- (f) For every regular language L , there is an NFA with ε -transitions accepting L whose start state is *not* accepting.

Yes

No

Given one with an accepting start state s , create a new start state $\bar{s}$ with ε -transition from $\bar{s}$ to s .

- (g) For any regular language L , the language $(L^*)^R$ (i.e., the reversal of the Kleene closure of L) is regular.

Yes

No

Regular languages are closed under both Kleene closure and reversal.

- (h) Every context-free grammar has at least two non-terminals.

Yes

No

$S \rightarrow \varepsilon \mid 0S1$ has exactly one non-terminal.

- (i) For any regular language L , its complement $\bar{L} = \Sigma^* \setminus L$ is context-free.

Yes

No

The complement of a regular language is regular and therefore context-free.

- (j) The language $\{0^n1^m0^n \mid n, m \geq 0\}$ is context-free.

Yes

No

This is generated by a grammar with $S \rightarrow A \mid 0S0$ and $A \rightarrow \varepsilon \mid 1A$.

CS/ECE 374 A ♦ Fall 2026
Midterm 1 Problem 2 Solution

Let $L = \{\emptyset^a 1^b 2^c 3^d \mid a, b, c, d \geq 0 \text{ and } (a = d \text{ or } b = c)\}$ be a language over $\{\emptyset, 1, 2, 3\}$.

(a) *Prove* that L is not a regular language.

Solution ($a \stackrel{?}{=} d$): Consider the set $F = \{\emptyset^i 2 \mid i \geq 0\}$.

Let x and y be arbitrary distinct strings in F .

Then $x = \emptyset^i 2$ and $y = \emptyset^j 2$ for some non-negative integers $i \neq j$.

Let $z = 3^i$.

- $xz = \emptyset^i 2 3^i = \emptyset^i 1^0 2^1 3^i \in L_a$.
- $yz = \emptyset^j 2 3^i = \emptyset^j 1^0 2^1 3^i \notin L_a$.

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

We conclude that F is an infinite fooling set for L , and L is not regular. ■

Solution ($b \stackrel{?}{=} c$): Consider the set $F = \{1^i \mid i \geq 0\}$.

Let x and y be arbitrary distinct strings in F .

Then $x = 1^i$ and $y = 1^j$ for some non-negative integers $i \neq j$.

Let $z = 2^i 3$.

- $xz = 1^i 2^i 3 = \emptyset^0 1^i 2^i 3^1 \in L_a$.
- $yz = 1^j 2^i 3 = \emptyset^0 1^j 2^i 3^1 \notin L_a$.

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

We conclude that F is an infinite fooling set for L , and L is not regular. ■

Rubric: 5 points: standard fooling set rubric (scaled).

- (b) Describe a context-free grammar for L . You do not need to prove your context-free grammar is correct.

Solution:

$$S \rightarrow A \mid C$$

L

$$A \rightarrow B \mid 0A3$$

$$\{0^a 1^b 2^c 3^d \mid a, b, c, d \geq 0 \text{ and } a = d\}$$

$$B \rightarrow \varepsilon \mid 1B \mid B2$$

$$\{1^b 2^c \mid b, c \geq 0\}$$

$$C \rightarrow D \mid 0C \mid C3$$

$$\{0^a 1^b 2^c 3^d \mid a, b, c, d \geq 0 \text{ and } b = c\}$$

$$D \rightarrow \varepsilon \mid 1D2$$

$$\{1^b 2^c \mid a, b, c, d \geq 0 \text{ and } b = c\}$$

Yes, this grammar is ambiguous. Its ambiguity does not affect credit. ■

Rubric: 5 points, standard context free grammar rubric (scaled).

CS/ECE 374 A ♦ Fall 2026

Midterm 1 Problem 3 Solution

Consider the function fence defined in the question handout.

- (a) **Prove** that $\text{fence}(x \bullet y) = \text{fence}(x) \bullet \text{fence}(y)$ for all strings x and y over $\{0, 1\}$.

Solution: Let x and y be arbitrary strings over $\{0, 1\}$.

Assume for every string $z \in \{0, 1\}^*$ such that $|z| < |x|$, we have $\text{fence}(z \bullet y) = \text{fence}(z) \bullet \text{fence}(y)$.

There are three cases to consider.

- Suppose $x = \varepsilon$. Then,

$$\begin{aligned}
 \text{fence}(x \bullet y) &= \text{fence}(\varepsilon \bullet y) & x &= \varepsilon \\
 &= \text{fence}(y) & \text{def. } (\bullet) \\
 &= \varepsilon \bullet \text{fence}(y) & \text{def. } (\bullet) \\
 &= \text{fence}(\varepsilon) \bullet \text{fence}(y) & \text{def. fence} \\
 &= \text{fence}(x) \bullet \text{fence}(y) & x &= \varepsilon
 \end{aligned}$$

- Suppose $x = 0 \cdot z$ for some string z . Then,

$$\begin{aligned}
 \text{fence}(x \bullet y) &= \text{fence}((0 \cdot z) \bullet y) & x &= 0 \cdot z \\
 &= \text{fence}(0 \cdot (z \bullet y)) & \text{def. } (\bullet) \\
 &= 101 \cdot \text{fence}(z \bullet y) & \text{def. fence} \\
 &= 101 \cdot (\text{fence}(z) \bullet \text{fence}(y)) & \text{ind. hypo.} \\
 &= (101 \cdot \text{fence}(z)) \bullet \text{fence}(y) & \text{def. } (\bullet) \\
 &= \text{fence}(0 \cdot z) \bullet \text{fence}(y) & \text{def. fence} \\
 &= \text{fence}(x) \bullet \text{fence}(y) & x &= 0 \cdot z
 \end{aligned}$$

- Suppose $x = 1 \cdot z$ for some string z . Then,

$$\begin{aligned}
 \text{fence}(x \bullet y) &= \text{fence}((1 \cdot z) \bullet y) & x &= 1 \cdot z \\
 &= \text{fence}(1 \cdot (z \bullet y)) & \text{def. } (\bullet) \\
 &= 1 \cdot \text{fence}(z \bullet y) & \text{def. fence} \\
 &= 1 \cdot (\text{fence}(z) \bullet \text{fence}(y)) & \text{ind. hypo.} \\
 &= (1 \cdot \text{fence}(z)) \bullet \text{fence}(y) & \text{def. } (\bullet) \\
 &= \text{fence}(1 \cdot z) \bullet \text{fence}(y) & \text{def. fence} \\
 &= \text{fence}(x) \bullet \text{fence}(y) & x &= 1 \cdot z
 \end{aligned}$$

In every case, $\text{fence}(x \bullet y) = \text{fence}(x) \bullet \text{fence}(y)$. ■

Rubric: 5 points: standard induction rubric (scaled). This solution is more detailed than necessary for full credit.

(b) **Prove** that $\#(1, \text{fence}(w)) \geq 2 \cdot \#(0, \text{fence}(w))$ for all strings w over $\{0, 1\}$.

Solution: Let w be an arbitrary string over $\{0, 1\}$.

Assume for every string $x \in \{0, 1\}^*$ such that $|x| < |w|$, we have $\#(1, \text{fence}(x)) \geq 2 \cdot \#(0, \text{fence}(x))$.

There are three cases to consider.

- Suppose $w = \varepsilon$. Then,

$$\begin{aligned}
 \#(1, \text{fence}(w)) &= \#(1, \text{fence}(\varepsilon)) & w = \varepsilon \\
 &= \#(1, \varepsilon) & \text{def. fence} \\
 &= 0 & \text{def. } (\#) \\
 &= 2 \cdot 0 & \text{arith.} \\
 &= 2 \cdot \#(0, \varepsilon) & \text{def. } (\#) \\
 &= 2 \cdot \#(0, \text{fence}(\varepsilon)) & \text{def. fence} \\
 &= 2 \cdot \#(0, \text{fence}(w)) & w = \varepsilon
 \end{aligned}$$

- Suppose $w = 0 \cdot x$ for some string x . Then,

$$\begin{aligned}
 \#(1, \text{fence}(w)) &= \#(1, \text{fence}(0 \cdot x)) & w = 0 \cdot x \\
 &= \#(1, 101 \cdot \text{fence}(x)) & \text{def. fence} \\
 &= 2 + \#(1, \text{fence}(x)) & \text{def. } (\#) \times 3 \\
 &\geq 2 + 2 \cdot \#(0, \text{fence}(x)) & \text{ind. hypo.} \\
 &= 2 \cdot (1 + \#(0, \text{fence}(x))) & \text{arith.} \\
 &= 2 \cdot \#(0, 101 \cdot \text{fence}(x)) & \text{def. } (\#) \times 3 \\
 &= 2 \cdot \#(0, \text{fence}(0 \cdot x)) & \text{def. fence} \\
 &= 2 \cdot \#(0, \text{fence}(w)) & w = 0 \cdot x
 \end{aligned}$$

- Suppose $w = 1 \cdot x$ for some string z . Then,

$$\begin{aligned}
 \#(1, \text{fence}(w)) &= \#(1, \text{fence}(1 \cdot x)) & w = 1 \cdot x \\
 &= \#(1, 1 \cdot \text{fence}(x)) & \text{def. fence} \\
 &= 1 + \#(1, \text{fence}(x)) & \text{def. } (\#) \\
 &\geq 1 + 2 \cdot \#(0, \text{fence}(x)) & \text{ind. hypo.} \\
 &> 2 \cdot \#(0, \text{fence}(x)) & \text{arith.} \\
 &= 2 \cdot \#(0, 1 \cdot \text{fence}(x)) & \text{def. } (\#) \\
 &= 2 \cdot \#(0, \text{fence}(1 \cdot x)) & \text{def. fence} \\
 &= 2 \cdot \#(0, \text{fence}(w)) & w = 1 \cdot x
 \end{aligned}$$

In every case, $\#(1, \text{fence}(w)) \geq 2 \cdot \#(0, \text{fence}(w))$. ■

Rubric: 5 points: standard induction rubric (scaled). This solution is more detailed than necessary for full credit.

CS/ECE 374 A ♦ Fall 2026

Midterm 1 Problem 4 Solution

For any string $w \in \{0, 1\}^*$, let $\text{Delete1st0}(w)$ denote the string obtained by deleting the first 0 in w if a 0 exists. Let L be an arbitrary regular language over the alphabet $\Sigma = \{0, 1\}$. **Prove** that the following languages are also regular.

(a) $\text{INSERTED1ST0}(L) = \{w \in \Sigma^* \mid \text{Delete1st0}(w) \in L\}$

Solution: Given an arbitrary DFA $M = (Q, \delta, s, A, \delta)$ for L , we construct an NFA $M' = (Q', \delta', s', A')$ for $\text{INSERTED1ST0}(L)$ as follows.

M' will simulate M until it encounters its first 0, drop that 0, and then continue simulating M faithfully.

$$\begin{aligned}
 Q' &= Q \times \{\text{waiting}, \text{seen}\} \\
 s' &= (s, \text{waiting}) \\
 A' &= A \times \{\text{waiting}, \text{seen}\} \\
 \delta'((q, \text{waiting}), 0) &= \{(q, \text{seen})\} && \text{for all } q \in Q \\
 \delta'((q, \text{waiting}), 1) &= \{(\delta(q, 1), \text{waiting})\} && \text{for all } q \in Q \\
 \delta'((q, \text{seen}), a) &= \{(\delta(q, a), \text{seen})\} && \text{for all } q \in Q, a \in \{0, 1\}
 \end{aligned}$$

More specifically, string x takes M' to state $(q, \text{waiting})$ if and only if x contains no 0s, and x takes M to state q . String x takes M' to state (q, seen) if and only if x contains at least one 0, and $\text{Delete1st0}(x)$ takes M to state q .

Yes, this could have been a DFA. ■

Rubric: 5 points: standard language transformation rubric (scaled).

(b) $\text{DELETED1ST0}(L) = \{\text{Delete1st0}(w) \mid w \in L\}$

Solution: Given an arbitrary DFA $M = (Q, \delta, s, A, \delta)$ for L , we construct an NFA $M' = (Q', \delta', s', A')$ with ε -transitions for $\text{DELETED1ST0}(L)$ as follows.

M' will simulate M while using ε -transitions to guess where to toss in a single deleted 0, making sure to take a guess before reading the first 0 that isn't deleted.

$$\begin{aligned}
 Q' &= Q \times \{\text{guessing}, \text{done}\} \\
 s' &= (s, \text{guessing}) \\
 A' &= A \times \{\text{guessing}, \text{done}\} \\
 \delta'((q, \text{guessing}), 0) &= \emptyset && \text{for all } q \in Q \\
 \delta'((q, \text{guessing}), 1) &= \{(\delta(q, 1), \text{guessing})\} && \text{for all } q \in Q \\
 \delta'((q, \text{guessing}), \varepsilon) &= \{(\delta(q, 0), \text{done})\} && \text{for all } q \in Q \\
 \delta'((q, \text{done}), a) &= \{(\delta(q, a), \text{done})\} && \text{for all } q \in Q, a \in \{0, 1\} \\
 \delta'((q, \text{done}), \varepsilon) &= \emptyset && \text{for all } q \in Q
 \end{aligned}$$

More specifically, string $\text{Delete1st0}(x)$ takes M' to state $(q, \text{guessing})$ if and only if x contains no 0s, and x takes M to state q . String $\text{Delete1st0}(x)$ takes M' to state (q, done) if and only if x contains at least one 0, and x takes M to state q . ■

Rubric: 5 points: standard language transformation rubric (scaled).

CS/ECE 374 A ♦ Fall 2026
Midterm 1 Problem 5 Solution

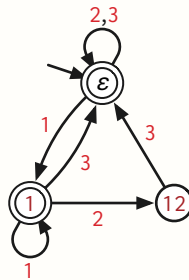
For each of the following languages L over the alphabet $\{1, 2, 3\}$, describe a DFA that accepts L **and** give a regular expression that represents L . **Give your DFA states mnemonic names.** You do not need to do more to prove your answers are correct.

- (a) All strings in which every instance of the substring **12** is immediately followed by a **3**.

Solution:

$$(2 + 3)^* (11^*(23 + 3)(2 + 3)^*)^* (\varepsilon + 11^*)$$

Each time we see a run of **1**s, it must either be the last run in the string, followed by a **3** (so **12** doesn't appear), or followed by a **23**. Outside of this, we can have any combination of **2**s and **3**s.



All missing transitions in the DFA go to a hidden dump state.

The states are labeled based on the last few symbols read assuming we have not yet failed to follow an instance of **12** with a **3**:

- **1**: We just read **1**.
- **12**: We just read **12**.
- ε : We did not just read **1** nor **12**.



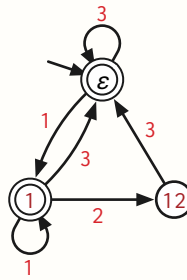
Rubric: 5 points = 2½ for regular expression + 2½ for DFA, standard rubrics (scaled). The state labels and explanations are not required for credit.

- (b) All strings in which every 2 is immediately **preceded** by a 1 and immediately **followed** by a 3 (i.e., every 2 belongs to an instance of the substring 123.)

Solution:

$$(1 + 3)^* (123 (1 + 3)^*)^* (1 + 3)^*$$

Each time we see a 2, it must be part of a substring 123. Outside of this, we can have any combination of 1s and 3s.



All missing transitions in the DFA go to a hidden dump state.

The states are labeled based on the last few symbols read assuming we have not yet failed to include a 2 in the substring 123:

- 1: We just read 1.
- 12: We just read 12.
- ε: We did not just read 1 or 12.

Note that in contrast to the solution to part (a), a 2 takes state ε to the dump state. ■

Rubric: 5 points = 2½ for regular expression + 2½ for DFA, standard rubrics (scaled). The state labels and explanations are not required for credit.