You have 120 minutes to answer five questions.

Write your answers in the separate answer booklet.

Please return this question sheet and your cheat sheet with your answers.

1. Let compress0s(w) be a function that takes a string w as input, and returns the string formed by compressing every run of 0s in w by half. Specifically, every run of $2n$ 0s is compressed to length n, and every run of $2n+1$ 0s is compressed to length $n+1$. For example:

LT

10000 ↓ 100

1000 ↳ 100

$$\text{compress0s}(\underline{00000}11\underline{000}1) = \underline{000}11\underline{00}1$$
$$\text{compress0s}(11\underline{0000}1\underline{0}) = 11\underline{00}1\underline{0}$$
$$\text{compress0s}(11111) = 11111$$

10 ↳ 10

Let L be an arbitrary regular language.

   (a) ***Prove*** that $\textsc{Decompressed}(L) := \{w \in \Sigma^* \mid \text{compress0s}(w) \in L\}$ is regular.

   (b) ***Prove*** that $\textsc{Compressed}(L) := \{\text{compress0s}(w) \mid w \in L\}$ is regular.

2. Let L be the language of all strings over $\{0, 1\}$ that contain at least 374 consecutive 1s.

   (a) Give a regular expression that matches L.

   You may use the notation R^k to denote the concatenation of k copies of the regular expression R; for example,

$$(1 + 01)^5 = (1 + 01)(1 + 01)(1 + 01)(1 + 01)(1 + 01)$$

   (b) Describe a DFA whose language is L. *[Hint: Do not try to **draw** your DFA!]*

   (c) ***Prove*** that any DFA whose language is L must have at least 375 states. *[Hint: What must be true if L has a fooling set of some size ℓ?]*

3. Consider the following recursive function Bond, which doubles the length of any run of 0s in its input string.

S.I.

$$\text{Bond}(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ 00 \cdot \text{Bond}(x) & \text{if } w = 0 \cdot x \text{ for some string } x \\ 1 \cdot \text{Bond}(x) & \text{if } w = 1 \cdot x \text{ for some string } x \end{cases}$$

   (a) ***Prove*** that $|\text{Bond}(w)| \geq |w|$ for all strings w.

   (b) ***Prove*** that $\text{Bond}(x \bullet y) = \text{Bond}(x) \bullet \text{Bond}(y)$ for all strings x and y.

As usual, you can assume any result proved in class, in the lecture notes, in labs, or in homework solutions.

4. Let L be the language $\left\{0^a 1^b 0^c \mid a = b \text{ or } a = c \text{ or } b = c\right\}$

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

   (b) Describe a context-free grammar for L. You do not need to justify your answer.

5. For each statement below, check "True" if the statement is always true and check "False" otherwise, and give a brief (one short sentence) explanation of your answer. Read these statements very carefully—small details matter!

   For any string $w \in (0+1)^*$, let w^c denote the *bitwise complement* of w, obtained by flipping every 0 in w to a 1, and vice versa. For example, $\varepsilon^c = \varepsilon$ and $000110^c = 111001$.

   (a) If $2 + 2 = 5$, then zero is odd. T (vacuously true)

   (b) $\{0^n 1 \mid n > 0\}$ is the only infinite fooling set for the language $\{0^n 1 0^n \mid n > 0\}$. F $\{0^n \mid n > 0\}$

   $10^i\ 10^j$

   (c) $\{0^n 1 0^n \mid n > 0\}$ is a context-free language.

   (d) The context-free grammar $S \to 00S \mid S11 \mid 01$ generates the language $\{0^n 1^n \mid n \geq 0\}$. ε 0001 F

   (e) Every regular language is recognized by a DFA with exactly one accepting state.

   (f) Any language that can be decided by an NFA with ε-transitions can also be decided by an NFA without ε-transitions. T

   (g) If L is a regular language over the alphabet $\{0, 1\}$, then $\{x y^c \mid x, y \in L\}$ is also regular.

   (h) If L is a regular language over the alphabet $\{0, 1\}$, then $\{w w^c \mid w \in L\}$ is also regular. $\to \{0^n 1^n\}$ F

   (i) The regular expression $(00 + 11)^*$ represents the language of all strings over $\{0, 1\}$ of even length. F 01 10

   (j) Let L_1 and L_2 be two regular languages. The language $(L_1 \cup L_2)^*$ is also regular.

   T $\to$ closure proportios

CS/ECE 374 A ✧ Fall 2026

Midterm 1 Practice 2

September 25, 2026

Name:	Aashima Sisodia
NetID:	aashima2

- ***Don't panic!***
- You have 120 minutes to answer five questions. **The questions are described in more detail in a separate handout.**
- If you brought anything except your writing implements, your **hand-written** double-sided 8½" × 11" cheat sheet, and your university ID, please put it away for the duration of the exam. In particular, please turn off and put away *all* medically unnecessary electronic devices.
- Please clearly print your name and your NetID in the boxes above.
- Please also print your name at the top of every page of the answer booklet, except this cover page. We want to make sure that if a staple falls out, we can reassemble your answer booklet. (It doesn't happen often, but it does happen.)
- Proofs or other justifications beyond those listed in the standard grading rubrics are required if and only if we explicitly ask for them, using the word ***prove*** or ***justify*** in bold italics. **In particular, DFAs need to have mnemonic state names.**

- **Do not write outside the black boxes on each page**. These indicate the area of the page that our scanners will actually scan. If the scanner can't see your work, we can't grade it.
- If you run out of space for an answer, feel free to use the overflow/scratch pages at the back of the answer booklet, but **clearly indicate where we should look**. If we can't find your work, we can't grade it.
- **Only work that is written into the stapled answer booklet will be graded**. We **do not** recommend detaching pages from the answer booklet, because any work on those detached pages will not be graded (and it disrupts our process for uploading solutions to Gradescope.) Please let us know if you detach a page either on purpose or accidentally so we can reattach it.
- We will provide additional blank scratch paper on request, but we will not grade anything written on it.
- **Please return all paper with your answer booklet** including your question sheet, your cheat sheet, and all scratch paper. Please put this loose paper **inside** your answer booklet.

Is between
between is and between
or between between and is
or between and and or
or between or and between
or between between and and
or between or and and
or between or and or?

CS/ECE 374 A ✧ Fall 2026 Midterm 1 Practice 2 Problem 1	Name: Aashima Sisodia (aashima2)

Consider the function compress0s defined in the question handout. Let L be an arbitrary regular language.

100 1000 10 100

(a) ***Prove*** that $\textsc{Decompressed}(L) := \{ w \in \Sigma^* \mid \text{compress0s}(w) \in L \}$ is regular.

10 10 ∈ L

(b) ***Prove*** that $\textsc{Compressed}(L) := \{ \text{compress0s}(w) \mid w \in L \}$ is regular.

100 1000 100000 ∈ L 1000

(a) Let $M = (Q, \delta, s, A)$ be a DFA accepting L.

Let $M' = (Q', \delta', s', A')$ be a DFA that accepts DECOMPRESSED(L). M' reads in w and compresses the string.

$$Q' = Q \times \{0, 1\}$$

0 represents no consecutive 0's

1 represents character that's going to be deleted

$$s' = (s, 0)$$

$$A' = A \times \{0, 1\}$$

$$\delta'((q, n), 1) = (\delta(q, 1), 0) \text{ for } n \in \{0, 1\}$$

$$\delta'((q, 0), 0) = (\delta(q, 0), 1)$$

$$\delta'((q, 1), 0) = (q, 0) \quad \text{for } q \in Q$$

☆ 1b on page 6

CS/ECE 374 A ✧ Fall 2026 Midterm 1 Practice 2 Problem 2	Name:

Let L be the language of all strings over $\{0, 1\}$ that contain at least 374 consecutive 1s.

(a) Give a regular expression that matches L. You may use the notation R^k to denote the concatenation of k copies of the regular expression R.

(b) Describe a DFA whose language is L.

(c) ***Prove*** that any DFA whose language is L must have at least 375 states.

(a) $(0+1)^* 1^{374} (0+1)^*$

(b) $Q = \{0, \dots, 374\}$

$s = 0$

$A = \{374\}$

$\delta(q, 1) = q+1$ for $q \in \{0, \dots, 373\}$

$\delta(q, 0) = 0$ for $q \in \{0, \dots, 373\}$

$\delta(374, a) = 374$ for $a \in \Sigma$

(c) We construct a finite fooling set

$F = \{1^i \mid 0 \le i < 375\}$.

Let $x = 1^i$ and $y = 1^j$ where $j \ne i$ and WLOG $i > j$.

Let $z = 1^{374-i}$.

$xz = (1^i)(1^{374-i}) = 1^{374-i+i} = 1^{374} \in L$

$yz = (1^j)(1^{374-i}) = 1^{\underbrace{374-i+j}} \notin L$ since $374-i+j < 374$

Thus, F is a fooling set for L. Since $|F| = 375$, a DFA whose language is L must also have at least 375 states. $\square$

CS/ECE 374 A ✧ Fall 2026 Midterm 1 Practice 2 Problem 3	Name:

Consider the recursive function Bond defined in the question handout.

$$\text{Bond}(w) = \begin{cases} \varepsilon & w = \varepsilon \\ 00 \cdot \text{Bond}(x) & w = 0x \\ 1 \cdot \text{Bond}(x) & w = 1x \end{cases}$$

(a) ***Prove*** that $|\text{Bond}(w)| \geq |w|$ for all strings w.

(b) ***Prove*** that $\text{Bond}(x \bullet y) = \text{Bond}(x) \bullet \text{Bond}(y)$ for all strings x and y.

(a) Let w be an arbitrary string.
Assume, for every string x such that $|x| < |w|$, that $|\text{Bond}(x)| \geq |x|$. (I.H.)
There are 3 cases to consider

- (Base case) $w = \varepsilon$. Then,

$\text{Bond}(w) = \varepsilon$ By def. of Bond
$|\text{Bond}(w)| = |\varepsilon|$ since $\text{Bond}(w) = \varepsilon$
$|\text{Bond}(w)| = |w|$ since $w = \varepsilon$
$|\text{Bond}(w)| \geq |w|$ since $|\text{Bond}(w)| = |w|$

- $w = 0x$. Then

$\text{Bond}(w) = 00 \cdot \text{Bond}(x)$ def.
$|\text{Bond}(w)| = |00 \cdot \text{Bond}(x)|$
$= |00| + |\text{Bond}(x)|$ $\quad |x \cdot y| = |x| + |y|$
$= 2 + |\text{Bond}(x)|$
$|w| = |0x| = |0| + |x| = 1 + |x|$
$2 + |\text{Bond}(x)| \geq 2 + |x|$ (I.H)
$\geq 1 + |w|$
$|\text{Bond}(w)| \geq 1 + |w|$
$|\text{Bond}(w)| \geq |w|$

CASE 3 pg 7

CS/ECE 374 A ✧ Fall 2026 Midterm 1 Practice 2 Problem 4	Name:

Let L be the language $\{0^a 1^b 0^c \mid a = b \text{ or } a = c \text{ or } b = c\}$

1. ***Prove*** that L is *not* a regular language. $a \neq b \neq c$

2. Describe a context-free grammar for L. You do not need to justify your answer.

---

1. Let F be the infinite set $\{0^i 1 \mid i > 1\}$.

Let x and y be arbitratry distinct strings in F.

Then $x = 0^i 1$ and $y = 0^j 1$ where $i \neq j$ and $j > 1$

Let $z = 0^i$

- $xz = 0^i 1 0^i \in L$
- $yz = 0^i 1 0^j$. $j \neq 1$, $i \neq 1$, and $i \neq j$ so $yz \notin L$.

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

We conclude that F is a fooling set for L.

Because F is infinite, L cannot be regular. $\square$

2. $S \to A \mid B \mid C$

$\boxed{0^n 1^n} 0^*$

$A \to XZ$

$X \to 0X1 \mid \varepsilon$

$Z \to 0Z \mid \varepsilon$

$0^n 1^* 0^n$

$B \to 0B0 \mid Y$

$Y \to 1Y \mid \varepsilon$

$0^* \boxed{1^n 0^n}$

$C \to ZW$

$W \to 1W0 \mid$

CS/ECE 374 A ✧ Fall 2026 Midterm 1 Practice 2 Problem 5	Name:

For each statement below, check "Yes" if the statement is ***always*** true and check "No" otherwise, and write a ***brief*** (one short sentence) explanation of your answer. Read these statements very carefully—small details matter!

For any string $w \in (0+1)^*$, let w^c denote the *bitwise complement* of w, obtained by flipping every 0 in w to a 1, and vice versa. For example, $\varepsilon^C = \varepsilon$ and $000110^c = 111001$.

(a) If $2+2=5$, then zero is odd.

[✓] Yes [ ] No — Vacuously true

(b) $\{0^n 1 \mid n > 0\}$ is the only infinite fooling set for the language $\{0^n 1 0^n \mid n > 0\}$.

[ ] Yes [✓] No — $\{0^n\}$ is also an infinite fooling set

(c) $\{0^n 1 0^n \mid n > 0\}$ is a context-free language.

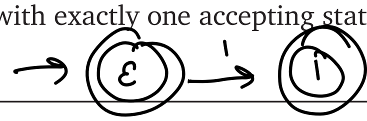
[✓] Yes [ ] No — $S \to 0S0 \mid 010$

(d) The context-free grammar $S \to 00S \mid S11 \mid 01$ generates the language $\{0^n 1^n \mid n \geq 0\}$.

[ ] Yes [✓] No — 0001, 0111

(e) Every regular language is recognized by a DFA with exactly one accepting state.

[ ] Yes [✓] No — $L = \{\varepsilon, 1\}$



(f) Any language that can be decided by an NFA with ε-transitions can also be decided by an NFA without ε-transitions.

[✓] Yes [ ] No — ε-transitions can always be eliminationed without changing the language.

(g) If L is a regular language over the alphabet $\{0, 1\}$, then $\{xy^c \mid x, y \in L\}$ is also regular.

[✓] Yes [ ] No — $\hookrightarrow L \cdot \boxed{L^c}$ is regular (closure)

(h) If L is a regular language over the alphabet $\{0, 1\}$, then $\{ww^c \mid w \in L\}$ is also regular.

[ ] Yes [✓] No — $\{0^n 1^n\}$ is not regular

(i) The regular expression $(00+11)^*$ represents the language of all strings over $\{0, 1\}$ of even length.

[ ] Yes [✓] No — $01, 10 \notin (00+11)^*$

(j) Let L_1, L_2 be two regular languages. The language $(L_1 \cup L_2)^*$ is also regular.

[✓] Yes [ ] No — Closure properties. $L_1 \cup L_2$ is regular, and Kleene star of a regular language is also regular.

(1b) Let DFA $M = (Q, s, \delta, A)$ be the DFA for L. We construct NFA $M' = (Q', s', \delta', A')$ that accepts COMPRESSED(L). It simulates adding in the omitted/compressed 0's.

in the run of 0's / the last 0 in run / haven't read in consecutive 0's

$$Q' = Q \times \{0, 1\} \cup \{\emptyset\}$$

NEEDS DUMP STATE!!

$$s' = (s, 0)$$

$$A' = A \times \{0, 1\}$$

$$\delta'((q, n), 1) = \{(\delta(q, 1), 0)\} \text{ for } n \in \{0, 1\}$$

$$\delta'((q, 0), 0) = \{(\delta(q, 0), 1), (\delta^*(q, 00), 0)\}$$

$$\delta'((q, 1), 0) = \{\emptyset\}$$

All ommitted transition go to $\emptyset$.

• $w = 1x$. Then

$$Bond(w) = 1 \cdot Bond(x)$$

$$|Bond(w)| = |1 \cdot Bond(x)|$$

$$= |1| + |Bond(x)|$$

$$|Bond(w)| = 1 + |Bond(x)| \quad (I.H.)$$

$$|Bond(w)| \geq \underbrace{1 + |x|}$$

$$|Bond(w)| \geq |w| \qquad \text{since } 1+|x| = |w|$$

In all cases, we conclude that

$|Bond(w)| \geq |w|$.

(b) Let w be an arbitrary string.

Assume for every string x such that $|x| < |w|$, that $Bond(x \cdot y) = Bond(x) \circ Bond(y)$ for all strings y. (I.H.)

There are 3 cases to consider

- (Base case) $w = \varepsilon$. Then,

$$\begin{aligned} Bond(\varepsilon \cdot y) &= Bond(y) \\ &= \varepsilon \circ Bond(y) \\ &= Bond(\varepsilon) \circ Bond(y) \\ &= Bond(w) \circ Bond(y). \end{aligned}$$

- $w = 0x$. Then,

$$\begin{aligned} Bond(w \circ y) &= Bond(0x \circ y) \\ &= 00 \circ Bond(x \circ y) \\ &= \underline{00 \circ Bond(x)} \circ Bond(y) \quad \text{(I.H.)} \\ &= Bond(w) \circ Bond(y) \end{aligned}$$

- $w = 1x$. Then,

$$\begin{aligned} Bond(w \circ y) &= Bond(1x \cdot y) \\ &= 1 \circ Bond(x \cdot y) \\ &= \underline{1 \circ Bond(x)} \cdot Bond(y) \quad \text{(I.H.)} \\ &= Bond(w) \circ Bond(y). \end{aligned}$$

In all cases, we conclude that

$Bond(w \circ y) = Bond(w) \circ Bond(y)$.