CS/ECE 374 A ✧ Fall 2023 Midterm 1 Solutions Problem 1	Name:

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

(a) Every irregular language is infinite.

[X] Yes [ ] No — Every finite language is regular.

(b) The language $(0 + 1(01^*0)^*1)^*$ is context-free.

[X] Yes [ ] No — Every regular language is context-free.

(c) Every subset of a regular language is regular.

[ ] Yes [X] No — Every irregular language is a subset of the regular language Σ^*

(d) The language $\{0^a 1^b \mid a + b \text{ is divisible by } 374\}$ is regular.

[X] Yes [ ] No — Intersection of regular languages 0^*1^* and $((0 + 1)^{374})^*$

(e) If language L is regular, then L has a finite fooling set.

[X] Yes [ ] No — Trivially; $\emptyset$ is a finite fooling set for every language

(f) For every language L, if for every string $w \in L$ there is a DFA that accepts w, then L is regular.

[ ] Yes [X] No — We need a single DFA that accepts every string in L (and nothing else).

(g) If language L is accepted by an NFA with n states, then its complement $\Sigma^* \setminus L$ is also accepted by an NFA with n states.

[ ] Yes [X] No — $\{\varepsilon\}$ is accepted by an NFA with 1 state, but $\Sigma^* \setminus \{\varepsilon\}$ requires at least 2 states.

(h) 0^*1^* is a fooling set for the language $\left\{0^i 1^j 0^{i+j} \;\middle|\; i, j \geq 0\right\}$.

[ ] Yes [X] No — Can't distinguish 000111 and 011111.

(i) Every regular language is accepted by a DFA with an odd number of accepting states.

[X] Yes [ ] No — We can always add an inaccessible accepting state.

(j) The context-free grammar $S \to 1T \mid T1 \mid \varepsilon$; $T \to 0S \mid S0$ generates all strings in which the number of 0s equals the number of 1s.

[ ] Yes [X] No — Cannot generate 11000011.

Rubric: 10 points = ½ for each correct answer + ½ for each correct explanation of a correct answer. These are not the only correct explanations. No negative scores for incorrect answers.

CS/ECE 374 A ✧ Fall 2023 Midterm 1 Solutions Problem 2	Name:

Let L be the set of all strings in $\{0,1\}^*$ in which every run of 0s is followed immediately by a longer run of 1s.

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

(b) Describe a context-free grammar for L.

Solution (a): Let $F = 0^*$.

Let x and y be arbitrary distinct strings in F.
Then $x = 0^i$ and $y = 0^j$ for some non-negative integers $i \neq j$.
Assume $i < j$ (otherwise swap variable names).
Let $z = 1^j$.
Then $xz = 0^i 1^j \in L$, because $i < j$.
But $yz = 0^j 1^j \notin L$, because the run of 1s has the same length as the run of 0s.
We conclude that F is a fooling set for L, and F is infinite, so L cannot be regular. ■

Rubric: 5 points: standard fooling set rubric. This is not the only correct solution. Boilerplate in gray is ***not*** required.

Solution (b):

$S \to SL \mid R$	RL^*
$L \to L1 \mid E1$	run of 0s followed by Longer run of 1s
$E \to 0E1 \mid 01$	run of 0s followed by Equal-length run of 1s
$R \to 1R \mid \varepsilon$	1^* = optional initial Run of 1s

■

Rubric: 5 points. −1 for a minor mistake. 2 points for significant progress. This is not the only correct solution. Nonterminal explanations (in gray) are ***not*** required.

CS/ECE 374 A ✧ Fall 2023 Midterm 1 Solutions Problem 3	Name:

Consider the recursive function sortpairs defined in the question handout.

(a) ***Prove*** that $\#(1, \text{sortpairs}(w)) = \#(1, w)$ for every string w.

(b) ***Prove*** that $\text{sortpairs}(\text{sortpairs}(w)) = \text{sortpairs}(w)$ for every string w.

Solution (a): Let w be an arbitrary string.
Assume $\#(1, \text{sortpairs}(x)) = \#(1, x)$ for every string x shortest than w.
There are two cases to consider:

- If $|w| \le 1$, we immediately have $\#(1, \text{sortpairs}(w)) = \#(1, w)$ by definition of sortpairs.
- If $w = 10x$ for some string x then

$$\begin{aligned}
\#(1, \text{sortpairs}(w)) &= \#(1, \text{sortpairs}(10x)) && \text{because } w = 10x \\
&= \#(1, 01 \cdot \text{sortpairs}(x)) && \text{def. sortpairs} \\
&= \#(1, 01) + \#(1, \text{sortpairs}(x)) && \#(1, yz) = \#(1, y) + \#(1, z) \\
&= 1 + \#(1, \text{sortpairs}(x)) && \text{def. } \# \\
&= 1 + \#(1, x) && \textit{induction hypothesis} \\
&= \#(1, 10) + \#(1, x) && \text{def. } \# \\
&= \#(1, 10x) && \#(1, yz) = \#(1, y) + \#(1, z) \\
&= \#(1, w) && \text{because } w = 10x
\end{aligned}$$

- Otherwise, $w = abx$ for some string x and symbols a, b such that $ab \neq 10$. Then

$$\begin{aligned}
\#(1, \text{sortpairs}(w)) &= \#(1, \text{sortpairs}(abx)) && \text{because } w = abx \\
&= \#(1, ab \cdot \text{sortpairs}(x)) && \text{def. sortpairs and } ab \neq 10 \\
&= \#(1, ab) + \#(1, \text{sortpairs}(x)) && \#(1, yz) = \#(1, y) + \#(1, z) \\
&= \#(1, ab) + \#(1, x) \quad \text{[IH]} && \textit{induction hypothesis} \\
&= \#(1, abx) && \#(1, yz) = \#(1, y) + \#(1, z) \\
&= \#(1, w) && \text{because } w = abx
\end{aligned}$$

■

Rubric: 5 points: standard induction rubric. Explanations for each step (in gray) are ***not*** required. No "bad style" penalties.

Solution (b): Let w be an arbitrary string.
Assume $\mathsf{sortpairs}(\mathsf{sortpairs}(x)) = \mathsf{sortpairs}(x)$ for every string x shortest than w.
There are two cases to consider:

- If $|w| \leq 1$, we immediately have $\mathsf{sortpairs}(\underline{\mathsf{sortpairs}(w)}) = \mathsf{sortpairs}(\underline{w})$ by definition of sortpairs.
- If $w = \mathtt{10}x$ for some string x then

$$\begin{aligned}
\mathsf{sortpairs}(\mathsf{sortpairs}(w)) &= \mathsf{sortpairs}(\mathsf{sortpairs}(\mathtt{10}x)) && \text{because } w = \mathtt{10}x \\
&= \mathsf{sortpairs}(\mathtt{01} \cdot \mathsf{sortpairs}(x)) && \text{def. sortpairs} \\
&= \mathtt{01} \cdot \mathsf{sortpairs}(\mathsf{sortpairs}(x)) && \text{def. sortpairs} \\
&= \mathtt{01} \cdot \mathsf{sortpairs}(x) && \textit{induction hypothesis} \\
&= \mathsf{sortpairs}(\mathtt{10}x) && \text{def. sortpairs} \\
&= \mathsf{sortpairs}(w) && \text{because } w = \mathtt{10}x
\end{aligned}$$

- Otherwise, $w = abx$ for some string x and symbols a, b such that $ab \neq \mathtt{10}$. Then

$$\begin{aligned}
\mathsf{sortpairs}(\mathsf{sortpairs}(w)) &= \mathsf{sortpairs}(\mathsf{sortpairs}(abx)) && \text{because } w = abx \\
&= \mathsf{sortpairs}(ab \cdot \mathsf{sortpairs}(x)) && \text{def. sortpairs and } ab \neq \mathtt{10} \\
&= ab \cdot \mathsf{sortpairs}(\mathsf{sortpairs}(x)) && \text{def. sortpairs and } ab \neq \mathtt{10} \\
&= ab \cdot \mathsf{sortpairs}(x) && \textit{induction hypothesis} \\
&= \mathsf{sortpairs}(abx) && \text{def. sortpairs and } ab \neq \mathtt{10} \\
&= \mathsf{sortpairs}(w) && \text{because } w = abx
\end{aligned}$$

■

Rubric: 5 points: standard induction rubric. Explanations for each step (in gray) are ***not*** required. No "bad style" penalties.

CS/ECE 374 A ✧ Fall 2023 Midterm 1 Solutions Problem 4	Name:

For any string $w \in \{0,1\}^*$, let compact(w) denote the string obtained by replacing each run with a single symbol from that run. Let L be an arbitrary regular language.

(a) ***Prove*** that the language $\textsc{Compact}(L) = \{\text{compact}(w) \mid w \in L\}$ is regular.

(b) ***Prove*** that the language $\textsc{Uncompact}(L) = \{w \in \Sigma^* \mid \text{compact}(w) \in L\}$ is regular.

Solution (a): Let $M = (Q, s, A, \delta)$ be any DFA that accepts L. We construct an NFA $M' = (Q', S', A', \delta')$ ***with multiple start states*** that accepts $\textsc{Compact}(L)$ as follows:

$$Q' = Q \times \{0\text{next}, 0\text{run}, 1\text{next}, 1\text{run}\}$$
$$S' = \{s\} \times \{0\text{next}, 1\text{next}\}$$
$$A' = A \times \{0\text{next}, 1\text{next}\}$$

$$\delta'((q, 0\text{next}), 0) = \{(\delta(q,0), 0\text{run})\}$$
$$\delta'((q, 0\text{run}), \varepsilon) = \{(\delta(q,0), 0\text{run}),\ (q, 1\text{next})\}$$

$$\delta'((q, 1\text{next}), 1) = \{(\delta(q,1), 1\text{run})\}$$
$$\delta'((q, 1\text{run}), \varepsilon) = \{(\delta(q,1), 1\text{run}),\ (q, 0\text{next})\}$$

All unspecified transitions go to $\emptyset$.

Intuitively, mode 0next means the next input bit, if any, must be 1; when M' reads that 1, it sends 1 to M. Mode 0run means M' is guessing the rest of the current run of 0s in the original string w; in this mode, M' sends 0s to M without consuming input symbols. Modes 1next and 1run are symmetric. ■

Rubric: 5 points: standard language transformation rubric. Explanation (in gray) is ***not*** required. This is not the only correct solution. -2½ for confusing (a) and (b).

Solution (b): Let $M = (Q, s, A, \delta)$ be any DFA that accepts L. We construct an NFA $M' = (Q', s', A', \delta')$ that accepts UNCOMPACT(L) as follows:

$$\begin{aligned} Q' &= Q \times \{\mathsf{prev0}, \mathsf{prev1}, \mathsf{start}\} \\ s' &= (s, \mathsf{start}) \\ A' &= A \times \{\mathsf{prev0}, \mathsf{prev1}, \mathsf{start}\} \end{aligned}$$

$$\begin{aligned} \delta'((q, \mathsf{prev0}), \mathtt{0}) &= \{(q, \mathsf{prev0})\} \\ \delta'((q, \mathsf{prev0}), \mathtt{1}) &= \{(\delta(q, \mathtt{1}), \mathsf{prev1})\} \\ \\ \delta'((q, \mathsf{prev1}), \mathtt{0}) &= \{(\delta(q, \mathtt{0}), \mathsf{prev0})\} \\ \delta'((q, \mathsf{prev1}), \mathtt{1}) &= \{(q, \mathsf{prev1})\} \\ \\ \delta'((q, \mathsf{start}), \mathtt{0}) &= \{(\delta(q, \mathtt{0}), \mathsf{prev0})\} \\ \delta'((q, \mathsf{start}), \mathtt{1}) &= \{(\delta(q, \mathtt{1}), \mathsf{prev1})\} \end{aligned}$$

M' passes only the first symbol in every run to M. Modes prev0 and prev1 remember M's previous input symbol; mode start means M' hasn't read anything yet. This NFA doesn't actually use any nondeterminism; we could make it a DFA just by stripping braces from the output of δ'. ■

Solution (b): Let $M = (Q, s, A, \delta)$ be any DFA that accepts L. We construct a **DFA** $M' = (Q', s', A', \delta')$ that accepts UNCOMPACT(L) as follows:

$$\begin{aligned} Q' &= Q \times \{\varepsilon, \mathtt{0}, \mathtt{1}\} \\ s' &= (s, \varepsilon) \\ A' &= \begin{cases} \left\{(q, a) \;\middle|\; a \in \{\mathtt{0}, \mathtt{1}\} \text{ and } \delta(q, a) \in A\right\} \cup \{(s, \varepsilon)\} & \text{if } s \in A \\ \left\{(q, a) \;\middle|\; a \in \{\mathtt{0}, \mathtt{1}\} \text{ and } \delta(q, a) \in A\right\} & \text{otherwise} \end{cases} \end{aligned}$$

$$\begin{aligned} \delta'((q, \varepsilon), \mathtt{0}) &= (q, \mathtt{0}) \\ \delta'((q, \varepsilon), \mathtt{1}) &= (q, \mathtt{1}) \\ \\ \delta'((q, \mathtt{0}), \mathtt{0}) &= (q, \mathtt{0}) \\ \delta'((q, \mathtt{0}), \mathtt{1}) &= (\delta(q, \mathtt{0}), \mathtt{1}) \\ \\ \delta'((q, \mathtt{1}), \mathtt{0}) &= (\delta(q, \mathtt{1}), \mathtt{0}) \\ \delta'((q, \mathtt{1}), \mathtt{1}) &= (q, \mathtt{1}) \end{aligned}$$

M' passes only the ***last*** symbol in every run to M, but only after reading the first symbol of the next run or after the input string ends. State $(q, a) \in Q'$ means M is in state q and M' just read symbol a. State (s, ε) means M' hasn't read anything yet; all other states (q, ε) are unreachable. ■

Rubric: 5 points: standard language transformation rubric. Explanation (in gray) is ***not*** required. These are not the only correct solutions. -2½ for confusing (a) and (b).

CS/ECE 374 A ✧ Fall 2023 Midterm 1 Solutions Problem 5	Name:

For each of the following languages L over the alphabet $\Sigma = \{0, 1\}$, describe a DFA that accepts L **and** give a regular expression that represents L. You do not need to justify your answers.

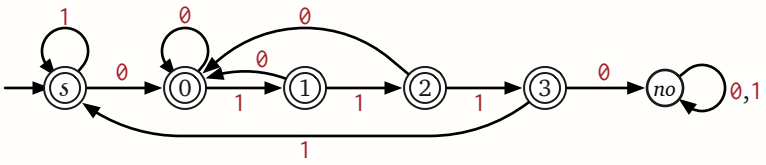
(a) Strings that do not contain the substring 01110.

(b) Strings that contain odd-length run of 0s, followed immediately by an even-length run of 1s.

Solution (a): Equivalently, strings in which no run of 1s has length 3, except possibly at the start and end of the string.

$$1^*(00^*\underbrace{(1 + 11 + 11111^*)}0^*1^*$$

The bracketed subexpression represents a run of 1s whose length is not 3.



The states have the following meanings:

- s: We are reading an initial run of 1s or we have just read more than three 1s in a run.
- 0: We are reading a run of 0s
- 1: We have just read the first 1 in a run
- 2: We have just read the first two 1s in a run
- 3: We have just read the first three 1s in a run
- *no*: We have read the forbidden substring

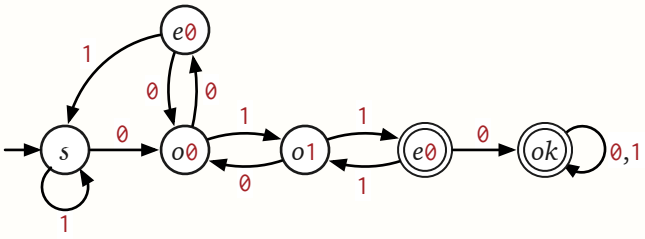
■

Rubric: 5 points = 2½ for regular expression (standard rubric) + 2½ for DFA (standard rubric). Explanations (in gray) and DFA state names are **not** required. These are not the only correct solutions.

Solution (b):

$$(\varepsilon + (0+1)^*1)\ \underbrace{0(00)^*}\ \underbrace{11(11)^*}\ (0(0+1)^* + \varepsilon)$$

The two bracketed subexpressions represent an odd-length run of 0s and an even-length run of 1s.



The states have the following meanings:

- s: We didn't just read anything interesting.
- $o0$: We have just read an odd run of 0s
- $e0$: We have just read an even run of 0s
- $o1$: We have just read an odd run of 0s followed by an odd run of 1s.
- $e1$: We have just read an odd run of 0s followed by an even run of 1s.
- ok: The input string contains the target pattern

■

Rubric: 5 points = 2½ for regular expression (standard rubric) + 2½ for DFA (standard rubric). Explanations and DFA state names (in gray) are **not** required. These are not the only correct solutions.