CSE396, Spring 2026 Problem Set 8 Due Mon. 4/13, 11:59pm

This homework is due on **Monday**, April 13, not on Friday. Prelim II has definitely been set for **Tuesday, April 21**, in class period. Once again, the last part of this homework arises from the current Tuesday lecture. Note: it has **four** numbered problems but only 83 points, fewer than last week.

Reading

For Thursday, please continue reading Chapter 4 to the end. For next week, please read section 5.1, skim/skip section 5.2, and read section 5.3. Then go back and re-read section 5.1 with the idea of employing the formal version of the reducibility concept. It strikes me as odd that the titular concept of the chapter is formalized only at the end—the text puts the intuition first, but my teaching style is more to regard intuition as guided by formal structure.

The dividing line between what is covered for Prelim II is basically: decidable languages *yes* (cumulatively regular languages through CFLs and the non-CFLs we've built efficient Turing machines for), methods for showing undecidability *no*. Thus up through section 4.1 in the text, but not 4.2 or chapter 5. Not uncountability, not diagonalization. The one exception is that in Thursday's lecture I may tell which problems involving grammars are undecidable as facts to know, without yet giving their proofs.

——**Homework**—part online and all *individual work*—due **Mon. 4/13, 11:59pm**———-

(1) Using *TopHat*, the "Worksheet" titled **Spr 2026 HW8 Online Part**. There are 10 questions, each worth 2 points, for 20 total. (Recall the TopHat deadline is strict.)

(2) Say that a context-free grammar $G = (V, \Sigma, \mathcal{R}, S)$ is in "short form" if for each $A \in V$, the rules for the variable A *collectively* have no more than two occurrences of variables *total* on *all their right-hand sides*. This form allows $A \to \epsilon$ freely and allows unit rules $A \to B$ (but at most two of them for each A) unlike Chomsky normal form. But if you have a rule $A \to BC$, or even $A \to BB$ or $A \to AA$, then no other rules for A may involve variables. For argument's sake we'll fix $\Sigma = \{0, 1\}$ for this problem.

The goal is to show that for every regular language L there is a CFG G in short form such that $L = L(G)$. Prove this by structural induction on the following grammar G_{REG} that generates regular expressions R:

$$E \to \emptyset \mid \epsilon \mid 0 \mid 1 \mid (E \cup E) \mid (E \cdot E) \mid (E^*)$$

Put another way, describe and verify an algorithm to convert any regular expression R into a short-form CFG G such that $L(G) = L(R)$, so that the algorithm works by recursion on sub-expressions. That way your answer should follow the same structure as the conversion from regexps to equivalent NFAs shown in class. (24 pts.)

(3) Define the **reversal** L^R of a language L by $L^R = \{x^R : x \in L\}$. Note that if every string in L is a palindrome, then $L^R = L$. More generally, if L includes x^R whenever it

includes a string x, then $L = L^R$. And vice-versa. Give a decision procedure for the following computational problem:

INSTANCE: A DFA M.
QUESTION: Does $L(M)$ equal its reversal ?

Hint: Give a pseudocode algorithm whose first "macro-step" is to convert the DFA M into an NFA N such that $L(N) = L(M)^R$. This is done by making s the only accepting state, giving N a new start state s' with ϵ-arcs to all the old final states of M, and reversing the direction of each arc of M. You do not need to repeat fine details of this conversion or prove it correct, but you may use it as a line of your procedure the way the text does in section 4.1. Ultimately your algorithm should produce a DFA M' such that deciding the E_{DFA} problem on M' tells you the answer about M. (This will flow together with how the text proves Theorem 4.5 but you should show all the steps anyway. 12 pts.)

(4) Consider the following context-free grammar G:

$$\begin{aligned} S &\to AC \mid DC \\ A &\to aS \mid BA \\ B &\to \epsilon \mid SCS \\ C &\to BD \mid AS \\ D &\to BB \mid b \end{aligned}$$

(a) Find the whole set N of nullable variables—please show how that set grows iteratively according to the algorithm in class starting from the initial $\{B\}$. Then carry out the step that adds rules skipping any subset of *occurrences* of nullable variables to get a new grammar G_1. Note that if S is nullable then you get $L(G_1) = L(G) \setminus \{\epsilon\}$, else you get $L(G_1) = L(G)$. (Do **not** do the text's initial step of adding a new start variable S_0. 9 + 6 = 15 pts.)

(b) Your grammar G_1 will have several unit rules—but don't include "self-rules" like $A \to A$. Draw a directed graph whose nodes are the five variables and which has an edge (A, B) if $A \to B$ is a unit rule. Then take the transitive closure of the graph, which will tell you all pairs (A, B) such that $A \Longrightarrow^* B$. (Here we again *ignore* self-loops; that is, we only consider $B \neq A$. 6 + 6 = 12 pts., for 27 total on this problem and 83 total on the set)

Please **do not** do any further steps toward Chomsky normal form—do not copy right-hand-sides of unit rules; do not alias terminals to variables; do not shorten long rules. [Once as an April Fool's joke, I did the equivalent of listing these steps as (c),(d),(e) worth −9 points each, so that the problem totaled 0 points—but the Autolab link still showed "27 points possible."] The nullable variables and unit-rules steps are the most meaningful ones, and will be reviewed again on Thursday 4/9.