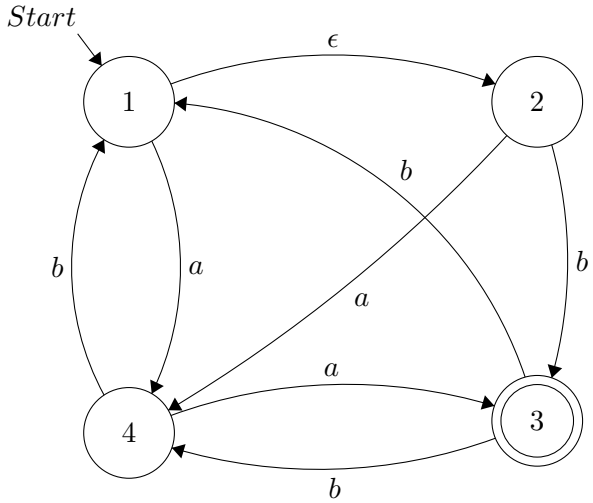
CSE396, Spring 2026 **Problem Set 3** **Due Fri. 2/20, 11:59pm**

Reading: Next Tuesday's lecture start the subject of nonregular languages in section 1.4, but covering the Myhill-Nerode Theorem (MNT) instead of the Pumping Lemma. I will use my notes https://cse.buffalo.edu/~regan/cse396/CSE396MNT.pdf on the course webpage. The text covers MNT in the chapter 1 exercises but with more mathematical sophistication than is required to apply it to prove that certain languages are not regular. So read section 1.4 but follow my notes using MNT on the text's "Pumping Lemma" examples. (We will later cover a stronger pumping lemma that works for non-context-free languages.)

Homework—part online and all *individual work*—due **Fri. 2/20, 11:59pm**:

(1) Using *TopHat*, the "Worksheet" titled **S26 HW3 Online Part**. There are 11 not 10 questions. Questions 1–5 are multiple choice and worth 2 points each as usual, but Q6 is a "matching" problem worth 5 points, and then 7–11 are true/false questions worth 1 point each. The total is still 20.

(2) Convert the followjng NFA into an equivalent DFA. The components are $\Sigma = \{a, b\}$, $Q = \{1, 2, 3, 4\}$, $s = 1$, $F = \{3\}$, and $\delta = \{(1, \epsilon, 2), (1, a, 4), (2, a, 4), (2, b, 3), (3, b, 1), (3, b, 4), (4, a, 3), (4, b, 1)\}$.



You must show the steps of the set-of-states method. Also answer the following questions:

(a) Find two strings x, x' of the shortest possible lengths such that for each of its four states q, N can process x from s to q, and likewise x'.

(b) Find the shortest string that N cannot process from s at all.

(c) Is there a string y such that regardless of what state q N starts in, N cannot process y?

You may find your DFA most helpful to answer those questions with. (Points are 18 for the DFA and 6+3+3 for the questions, making 30.)

(3) Use the FA-to-regexp algorithm to convert N (not your DFA) from problem (2) into an equivalent regular expression r. Eliminating state 2 should be easy, but eliminating 4 will take more work. It is fine to use either the "graphical style" or the "code style" (with T-matrix) of the basic algorithm, both of which were exemplified at the end of the Thu. 2/12 lecture. Or you can combine them. Getting your final r from the two-state formula given in lecture will save work over the text's way. **You must show your algorithm steps**, not just give the final r. (18 pts., for 68 on the set)