CSCI 432/532, Spring 2025

Homework 3

Due Tuesday, February 10, 2025 at 11:59pm Mountain Time

Submission Requirements

- Type or clearly hand-write your solutions into a PDF format so that they are legible and professional. Submit your PDF to the appropriate Canvas dropbox.
- Do not submit your first draft. Type or clearly re-write your solutions for your final submission.
- You may work with a group of up to three students and submit ***one single document*** for the group. Just be sure to list all group members at the top of the document.
- When possible, the homework will include at least one fully solved problem, similar to that week's assigned problems, together with the rubric we would use to grade this problem if it appeared in an actual homework or exam. These model solutions show our recommendations for structure, presentation, and level of detail in your homework solutions. (Obviously, the actual *content* of your solutions won't match the model solutions, because your problems are different!)

Academic Integrity

Remember, you may access ***any*** resource in preparing your solution to the homework. However, you must

- write your solutions in your own words, and
- credit every resource you use (for example: "Bob Smith helped me on problem 2. He took this course at UM in Fall 2020"; "I found a solution to a problem similar to this one in the lecture notes for a different course, found at this link: www.profzeno.com/agreatclass/lecture10"; "I asked ChatGPT how to solve problem 1 part (c); "I put my solution for problem 1 part (c) into ChatGPT to check that it was correct and it caught a missing case and suggested some grammer fixes.") If you use the provided LaTeX template, you can use the `sources` environment for this. Ask if you need help!

Grading Rubrics

For the fooling set problems:

Standard fooling set rubric. 10 points =

+ 4 for the fooling set.
  - +2 for explicitly describing the proposed fooling set F.
  - +2 if F is truly a fooling set for the lanugage.
  - −4 if F is not a fooling set for the target language.
  - No credit for the *problem* if F is finite.

+ 6 for the proof.
  - The proof must correctly consider arbitrary pairs of distinct strings $x, y \in F$.
  - No credit for the proof unless both x and y are always in F.
  - No credit for the proof unless x and y can be any pair of distinct strings in F.
  - +2 for explicitly describing a suffix z that distinguishes x and y.
  - +2 for proving either $xz \in L$ or $yz \in L$.
  - +2 for proving either $xz \notin L$ or $yz \notin L$, respectively.

For the NFAs:

NFA/DFA rubric. 10 points =

+ 2 for an unambiguous description of a DFA or NFA, including the states set Q, the start state s, the accepting states A, and the transition function δ.
  - Drawings:
    * Use an arrow from nowhere to indicate the start state s.
    * Use doubled circles to indicate accepting states A.
    * If $A = \emptyset$, say so explicitly.
    * If your drawing omits a junk/trash/reject/hell state, say so explicitly.
    * Draw neatly! If we can't read your solution, we can't give you credit for it.
  - Text descriptions: You can describe the transition function either using a 2d array, using mathematical notation, or using an algorithm.
    * You must explicitly specify $\delta(q, a)$ for every state q and every symbol a.
    * If you are describing an NFA with ε-transitions, you must explicitly specify $\delta(q, \varepsilon)$ for every state q.
    * If you are describing a DFA, then every value $\delta(q, a)$ must be a single state.
    * If you are describing an NFA, then every value $\delta(q, a)$ must be a set of states.
    * In addition, if you are describing an NFA with ε-transitions, then every value $\delta(q, \varepsilon)$ must be a set of states.
  - Product constructions: You must give a complete description of each of the DFAs you are combining (as either drawings, text, or recursive products), together with the accepting states of the product DFA. In particular, we will not assume that product constructions compute intersections by default.

+ 4 for briefly explaining the purpose of each state in English. This is how you argue that your DFA or NFA is correct.
  - In particular, each state must have a mnemonic name.
  - For product constructions, explaining the states in the factor DFAs is both necessary and sufficient.
  - Yes, we mean it. A perfectly correct drawing of a perfectly correct DFA with no state explanation is worth at most 6 points.

+ 4 for correctness.

- −1 for a single mistake: a single misdirected transition, a single missing or extra accepting state, rejecting exactly one string that should be accepted, or accepting exactly one string that should be accepted. (The incorrectly accepted/rejected string is almost always the empty string ε.)
- −4 for incorrectly accepting every string, or incorrectly rejecting every string.
- −2 for incorrectly accepting/rejecting more than one but a finite number of strings.
- −4 for incorrectly accepting/rejecting an infinite number of strings.

For the language transformation problems:

Language transformation rubric. 10 points =

+ 2 for a formal, complete, and unambiguous description of the output automaton M', including the states, the start state(s), the accepting states, and the transition function, as functions of an *arbitrary* given DFA M. The description must state whether the output automaton is a DFA or an NFA.
  - No points for the rest of the problem if this is missing.

+ 2 for a *brief* English explanation of the output automaton. We explicitly do *not* want a formal proof of correctness, or an English *transcription*, but a few sentences explaining how your machine works and justifying its correctness. What is the overall idea? What do the states represent? What is the transition function doing? Why these accepting states?
  - No points for the rests of the problem if this is missing.

+ 6 for correctness
  + 1 for correct states–Almost always a product of the states Q and some additional information. Does the additional information make sense?
  + 1 for correct start state(s)
  + 1 for correct accepting states
  + 3 for correct transition function
    - −1 for a single minor mistake

1. Prove that the following languages over the alphabet $\Sigma = \{0, 1\}$ are not regular by proving that there is an infinite fooling set for each of them.

   (a) $\{0^a 1 0^b 1 0^c : 2b = a + c\}$.

   (b) The set of all palindromes in Σ^* whose lengths are divisible by 7.

2. For each of the following languages over the alphabet $\Sigma = \{0, 1\}$, describe an NFA that accepts the language. (Note that (a) is slightly changed from Homework 2 and (b) is exactly the same as Homework 2.) Remember that a complete drawing is sufficient to describe an NFA, but you should explain what the states mean.

   (a) All strings that are either of the form 10^*1 or whose length is a multiple of 3 (or both).

   (b) All strings whose ninth-to-last symbol is 0, or equivalently, the set

$$\{x0z : x, z \in \Sigma^* \text{ and } |z| = 8\}.$$

3. Consider the following string function:

$$double0(w) := \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ 00 \bullet double0(x) & \text{if } w = 0x \\ 1 \cdot double0(x) & \text{if } w = 1x \end{cases}$$

   For example, $double0(1001) = 100001$.

   Let L be an arbitrary regular language over the alphabet $\Sigma = \{0, 1\}$. Prove that $\text{DOUBLE0}(L) = \{double0(w) : w \in L\}$ is also regular.

Solved problems (use the proof template for your homework!)

1. Prove that the language of all palindromes, $\{s : w = rev(w)\}$, is not regular by proving that there is an infinite fooling set for it.

Solution: Let $F = \{0^n 1 : n \geq 0\}$.
Let x and y be different arbitrary elements of F.
Then $x = 0^i 1$ and $y = 0^j 1$ for $i \neq j$.
Consider $z = 0^i$. Then

- $xz = 0^i 1 0^i$ is a palindrome is therefore in L.
- $xz = 0^j 1 0^i$ is not a palindrome since $i \neq j$ and is therefore not in L.

Thus, z is a distinguishing suffix for x and y and F is a fooling set for L.
Since F is infinite, L is not regular. ■