CS/ECE 374 ✧ Spring 2021

Homework 2

Due ~~Wednesday~~ Thursday, February ~~17~~ 18, 2021 at 10am

Starting with this homework, groups of up to three people can submit joint solutions. Each problem should be submitted by exactly one person, and the beginning of the homework should clearly state the Gradescope names and email addresses of each group member. In addition, whoever submits the homework must tell Gradescope who their other group members are.

The following unnumbered problems are not for submission or grading. No solutions for them will be provided but you can discuss them on Piazza.

- Suppose $M = (Q, \Sigma, \delta, s, A)$ is a DFA. For states $p, q \in Q$ (p can be same as q) argue that $L_{p,q} = \{w \mid \delta^*(p, w) = q\}$ is regular. Recall that $\text{PREFIX}(L) = \{w \mid wx \in L, x \in \Sigma^*\}$ is the set of all prefixes of strings in L. Express $\text{PREFIX}(L(M))$ as $\cup_{q \in Z} L_{s,q}$ for a suitable set of states $Z \subseteq Q$. Why does this prove that $\text{PREFIX}(L(M))$ is regular whenever L is regular?

- For a language L let $\text{MID}(L) = \{w \mid xwy \in L, x, y \in \Sigma^*\}$. Prove that $\text{MID}(L)$ is regular if L is regular.

1. (a) Draw an NFA that accepts the language $\{w \mid$ there is exactly one block of 1s of even length$\}$. (A "block of 1s" is a maximal substring of 1s.)

   (b) i. Draw an NFA for the regular expression $(010)^* + (01)^* + 0^*$.

   ii. Now using the powerset construction (also called the subset construcion), design a DFA for the same language. Label the states of your DFA with names that are sets of states of your NFA. You should use the incremental construction so that you only generate the states that are reachable form the start state.

2. For a language L let $\text{SUFFIX}(L) = \{y \mid \exists x \in \Sigma^*, xy \in L\}$ be the set of suffixes of strings in L. Let $\text{PSUFFIX}(L) = \{y \mid \exists x \in \Sigma^*, |x| \geq 1, xy \in L\}$ be the set of proper suffixes of strings in L.

   (a) Prove that if L is regular then $\text{PSUFFIX}(L)$ is regular via the following technique. Let $M = (Q, \Sigma, \delta, s, A)$ be a DFA accepting L. Describe a NFA N in terms of M that accepts $\text{PSUFFIX}(L)$. Explain the construction of your NFA.

   (b) Prove that if L is regular then $\text{PSUFFIX}(L)$ is regular via the following alternate technique. Let r be a regular expression. We will develop an algorithm that given r constructs a regular expression r' such that $L(r') = \text{PSUFFIX}(L(r))$. Assume $\Sigma = \{0, 1\}$. No correctness proof is required but a brief explanation of the derivation would help you get partial credit in case of mistakes.

   i. For each of the base cases of regular expressions $\emptyset, \epsilon$ and $\{a\}, a \in \Sigma$ describe a regular expression for $\text{PSUFFIX}(L(r))$.

ii. Suppose r_1 and r_2 are regular expressions, and r_1' and r_2' are regular expressions for the languages PSUFFIX($L(r_1)$) and PSUFFIX($L(r_2)$) respectively. Describe a regular expression for the language PSUFFIX($L(r_1 + r_2)$) using r_1, r_2, r_1', r_2'.

iii. Same as the previous part but now consider $L(r_1 r_2)$.

iv. Same as the previous part but now consider $L((r_1)^*)$.

v. Apply your construction to the regular expression $r = 0^* + (01)^* + 011^*0$ to obtain a regular expression for the language PSUFFIX($L(r)$).

3. Recall that if M is a DFA that accepts a language L then it is easy to construct a DFA M' to accept the language $\overline{L}$ (the complement of L) by simply altering the final states. The product construction allows one to take two DFAs M_1 and M_2 and construct a machine M that accepts $L(M_1) \cap L(M_2)$. Here we explore NFAs. If N_1 and N_2 are two NFAs then we saw in lecture that it is easy to construct a NFA N that accepts $L(N_1) \cup L(N_2)$.

- Let $N_1 = (Q_1, \Sigma, \delta_1, s_1, A_1)$ and $N_2 = (Q_2, \Sigma, \delta_2, s_2, A_2)$ be two NFAs. Show that the product construction can be generalized to create an NFA $N = (Q_1 \times Q_2, \Sigma, \delta, s, A)$ such that $L(N_1) = L(N_1) \cap L(N_2)$. Your main task is to formally define δ, s, A in terms of the parameters of N_1 and N_2. Briefly justify why your construction works. A formal proof by induction is not needed.
- Let $N = (Q, \Sigma, \delta, s, A)$ be an NFA, and define the NFA $N_{\text{comp}} = (Q, \Sigma, \delta, s, Q \setminus A)$. In other words we simply complemented the accepting states of N to obtain N_{comp}. Describe a concrete example of a machine N to show that $L(N_{\text{comp}}) \neq \overline{L(N)}$. You need to explain for your machine N what $\overline{L(N)}$ and $L(N_{\text{comp}})$ are.
- Define an NFA or DFA that accepts $\overline{L(N)} - L(N_{\text{comp}})$, and explain how it works.
- **Not to submit:** Define an NFA that accepts $L(N_{\text{comp}}) - \overline{L(N)}$, and explain how it works.

Hint: For the last three parts it is useful to classify strings in Σ^* based on the behaviour of N.

Solved problem

4. Let L be an arbitrary regular language. Prove that the language $half(L) := \{w \mid ww \in L\}$ is also regular.

Solution: Let $M = (\Sigma, Q, s, A, \delta)$ be an arbitrary DFA that accepts L. We define a new NFA $M' = (\Sigma, Q', s', A', \delta')$ with ε-transitions that accepts $half(L)$, as follows:

$$Q' = (Q \times Q \times Q) \cup \{s'\}$$
$$s' \text{ is an explicit state in } Q'$$
$$A' = \{(h, h, q) \mid h \in Q \text{ and } q \in A\}$$
$$\delta'(s', \varepsilon) = \{(s, h, h) \mid h \in Q\}$$
$$\delta'((p, h, q), a) = \big\{\big(\delta(p, a), h, \delta(q, a)\big)\big\}$$

M' reads its input string w and simulates M reading the input string ww. Specifically, M' simultaneously simulates two copies of M, one reading the left half of ww starting at the usual start state s, and the other reading the right half of ww starting at some intermediate state h.

- The new start state s' non-deterministically guesses the "halfway" state $h = \delta^*(s, w)$ without reading any input; this is the only non-determinism in M'.
- State (p, h, q) means the following:
  - The left copy of M (which started at state s) is now in state p.
  - The initial guess for the halfway state is h.
  - The right copy of M (which started at state h) is now in state q.
- M' accepts if and only if the left copy of M ends at state h (so the initial non-deterministic guess $h = \delta^*(s, w)$ was correct) and the right copy of M ends in an accepting state.

■

Rubric: 5 points =

+ 1 for a formal, complete, and unambiguous description of a DFA or NFA
  - No points for the rest of the problem if this is missing.
+ 3 for a correct NFA
  - −1 for a single mistake in the description (for example a typo)
+ 1 for a *brief* English justification. We explicitly do *not* want a formal proof of correctness, but we do want one or two sentences explaining how the NFA works.