CS/ECE 374 ✧ Spring 2021

Homework 3

Due Thursday, February 25, 2021 at 10am

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. Some of them (and others) are available on PrarieLearn as extra practice problems which gives you a chance to check your answers.

- Let L be an arbitrary regular language.
  - Prove that the language $palin(L)\{w \mid ww^R \in L\}$ is also regular.
  - Prove that the language $drome(L)\{w \mid w^R w \in L\}$ is also regular.
- Suppose F is a fooling set for a language L. Argue that F cannot contain two distinct string x, y where both are not prefixes of strings in L.
- Prove that the language $\{0^i 1^j \mid \gcd(i, j) = 1\}$ is not regular.
- Consider the language $L = \{w : |w| = 1 \mod 5\}$. We have already seen that this language is regular. Prove that any DFA that accepts this language needs at least 5 states.

1. (a) Prove that the following languages are not regular by providing a fooling set. You need to provide an infinite set and also prove that it is a valid fooling set for the given language.
   i. $L = \{\mathtt{0}^i \mathtt{1}^j \mathtt{2}^k \mid i + j = k + 1\}$.
   ii. Recall that a block in a string is a maximal non-empty substring of indentical symbols. Let L be the set of all strings in $\{\mathtt{0}, \mathtt{1}\}^*$ that contain two non-empty blocks of `1`s of unequal length. For example, L contains the strings `01101111` and `01001011100010` but does not contain the strings `000110011011` and `00000000111`.
   iii. $L = \{0^{n^3} \mid n \geq 0\}$.

   (b) Suppose L is not regular. Prove that $L \cup L'$ is not regular for any finite language L'. Give a simple example of a non-regular language L and a regular language L' such that $L \cup L'$ is regular.

2. Given languages L_1 and L_2 we define $delete(L_1, L_2)$ to be the language $\{uw \mid uvw \in L_2, v \in L_1\}$ to be the set of strings obtained by "deleting" a string of L_1 from a string of L_2. For example if $L_1 = \{\texttt{not}\}$ and $L_2 = \{\texttt{CS374isnotfun}, \texttt{not}, \texttt{notnotnot}, \texttt{blah}\}$ then

$$delete(L_1, L_2) = \{\texttt{CS374isfun}, \varepsilon, \texttt{notnot}\}.$$

Prove that if L_1, L_2 are regular then $delete(L_1, L_2)$ is also regular. In particular you should describe how to construct an NFA $N = (Q, \Sigma, \delta, s, A)$ from two DFAs $M_1 = (Q_1, \Sigma, \delta_1, s_1, A_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, s_2, A_2)$ such that $L(N) = delete(L(M_1), L(M_2))$. You do not need to prove the correctness of your construction but you should explain the ideas behind the construction (see lab 3.5 solutions).

3. Let $L_k = \{w \in \{0,1\}^* : |w| \geq 2k$ and last $2k$ characters of w has equal number of $\texttt{0}$s and $\texttt{1}$s$\}$. If $k = 3$ then $\texttt{010011}$ and $\texttt{000111000}$ are in L_3 while $\texttt{0001111}$ and $\texttt{01000110}$ are not.

   - Describe an NFA for L_k that has $O(k^2)$ states.
   - Prove that the set of all binary strings of length k is a fooling set for L_k. Conclude that any DFA for L_k needs at least 2^k states.
   - **Extra credit:** Improve size of the fooling set for L_k or prove that there is a DFA for L_k with 2^k states.

4. **Not to submit:** Consider all regular expressions over an alphabet Σ. Each regular expression is a string over a larger alphabet $\Sigma' = \Sigma \cup \{\emptyset\text{-Symbol}, \epsilon\text{-Symbol}, +, (,), *\}$. We use $\emptyset$-Symbol and ϵ-Symbol in place of $\emptyset$ and ϵ to avoid confusion with overloading; technically one should do it with $+, (,)$ as well. Let R_Σ be the language of regular expressions over Σ.

   (a) Prove that R_Σ is not regular.

   (b) Describe a context free grammar (CFG) for R_Σ which will prove that it is a CFL.

   This shows that we need more expressive languages than regular languages to describe regular expressions.