GROUP HOMEWORK 6, CPSC 421/501, FALL 2023

JOEL FRIEDMAN

Please note:

(1) You must justify all answers; no credit is given for a correct answer without justification.
(2) Proofs should be written out formally.
(3) You do not have to use LaTeX for homework, but **homework that is too difficult to read will not be graded.**
(4) You may work together on homework in groups of up to four, **but you must submit a single homework as a group submission under Gradescope.**

---

(1) Show that if $L_1, L_2 \subset \Sigma^*$ are regular languages over an alphabet, Σ, then the following languages are regular:
   (a) $\Sigma^* \setminus L_1$;
   (b) $(L_1 \cap L_2)$ is regular (you may use the fact that $L_1 \cup L_2$ is regular, and part (a));
   (c) $L_1 \setminus L_2 = \{w \in L_1 \mid w \notin L_2\}$.

(2) (a) Show that

$$L_1 = \{a^n b^m \mid n, m \text{ are non-negative integers and } n \text{ is a perfect cube}\}$$

is non-regular, by assuming that L_1 is regular, and obtaining a contradiction as follows: choose an appropriate regular language L_2 and show that $L_1 \cap L_2$ is a language that we already know (from class and/or the handout "Non-regular languages...") is non-regular.

   (b) Similarly show that

$$L_1 = \{a^n b^m \mid n, m \text{ are non-negative integers and } n + m \text{ is a perfect cube}\}$$

is non-regular. (In parts (a) and (b), it doesn't matter whether or not we consider 0 to be a perfect cube.)

Research supported in part by an NSERC grant.

(c) **Extra Credit:** Show that

$L_1 = \{a^n b^m \mid n, m \text{ are positive integers and } n \text{ is a perfect cube}\}$

is non-regular, without using Sections 1.3, 1.4 or the Myhill-Nerode Theorem.

(3) [Sip], Problem 1.31, but do this as follows: explain how to take a DFA, M, recognizing a language, L, and produce from M an NFA, M', recognizing L^{rev}; in particular, use only Sections 1.1 and 1.2, and do not use Section 1.3 on regular expressions. Explain your general method for doing this, and **give one example that illustrates all of the main ideas of your method**.

(4) Let $\Sigma = \{a, b\}$.
   (a) Write down a 3-state NFA that recognizes $\{a, ab\}$.
   (b) Using the NFA in part (a), write down a 3-state NFA that recognizes $\{a, ab\}^*$.
   (c) Using the NFA in part (b), write down a DFA that recognizes $\{a, ab\}^*$, using the general procedure for converting an NFA to a DFA; however, of the 8 possible states, you can ignore any states that are irrelevant to the DFA (i.e., that are never reached on any input).

Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, CANADA.
E-mail address: `jf@cs.ubc.ca`
URL: `http://www.cs.ubc.ca/~jf`