CS600 February 7, 2018

Homework 1

Students are welcome to work together, but *every student must write up their own solutions, independently!* I strongly encourage students to use LaTex for writing up their solutions. Please see the course web-page for a template file. When problems require you to draw a state machine, feel free to include a hand-drawn picture with your typed up solutions.

One question will be graded, but students should submit solutions to all problems.

Question 1: Suppose that language L_1 is decided by some automata $M_1 = (\Sigma, Q_1, S_1, \mathcal{A}_1, \delta_1)$, and that language L_2 is decided by automata $M_2 = (\Sigma, Q_2, S_2, \mathcal{A}_2, \delta_2)$. Construct an NFA $M = (\Sigma, Q, S, \mathcal{A}, \delta)$ that decides L_1L_2. (Give both a graphical representation of M, as well as a formal description of M.)

Question 2: Suppose that language L is decided by some automata $M' = (\Sigma, Q', S', \mathcal{A}', \delta')$. Construct an NFA $M = (\Sigma, Q, S, \mathcal{A}, \delta)$ that decides L^*. (Give both a graphical representation of M, as well as a formal description of M.)

Things to understand: both formally and informally, why are the Regular Languages equivalent to the languages decided by DFAs?

Question 3: Construct a DFA for the language L over alphabet $\{a, b\}$, where
$L = \{\{a, b\}^* \mid \text{ the 3rd to last character is an } a\}$. (A graphical representation of the DFA suffices.)

Question 4: Construct a NPDA (push-down automata) for the language
$L = \{\{a, b\}^* \mid \text{ there are more } a\text{s than } b\text{s}\}$

Question 5: Use the pumping lemma to prove that the language from the previous question is not regular.

Question 6: As we have shown, there is an NPDA for the language $L = \{a^n b^n \mid n \geq 0\} \cup \{a^n b^{2n} \mid n \geq 0\}$. We also showed that if there were a DPDA for this language, then we can build a push-down automata for the language $L' = \{a^n b^n c^n \mid n \geq 0\}$. Because this violates a pumping lemma for PDAs (which we didn't prove), the contradiction allows us to conclude that there is no DPDA for the language L. Recall that in this proof, we construct a machine for L' by using 2 copies of the DPDAs for L. Where does the proof breakdown if we instead use 2 NPDAs for L?