Assignment 2
Due no later than June 10 at 23:00

For full credit it is enough to accumulate 10 points.

Please give succinct, unambiguous, and well-phrased answers to the problems on the assignment. These qualities will be taken into account in the assessment. Each problem needs to be written on a single page and the whole assignment needs to be submitted on Gradescope.

A bonus of **2 points** will be given if the assignment is typeset in LaTeX. You may draw the automatons outside of latex (paint, take picture of handwritten), and insert picture files, but the rest must be typeset in LaTeX to receive full bonus.

Exercise 1 (3 points) Consider the language L_1 of all strings over the alphabet

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, =\}$$

that constitute valid equations of the form $x + y = z$, where x, y, and z are non-negative integers represented in base 10, without leading zeros. Some elements of L_1 include $13 + 17 = 30$ and $99 + 1 = 100$, but not $13 + 17 = 29$ or $99 + 01 = 100$. Prove that L_1 is not regular using proof by contradiction and the pumping lemma.

Exercise 2 (3 points). Design a context-free grammar that generates the language L_2 over $\Sigma = \{a, b\}$ consisting of strings w with the length a multiple of 4 such that the first quarter of the characters in w contains at least one b. For example $baaa \in L_1$ and $abbaaabbbaaa \in L_1$, but $\epsilon \notin L_1$ and $aabaabaa \notin L_1$. Show a derivation for $abaaaaaa$. You need to briefly explain how the grammar works and what is the role of each nonterminal. No points to solutions which do not include explanations.

Exercise 3 (4 points) PDA Design

1. **(1 point)** Give a pushdown automaton that that accepts exactly the strings with correctly matched parentheses in the alphabet containing just two symbols, left and right parentheses $\Sigma = \{(,)\}$. Explain your construction.

2. **(1 point)** Give a pushdown automaton that accepts exactly the strings over $\Sigma = \{a, b\}$ that contain the same number of as and bs. For example, *aabbaabbab* is accepted but *aba* is not. Also *bbbbaaaa* and *aaaabbbb* are both accepted. Explain your construction.

3. **(2 points)** Give a pushdown automaton that accepts strings in the alphabet $\Sigma = \{a, b,), (\}$ such that the parentheses are correctly matched, and between each left parenthesis and the next right parenthesis, and in the whole string, there are equal numbers of a's and bs.

   For example, $((abab)(ab))$ is accepted but $((aab(ab))$ is not accepted. Also $((abab)ab(ab))$ is accepted, but not $((abab)aba(ab))$. The stated conditions guarantee that, as you proceed from left to right, each time you encounter either kind of parenthesis, the numbers of as and bs already processed are equal. Explain how $((abab)ab(ab))$ is accepted by your PDA, and $((abab)aba(ab))$ is not.