Computation Theory (COMP 170), Fall 2020
Assignment 06

Answer each problem below to the best of your ability. Submit all parts by 9:00 AM on Monday, November 2. List your collaborators. Late homework is accepted within 24 hours for half credit. After 24 hours no credit is given. The first late assignment (up to 24 hours) per student incurs no penalty. **Make sure that your submission follows the formatting guidelines given at the end of this document.**

Reading: Sipser Chapter 4

[1] (*6 pts.*) **CFG → PDA**

In class we showed that the language

$$L = \{xy \mid x, y \in \Sigma^*, |x| = |y|, x \neq y\}$$

where $\Sigma = \{a, b\}$ is context free by constructing the following grammar G:

$$\begin{aligned} S &\to AB \mid BA \\ A &\to XAX \mid a \\ B &\to XBX \mid b \\ X &\to a \mid b \end{aligned}$$

Convert G into an equivalent PDA, using the method discussed in class, or the equivalent procedure of Lemma 2.21 (Sipser, 3rd ed., pp. 117-120). List the transition function fully or draw the PDA pictorially.

Note: In lecture, we allowed pushing multiple symbols on the stack at a time, whereas Sipser restricts pushing to one symbol at a time. Your solution can be either. The former will be briefer.

[2] (*8 pts.*) **Combining Machine Types**

(This is problem 2.20 from Sipser).

Let $A/B = \{w \mid wx \in A \text{ for some } x \in B\}$. Show that if A is context free and B is regular, then A/B is context free.

For consistency, assume you're given a PDA

$$M_A = (Q_A, \Sigma, \Gamma, \delta_A, s_A, F_A)$$

that recognizes A and DFA

$$M_B = (Q_B, \Sigma, \delta_B, s_B, F_B)$$

that recognizes B. Give an explicit construction of a new PDA M that recognizes A/B. Be sure to be explicit about the following:

- The set of states used by M
- The transition rule(s) that your machine includes if $\delta_A(q_A, a, A)$ contains (p_A, B) for $a \in \Sigma_\varepsilon$, $A, B \in \Gamma_\varepsilon$, $q_A, p_A \in Q_A$.

[3] (*6 pts.*) **Primal**

Let $L = \{a^p \mid p \text{ is prime}\}$. Show that L is Turing decidable. Give a description at the level used in class. So you can say things like "return to the beginning of the tape" or "zig-zag between the beginning and the end, marking the outer-most unmarked characters". Saying "check whether p is divisible by k" is too high-level for a formal description, although would be appreciated by the graders. A complete description of δ is too low-level.

Make sure you're explicit about the set of symbols you'll need. Also, prefer simplicity over efficiency. E.g., if you're writing a Java method to test the primality of n, you probably don't want to check whether it's divisible by 4, or look for factors greater than $\sqrt{n}$. For designing a TM, don't bother with those sorts of optimizations.