Assignment 3
Due: Thursday, July 4

On this assignment you will be asked to describe PDAs and DSMs that perform certain tasks. Please work hard to make your PDAs and DSMs as clear and simple as possible, be sure to correctly use the notation described in the notes, and include helpful commentary if you believe it will assist the grader in understanding your answer—for you may lose points if it is too difficult for the grader to verify the correctness of your solutions, even if they turn out to be correct!

1. [8 points] In this question, you are asked to provide descriptions of PDAs for two languages over the binary alphabet $\Sigma = \{0, 1\}$. The shorthand notation for PDAs introduced in Lecture 11 may help to make your PDAs state diagrams more concise.

(a) Describe a PDA that recognizes the language $A = \{w \in \Sigma^* : |w|_0 > |w|_1\}$.

Here, the notation $|w|_a$ denotes the number of times the symbol $a$ appears in $w$.

(b) Describe a PDA that recognizes the language $B = \{uv : u, v \in \Sigma^*, |u| = |v|, u \neq v\}$.

2. [8 points] For a given alphabet $\Sigma$, recall from Lecture 11 that we define

$$\text{Stack}(\Sigma) = \{\uparrow, \downarrow\} \times \Sigma,$$

and we say that a string $w \in \text{Stack}(\Sigma)^*$ is a valid stack string if, when it is read from left to right, it represents a legal sequence of pushes and pops for a single stack (where $(\downarrow, a)$ means “push the symbol $a$ onto the stack” and $(\uparrow, a)$ means “pop the symbol $a$ off of the stack”). Note that a valid stack string does not need to result in an empty stack—the only thing that can make a stack string invalid is that it attempts to either pop the wrong symbol or pop an empty stack.

Prove that the language $A = \{w \in \text{Stack}(\Sigma)^* : w \text{ is not a valid stack string}\}$ is context-free.

3. [8 points] Give the description of a DSM $M$ having input alphabet $\Sigma = \{0, 1\}$ that operates as follows:

- If the input to $M$ is $\varepsilon$, then $M$ should reject.
- If the input to $M$ is $x \neq \varepsilon$, then $M$ should accept. Moreover, when $M$ accepts, its input stack (i.e., stack 0, or $X$ if you prefer) should store the string $y$ that comes immediately before $x$ with respect to the lexicographic ordering of $\Sigma^*$.

5 bonus points go to the person or persons having a correct solution with the smallest number of states (including the accept state and reject state). You may use subroutines if you wish, but they contribute the total number of states required to implement them.

4. [1 point] For each of the questions above, list the full name of each of your 360 classmates with whom you worked on that question. (If you didn’t work with anyone, that is fine: just indicate that you worked alone.)