

Due 10/21

Construct a PDA that accepts  $\{ x\#y \mid x, y \text{ in } \{0, 1\}^* \text{ such that } x \neq y \text{ and } x_i = y_i \text{ for some } i, 1 \leq i \leq \min(|x|, |y|) \}$ .

For your PDA to work correctly it will need to be non-deterministic. You can assume that you will always be given a valid string – that is, the input will always contain one # and x and y will be strings over  $\{0, 1\}$ . My more simplistic PDA has 31 states and is broken into two major sections, one for  $|x| = |y|$  and one for  $|x| \neq |y|$ . My more complex PDA has 19 states, and is also broken into the same two major sections.

For the case where we assume that  $|x| = |y|$ , you need to find a symbol that matches at the same index of x and y ( $x_i = y_i$  for some i) and a symbol that does not match at the same index of x and y ( $x_j \neq y_j$  for some j). One way that this can be accomplished is by finding an index i such that  $x_i = y_i$  and  $x_{i+1} \neq y_{i+1}$  or  $x_{i+1} = y_{i+1}$  and  $x_i \neq y_i$ . As in programming assignment 3, you can store the index in the stack and the values of  $x_i$  and  $x_{i+1}$  in the state.

For the case where we assume that  $|x| \neq |y|$ , you need to find an index i where  $x_i = y_i$ . Since the lengths are different, we get that  $x \neq y$  without finding an index j in which  $x_j \neq y_j$ . For this case, you can simply check that  $x_1 = y_1$  and verify that  $|x| \neq |y|$ . If  $x_1 \neq y_1$ , then the other portion of the code (where we assume that  $|x| = |y|$ ) will accept the string.

Your PDA will need to be able to handle input strings of length up to about 80 symbols using at most 1GB of memory. My PDA can handle all the test strings with the default 64MB of memory that is allocated to the heap.

You should be able to use the concepts learned for programming assignment 3 to assist you in constructing this PDA.

If you are having memory issues, try starting JFLAP from the command line allocating more memory. As an example, from the directory that JFLAP is in, enter "java -Xmx500m -jar JFLAP7.1.jar". The "-Xmx500m" tells java to allocate 500MB of memory to the heap. By default, java only allocates 64MB to the heap.

Post your program on Brightspace for the assignment associated with this programming assignment by 11:59:59pm on the date due. Your submission is to be a JFLAP file (jff extension) with the filename being your last name in lower case followed by "\_p4.jff" (my filename would be "garrison\_p4.jff").