University of Nevada, Las Vegas Computer Science 456/656 Fall 2019

Assignment 5: Due October 21, 2019

Name:_____

You are permitted to work in groups, get help from others, read books, and use the internet. But the handwriting on this document must be your own. Print out the document, staple, and fill in the answers. You may attach extra sheets. Turn in the pages to the graduate assistant at the beginning of class, September 4. Except for Problem 1, each problem is in both fifth and sixth editions.

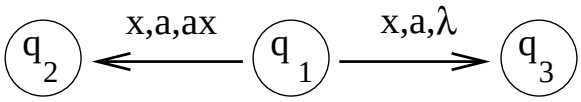
1. Let L be the language defined in Example 6.11 on page 179 of the sixth edition, or on page 176 of the fifth edition. Work either Exercise 1 on page 180 of the sixth edition, or Exercise 1 on page 177 of the fifth edition. (Warning: these two exercises are not identical.)

The notation I used in my discussion of push-down automata in class on October 14, 2019, differs from the notation introduced in our textbook, Linz, in some important ways. As you work the homework problems, please use the notation given in the textbook. My notation was derived from the notation used for LALR parsers in the "Dragon" book, a standard text for compiler theory.

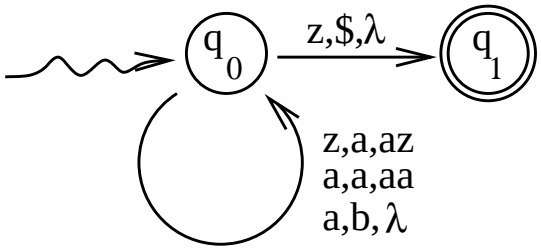
In the notation given in Linz, an **i.d.** is an ordered triple (state,stack,stream), where the stack is written top-to-bottom and the input stream is written with the next input symbol on the left.

The left side of each transition rule is of the form δ(state, input symbol or λ, stack symbol), and the right side is a let of ordered pairs of the form (state, string of stack symbols).

For example, suppose that $\delta(q_1, a, x) = \{(q_2, ax).(q_3, \lambda)\}$, and the current **i.d.** is (q_1, aab, xxz), then the next **i.d.** is either $(q_2, ab, axxz)$ or (q_3, ab, xz). The following figure shows part of the transition diagram.



This is the state diagram for a DPDA which accepts the Dyck language. For a DPDA to work, we must assume every string ends with an **eof** symbol, which we denote as \$.



2. This just a yes/no question! Consider the ambiguous context-free grammar G for algebraic expressions, where the start symbol is E

   $E \to E + E$
   $E \to E - E$
   $E \to E * E$
   $E \to -E$
   $E \to (E)$
   $E \to x$
   $E \to y$
   $E \to z$

   Does there exist a DPDA which accepts $L(G)$?
   Hint: This grammar can be parsed using an LALR parser.

3. Work problem 6(b) on page 189 of the sixth edition, which problem 4(b) on page 185 of the fifth edition. Give a state diagram as well as the transition function. Can this language be accepted by a DPDA?

4. We write $L_1 \subseteq_{\mathcal{P}} L_2$ to mean that there s a $\mathcal{P}$–TIME reduction of L_1 to L_2. For example:
$$\text{SAT} \subseteq_{\mathcal{P}} \text{3-SAT} \subseteq_{\mathcal{P}} \text{Independent Set} \subseteq_{\mathcal{P}} \text{Subset Sum} \subseteq_{\mathcal{P}} \text{Partition}$$

   Prove that Subset Sum $\subseteq_{\mathcal{P}}$ Partition.

I have deleted the next problem because we have not covered conjunction normal form.