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

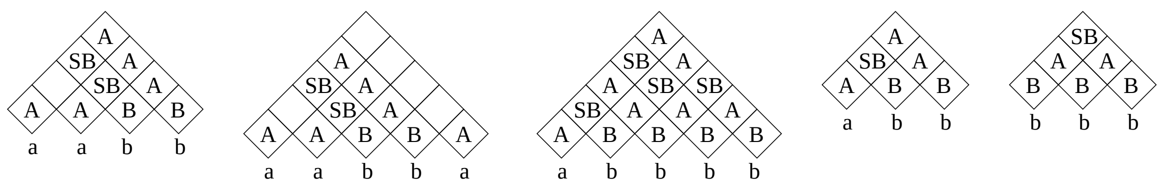
Answers to Assignment 5: Due October 21, 2019

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 grammar is:
   $S \to AB$
   $A \to BB|a$
   $B \to AB|b$

   Use the CYK algorithm to determine if the given strings are in L. The fifth edition gives the strings $aabb$, $aabba$, and $abbbb$. The sixth edition gives the strings abb, bbb, $aabba$, and $abbbb$.



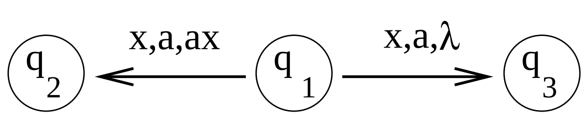
   Thus, $bbb \in L$, but none of the others are in L.

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.

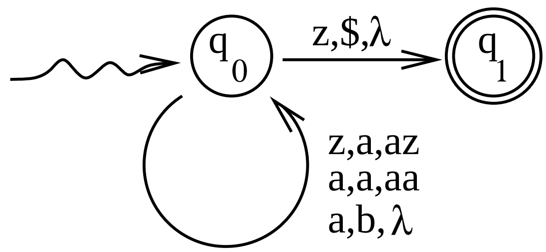
In the notation given in Linz, an **i.d.** is an ordered triple (state,stack,stream), where the stack is written top-to-bottom and the input stream is written with the next input symbol on the left.

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 input string ends with an **eof** symbol, which we denote as $.



And this is the transtion table.

state	read	pop	state	push
q_0	λ	z	q_1	λ
q_0	a	z	q_0	az
q_0	a	a	q_0	aa
q_0	b	a	q_0	λ

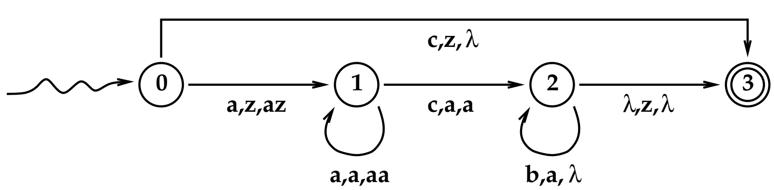
2. This just a yes/no question! Consider the ambiguous context-free grammar G for algebraic expressions, where the start symbol is E

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

   Does there exist a DPDA which accepts $L(G)$? Hint: You can use an LALR parser for this grammar.

   Yes. The LALR parser is deterministic.

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?



state	read	pop	state	push
q_0	c	z	q_3	λ
q_0	a	z	q_1	az
q_1	a	a	q_1	aa
q_1	c	a	q_2	a
q_2	b	a	q_2	λ
q_2	λ	z	q_3	λ

   Yes, there is a DPDA that accepts $\{a^n cb^n\}$.

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:

   SAT $\subseteq_{\mathcal{P}}$ 3-SAT $\subseteq_{\mathcal{P}}$ Independent Set $\subseteq_{\mathcal{P}}$ Subset Sum $\subseteq_{\mathcal{P}}$ Partition

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

   An instance of the subset sum problem consists of a number K and a set $\mathcal{X}$ of items $I_1, \ldots I_n$ where the weight of $I_i = x_i$. The question is whether there is a subset $S \subseteq \mathcal{X}$ of weight K.

   We reduce that instance of the subset sum problem to an instance of the partition problem. Let $W = \sum_{i=1}^{n} x_i$. Let $\mathcal{X}^*$ consist of $\mathcal{X}$ together with two additional items, I_{n+1} and I_{n+2} of weights $x_{n+1} = W - K + 1$ and $x_{n+2} = K + 1$, respectively. The weight of $\mathcal{X}^*$ is $\sum_{i=1}^{n+2} x_i = 2W + 2$, and the question is whether $\mathcal{X}^*$ can be partitioned into two subsets each of weight $W + 1$.

   If there is a solution S to the subset sum instance, let $S^* = S \cup \{I_{n+1}\}$, a subset of $\mathcal{X}^*$ of weight $K + (W - K + 1) = W + 1$, and we are done.

   Conversely, suppose there is a solution to the partition instance. The items I_{n+1} and I_{n+2} cannot be in the same subset, since the total weight of that subset would be at least $K+1+W-K+1 = W+2 > W+1$. Let S^* be the subset of weight $W + 1$ which contains I_{n+1} and let S be the subset of $\mathcal{X}$ obtained by removing I_{n+1} from S^*. The weight of S is $W + 1 - (W - K + 1) = K$. Thus, S is a solution to the subset sum instance.