CS 321 Quiz 6 (20 min., 5 pts) November 13, 2015

Last Name = _____, First Name = _____

ONID login = = _____@oregonstate.edu

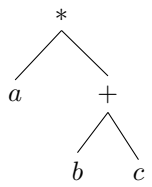
1. (3 pts) For the grammar $E \to E + E \mid E * E \mid a \mid b \mid c$

   (a, 1 pt) Show it is ambiguous (draw two parse trees for one string).

   (b, 1 pt) Write an unambiguous grammar for this language such that, as shown in the textbook and slides, + has a lower precedence than $*$, e.g., $a + b * c$ is always interpreted as $a + (b * c)$.

   (c, 0.5 pts) To allow + to sometimes take a higher priority than $*$ *without* introducing parentheses, the Polish logician Jan Łuaksiewicz invented the famous *reverse Polish notation* (aka "postfix"), where operators come after the operands, e.g., $(a + b) * c$ is written as $a\ b + c\ *$, and $a * (b + c)$ is written as $a\ b\ c + *$. This is simply the "post-order traversal" of the expression tree whereas the conventional infix notation is the "in-order traversal" (see below). Write a grammar for the reverse Polish notation (the terminals are still $a, b, c, +, *$).

   Example:

   infix: $a * (b + c)$; postfix: $a\ b\ c + *$

   (d, 0.5 pts) Is the above grammar for reverse Polish notation ambiguous? Justify.

2. (1 pt) Prove the following language is not regular: $L = \{w \in \{0, 1\}^* \mid w \text{ has equal numbers of 0's and 1's}\}$.

3. (1 pt) Here is a CFG for the above language $S \to SS \mid 0S1 \mid 1S0 \mid \epsilon$. Is this grammar ambiguous?