CS154: Homework #4

Due: Thursday, August 9, 2012 by 5PM

Problem 1. This is a CFG for regular expressions over the alphabet $\{a, b\}$. Rewrite the grammar into an equivalent grammar (i.e., one with the same language) that is unambiguous. The resulting grammar should group operations according to their precedence ($*$ should be *stickier* than $\cdot$, which should be *stickier* than $+$). Draw the parse tree for $(a + b \cdot a)^*$ using your new grammar.

$$\begin{array}{rcl} R & \rightarrow & \emptyset \\ R & \rightarrow & e \\ R & \rightarrow & a \\ R & \rightarrow & b \\ R & \rightarrow & R + R \\ R & \rightarrow & R \cdot R \\ R & \rightarrow & R^* \\ R & \rightarrow & (R) \end{array}$$

E -> E + T | T
T -> T*F | F
F -> F* | I
I -> K | (E)
E-> a | b | e | phi

Note: If it is not obvious how to do this, example 2.4 in Sipser does this for simple arithmetic expressions.

Problem 2. Consider the following language over the alphabet $\Sigma = \{0, 1\}$.

$$\text{L} = \{0^i 1^j \mid i \leq j \leq 2i \text{ and } i \geq 0\}$$

This is the set of strings where all the 0's come before all the 1's, and the number of 1's is at least the number of 0's but no more than *twice* the number of 0's.

a) Provide a context-free grammar for L.

b) Design a PDA that accepts L by final state.

Problem 3. Use the CFL pumping lemma to show that the following language is not context-free.

$$L = \{0^p \mid p \text{ is a prime.}\}$$

Problem 4. Let G be the following grammar:

b -> B, a -> A,C
ba -> S, A aa ->

$$\begin{array}{rcl} S & \rightarrow & AB|BC \\ A & \rightarrow & BA|a \\ B & \rightarrow & CC|b \\ C & \rightarrow & AB|a \end{array}$$

Use the CYK algorithm to determine whether (1) *baaab*, and (2) *aabab* is in $L(G)$. Show the contents of the table filled in by the CYK algorithm.

Problem 5. The shuffle of languages L_1 and L_2 is the language of strings that are *arbitrary interleavings* of symbols from some string in L_1 and some string in L_2. Using closure properties of context-free languages and a known non-CFL, prove that CFL's are not closed under shuffle.

consider A = #(a) = #(b)
consider B = #(c) = #(d)
then the interleaving of A and B will produce all strings with #(a) = #(b) and #(c) = #(d)
clearly not context-free