CSE396, Spring 2021 Problem Set 5 Due Tue. 3/23, 11:59pm

Reading:

For Tuesday's lecture, please read the handout "Structural Induction and CFGs" on the course webpage at https://cse.buffalo.edu/˜regan/cse396/CSE396SI.pdf This supplements the text's explanations of how grammars work. The final piece of section 2.1, on Chomsky normal form (ChNF), will come on Thursday. I regard this as feeding into section 2.3 (jumping over 2.2), because we are free to suppose the postulated grammar is in ChNF, and that makes the parse trees used in the proof especially nice—they are binary trees. So please also read ahead to section 2.3, plus my handout https://cse.buffalo.edu/˜regan/cse396/CSE396ChNF.pdf on ChNF. I may choose to skim the whole proof of conversion to ChNF, but I regard the first step of determining which variables can derive ϵ (first page of this handout) as an important example of algorithms that come in the first half of chapter 4. (And to complete the reasons for going out of order here, I prefer to save learning a whole separate ϵ-filled notation for pushdown automata by presenting them as a case of two-tape Turing machines in chapter 3.) Bottom line: take extra reading time to finish 2.1, read 2.3, and the handouts.

Homework—this written part only (**No TopHat part**)—and all your *individual work*:

(1) For the following three languages L_1, L_2, L_3 over $\{0,1\}$, design context-free grammars G_1, G_2, G_3 such that $L(G_1) = L_1$, $L(G_2) = L_2$, and $L(G_3) = L_3$. You need not prove your grammars correct, but as usual you should include a few comments explaining how and why the grammars work correctly. ($3 \times 9 = 27$ pts.)

1. $L_1 = (0 \cup 10)^*(1 \cup \epsilon)$
2. $L_2 = \{0^m 1^n 0^n 1^m : m \geq 1, n \geq 0\}$,
3. $L_3 = \{x0y : \#0(x) = \#1(y)\}$.

(2) This is a variation on part of the Spring 2019 HW5 problem (3), and you are welcome to refer to it and its key. Again, we are discussing a grammar G' to represent regular expressions over the alphabet $\Sigma = \{a, b\}$. This time we will change up a little to say the terminal alphabet for the grammar is $\Sigma' = \{a, b, \epsilon, \emptyset, +, \cdot, *, (,)\}$, where $\emptyset$ is the regular-expression symbol for $\emptyset$ and ϵ is a symbol for for ϵ (but neither of these symbols is used in the questions or your answers so never mind). The grammar G' is

$$S \rightarrow a \mid b \mid \emptyset \mid \epsilon \mid S + S \mid S \cdot S \mid S^* \mid (S).$$

The periods here and below are just punctuation.

(a) Give both a parse tree and a leftmost derivation for the string $r = b + (b \cdot b + a)^*$.

(b) Show that r is ambiguous in G by giving a different parse tree and corresponding leftmost derivation for it. (There are multiple answers.)

(c) Briefly explain why one parse should be preferred over the other. (Or, if you think you found two "bad" parses, say why. 15 pts. total, for 42 on the set.)