CS 517: Problem Set 2

Due: Thursday Apr 25; **typed** and **submitted electronically**.

1. Recall that a **context-free grammar (CFG)** consists of a set of string-manipulation rules. A rule of the form "$A \to x$" means: "replace any occurrence of character A with string x." In a CFG, the left-hand side of every rule (the "A" in this example) must be a single character.

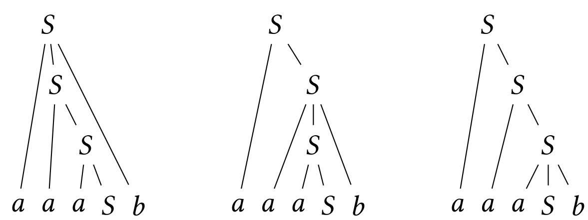
   The language of a CFG consists of all strings that can be generated by applying these rules (in any order), starting from a special *starting symbol*, usually called S. For example, the following grammar produces all strings of the form $a^n S b^m$ where $n \geq m$:

   $$S \to aSb, \qquad S \to aS$$

   (For the purposes of this question, we don't need to differentiate between terminal and non-terminal characters.)

   If a string can be generated by the CFG, we can define a **parse tree** for that string in the natural way. All nodes are labeled with a character. The tree's root is labeled with the starting symbol. If a node is labeled A, and its children, when read left-to-right, read the string x, then $A \to x$ should be a valid CFG rule. The leaves of the tree should read the target string, left-to-right.

   For example, the string $aaaSb$ has several possible parse trees in the CFG given above:

   Prove that the following problem is undecidable: Given a CFG (i.e., a finite set of string-manipulation rules), decide whether there exists a string x that has more than one parse tree in this CFG.

   Hint: Use a reduction involving the PCP problem. Two parse trees should somehow correspond to the top and bottom strings of a domino arrangement.

2. Show that the following language is undecidable:

   $$\{M \mid M \text{ is a TM that has } \textit{polynomial} \text{ worst-case running time}\}$$

   Rice's thereom doesn't apply here! Be careful, because the worst-case running time of a TM is a *global* property of the TM, that involves its behavior on *all* inputs.

   Also, this problem is **not** asking about $\{M \mid L(M) \in \mathsf{P}\}$. That language is undecidable as an immediate corrolary of Rice's theorem. Remember that a TM M might decide a problem in P but do it with an exponential-time algorithm. The language in this problem is asking about the running time of M, not about an inherent property of $L(M)$.

3. Let L be an arbitrary NP language characterized by witness-checking algorithm R. So, $L = \{x \mid \exists w : R(x, w) = 1\}$. If $R(x, w) = 1$ then we say that w is a valid witness for x.

   Show that if $\mathsf{P} = \mathsf{NP}$ then there is a polynomial time algorithm M where: on input $x \in L$, $M(x)$ outputs a valid witness for x, and on input $x \notin L$, $M(x)$ outputs the string "no witness". In other words, $\mathsf{P} = \mathsf{NP}$ has ramifications, not just for answering the *decision* problem (yes/no), but also for actually finding witnesses.

   Hint: The problem assumes $\mathsf{P} = \mathsf{NP}$, so not only is $L \in \mathsf{P}$, but any other NP-language you can think of is also in P. Think of an NP language that is *related* to L, that will help you efficiently find witnesses.

4. (a) Show that $\mathsf{NP} = \{L \mid L \leq_p \mathsf{SAT}\}$. This gives another equivalent definition for NP. (You need to prove both the $\subseteq$ and $\supseteq$ directions)

   (b) Prove that a language L is NP-complete if and only if $\overline{L}$ is coNP-complete.