CS:5340 Homework 4

Due: Tue, 10/3

Notes: (a) Any problem numbers mentioned in the handout refer to problems in the textbook, by Arora and Barak. (b) It is possible that solutions to some of these problems are available to you via other theory of computation books or on-line lecture notes, etc. If you use any such sources, please acknowledge these in your homework *and* present your solutions in your own words. You will benefit most from the homework, if you seriously attempt each problem on your own first, before seeking other sources. (c) As mentioned in the syllabus, it is okay to form groups of two in solving and submitting homework solutions. But, my advice from (b) still applies: you will benefit most from the homework, if you seriously attempt each problem on your own first, before seeking help from your group partner. (d) Discussing these problems with any of your classmates is okay, provided you and your classmates are not being too specific about solutions. In any case, make sure that you take no written material away from these discussions *and* (as in (b)) you present your solutions in your own words. When discussing homework with classmates please be aware of guidelines on "Academic Dishonesty" as mentioned in the course syllabus.

1. I defined the complexity class coNP-hard as follows in class.

$$\text{coNP-hard} = \{L' \subseteq \{0,1\}^* \mid \forall L \in \text{coNP} : L \leq_P L'\}.$$

A student in class provided an alternate definition:

$$\text{coNP-hard} = \{L \subseteq \{0,1\}^* \mid \overline{L} \in \text{NP-hard}\}.$$

Prove or disprove the following claim: these two definitions are equivalent.

2. This question refers to the proof of the Cook-Levin Theorem (Lemma 2.11). Suppose that $L \in NP$. Then there is a (deterministic) TM M and polynomials $p(\cdot)$ and $q(\cdot)$ such that M runs in time $q(n)$ and for all $x \in \{0,1\}^*$, $x \in L$ iff there exists $u \in \{0,1\}^{p(|x|)}$ such that $M(x,u) = 1$. Suppose that for this particular L, $p(n) = n^2$ and $q(n) = n^3$. Furthermore, suppose that the alphabet for M (denoted Γ) is $\{0, 1, \rhd, \Box\}$ and that M has 5 states (i.e., $|Q| = 5$). As in the proof in the textbook, assume that M uses two tapes, an input tape and a work/output tape, and that M is oblivious.

   (a) Let c denote the fewest number of bits needed to represent a snapshot of M. Calculate the value of c.

   (b) For a string $x \in \{0,1\}^n$, calculate the number of variables that the boolean formula φ_x has (as a function of n).

   (c) Let $x \in \{0,1\}^n$ and let φ_x be the boolean formula constructed from x. Let C be the smallest number such that every clause in φ_x has at most C literals. Calculate C.

3. Let $L \subseteq \{1\}^*$ be a language that contains strings of 1's. Prove the following claim: `INDSET` $\leq_P$ L implies that `INDSET` $\in P$.

   Here is how you might want to think about this proof. Let $G = (V, E)$ be an n-vertex graph and k be a positive integer. For any vertex $v \in V$, let $N(v)$ denote the *closed neighborhood* of v, i.e., the set containing all vertices adjacent to v *and* v itself. Let A be the following exponential time algorithm that solves `INDSET` on (G, k).

Pick an arbitrary vertex v and return 1 if either the recursive call to $A((G \backslash N(v), k-1)$ returns 1 or the recursive call to $A(G \setminus \{v\}, k)$ returns 1; otherwise return 0.

The first recursive call represents the choice to place v in the independent set being (implicitly) constructed. The second recursive call represents the choice to exclude v from the independent set. By the way, I have not specified the base cases of this recursive algorithm. The computation being performed by this recursive algorithm can be viewed as a binary tree of exponential size. The idea that drives the proof is that the reduction $\texttt{INDSET} \leq_P L$ allows us to prune this tree substantially and reduce its size to polynomial in the size of the input (G, k). Specifically, let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function such that $x \in \texttt{INDSET}$ iff $f(x) \in L$. Then, assuming that $k \leq n$, $|f((G,k))| \leq c \cdot n^d$ for some constants c, d. In fact, for any subgraph G' of G and any $k' \leq k$, $|f((G', k'))| \leq c \cdot n^d$. (You will have to prove this as part of your proof.) You can now use this fact along with the property that L is "sparse," i.e., it only contains at most one string of each length, to show that the exponential-time search by algorithm A can be pruned to give us a polynomial-time algorithm for `INDSET`.

4. Problem 2.33 (Page 67).