# CS:5340 Homework 3

**Due: Thu, 9/21**

---

**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. Let $L \subseteq \{0,1\}^*$ be a language. Let $L^+$ denote the language consisting of all strings $x$ such that $x$ is obtained by concatenating a finite number strings belonging to $L$. (In other words, for $n \geq 1$, let $L^n = \{x_1 \cdot x_2 \cdot \ldots \cdot x_n \mid x_1, x_2, \ldots, x_i \in L\}$, where "$\cdot$" denotes the concatenation operator. Then define $L^+ = \cup_{n \geq 1} L^n$.) Prove that if $L \in NP$ then $L^+ \in NP$.

2. Recall the function $R$ from Problem 2, Homework 2. Let $L_R = \{\alpha \in \{0,1\}^* \mid R(M_\alpha) = 1\}$. (a) Prove that $L_R$ is NP-hard. (b) Is $L_R \in NP$? (Yes/No). Justify your answer in a sentence.

3. Prove that if $P = NP$ there is a polynomial-time algorithm that takes an undirected graph $G = (V, E)$ as input and returns a largest independent set in $G$ as output. Theorem 2.18 provides an indirect way of solving this problem, using a reduction to `SAT`. Here I want you to solve this problem directly, without reliance on `SAT`, by using the fact that if $P = NP$, then (the decision problem) `INDSET` has a polynomial time algorithm.

---