CS 535: Complexity Theory, Fall 2020

Homework 2

Due: 8:00PM, Friday, September 18, 2020.

Reminder. Homework must be typeset with LaTeX preferred. Make sure you understand the course collaboration and honesty policy before beginning this assignment. Collaboration is permitted, but you must write the solutions *by yourself without assistance*. You must also identify your collaborators. Assignments missing a collaboration statement will not be accepted. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Problem 1 (coNP). Recall that the complexity class **coNP** consists of languages L such that $\overline{L} \in \mathbf{NP}$.

(a) Show that a language L is **NP**-complete if and only if $\overline{L}$ is **coNP**-complete. Recall that completeness for both classes is defined with respect to polynomial-time (Karp) reductions.

(b) In the discrete art gallery problem, there are n paintings numbered $1, \ldots, n$ and m guard posts. A guard stationed at guard post i is able to see some set $S_i \subseteq [n]$ of paintings. An art gallery is *k-vulnerable* if for every assignment of k guards to guard posts, there exists a painting that none of those guards can see. That is, define

$$\mathsf{VUL} = \{\langle S_1, \ldots, S_m, n, k\rangle \mid \forall T \subseteq [m], |T| = k \quad \exists j \in [n], j \notin \cup_{i \in T} S_i\}.$$

Prove that VUL is **coNP**-complete.

Hint: You may use, without proof, the fact that $\mathsf{VERTEXCOVER}$ is **NP**-complete.

(c) Find the first error in the following "proof" that $\mathbf{NP} = \mathbf{coNP}$, and explain why it is an error: Let M be a nondeterministic polynomial-time algorithm computing SAT. We design a nondeterministic polynomial-time algorithm computing

$$\mathsf{UNSAT} = \{\varphi \text{ a CNF formula } \mid \forall x \; \varphi(x) = 0\}$$

as follows. On input φ (an instance of UNSAT), evaluate $b = M(\varphi)$. If $b = 0$, output 1, and if $b = 1$, output 0. This runs in nondeterministic polynomial-time as long as M does, and $\varphi \in \mathsf{SAT}$ iff $\varphi \notin \mathsf{UNSAT}$, so it decides UNSAT. Therefore, $\mathsf{UNSAT} \in \mathbf{NP}$. Since UNSAT is **coNP**-complete, it follows that $\mathbf{coNP} \subseteq \mathbf{NP}$. A similar argument shows that $\mathbf{NP} \subseteq \mathbf{coNP}$, hence $\mathbf{NP} = \mathbf{coNP}$.

Problem 2 (Decision vs. Optimization). An **NP**-optimization problem is specified by a polynomial-time computable objective function $f : \{0,1\}^* \times \{0,1\}^* \to \mathbb{N}$ and a polynomial p. Given an input $x \in \{0,1\}^*$, let $Y_x = \{y \in \{0,1\}^* \mid |y| \leq p(|x|)\}$. The problem is to find a $y \in Y_x$ that maximizes $f(x,y)$, i.e., find a string in $\operatorname*{argmax}_{y \in Y_x} f(x,y)$.

(a) Formulate the problem of finding a largest independent set in a graph as an **NP**-optimization problem.

(b) Show that **P** = **NP** if and only if every **NP**-optimization problem can be solved in polynomial time.

Hint: It may help to think about how you would use a polynomial-time algorithm for INDSET to solve the problem from part (a).