CPSC 421: Introduction to Theory of Computing
Practice Problem Set #5, Not to be handed in

Note: $\mathbb{N} = \{0, 1, \dots\} \subset \{1, 2, \dots\} = \mathbb{N}_+$. For simplicity (and wlog), the alphabet is $\{0, 1\}$.

coNP

1. (Easy) Recall the verifier definition of the complexity class NP. A language $L \subseteq \{0,1\}^*$ is in NP if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time TM M such that for every $x \in \{0,1\}^*$,

$$x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)} \text{ s.t. } \mathrm{Result}(M, \langle x, u\rangle) = \mathsf{yes}.$$

   If $x \in L$ and $u \in \{0,1\}^{p(|x|)}$ satisfy $\mathrm{Result}(M, \langle x, u\rangle) = \mathsf{yes}$, then we call u a *certificate* for x (with respect to the language L and machine M).

   Define the complexity class $\mathsf{coNP} = \{L : \overline{L} \in \mathsf{NP}\}$, where $\overline{L} = \{0,1\}^* \setminus L$.

   Give a definition for the class coNP that parallels the definition of NP above; i.e., in terms of certificates.

2. (Easy) Prove that $\mathsf{coNP} \subseteq \mathsf{EXP}$.

3. (Easy) Prove that, if $\mathsf{P} = \mathsf{NP}$, then $\mathsf{NP} = \mathsf{coNP}$.

4. (Easy) Suppose that $L \in \mathsf{NP}$ and you have a polynomial-time **N**TM N that decides L. Can you use N to decide $\overline{L}$ in (nondeterministic) polynomial-time ?

5. (Easy) **TRUE/FALSE**. NP is not a proper subset of coNP.

6. (Easy) **TRUE/FALSE**. $\mathsf{NP} \cap \mathsf{coNP}$ is closed under complement.

7. (Medium) **TRUE/FALSE**. If an NP-hard problem lies in coNP, then $\mathsf{NP} \subseteq \mathsf{coNP}$.

8. (Easy) Is coNP the complement of NP ? If not, then what does the class $\mathsf{NP}\cap\mathsf{coNP}$ intuitively mean ? (in terms of the sizes of yes/no certificates);

9. (Easy) Show that $\mathsf{P} \subseteq \mathsf{NP} \cap \mathsf{coNP}$.
   (Extremely Difficult) Is $\mathsf{NP} \cap \mathsf{coNP} \subseteq \mathsf{P}$?

10. Let

$$\mathsf{TAUTOLOGY} = \{\varphi : \varphi \text{ is a Boolean formula that is satisfied by every assignment}\}.$$

   (a) (Easy) Show that $\mathsf{TAUTOLOGY}$ is coNP-complete. **Hint:** Consider the problem

$$\mathsf{NOSAT} = \{\varphi : \varphi \text{ is a Boolean formula and no variable assignment satisfies } \varphi\}.$$

   Show that $\mathsf{NOSAT} \in \mathsf{coNP}$, and show that NOSAT is coNP-complete by slightly modifying the proof of the Cook-Levin Theorem. Then, write $\mathsf{TAUTOLOGY}$ in terms of NOSAT.

   (b) (Extremely Difficult) Give a problem that is $\mathsf{NP} \cap \mathsf{coNP}$-complete;

(c) (Medium) Show that $\mathsf{NP} = \mathsf{coNP}$ iff $\mathsf{3SAT}$ and $\mathsf{TAUTOLOGY}$ are polynomial-time reducible to one another.

11. (Medium) Define

$$\begin{aligned} \mathsf{MINVC}_k &= \{ \langle G \rangle : \text{the minimum vertex cover in } G \text{ has size exactly } k \} \\ \mathsf{VC}_k &= \{ \langle G \rangle : G \text{ has a vertex cover of size} \le k \} \end{aligned}$$

Consider the following purported proof that $\mathsf{MINVC} \in \mathsf{NP} \cap \mathsf{coNP}$.

Proof. The graphs whose minimum vertex cover has size exactly k are precisely the graph which

- have a vertex cover of size $\le k$, and
- have no vertex cover of size $\le k - 1$.

Thus, $\mathsf{MINVC}_k = \mathsf{VC}_k \cap \overline{\mathsf{VC}_{k-1}}$. Since $\mathsf{VC}_k \in \mathsf{NP}$, we have $\overline{\mathsf{VC}_{k-1}} \in \mathsf{coNP}$. This shows that MINVC_k is the intersection of a language in NP, and a language in coNP, so $\mathsf{MINVC}_k \in \mathsf{NP} \cap \mathsf{coNP}$. □

Is this a valid proof? If so, explain how it can be made precise. If not, explain what the flaw is. In either case, ensure that your answer is explained carefully.

12. (Medium) Let

$$\mathsf{MAXCLIQUE} = \{\langle G, k \rangle : \text{the largest clique in } G \text{ has exactly } k \text{ nodes in it}\}.$$

Explain why the following argument fails to show that $\mathsf{MAXCLIQUE} \in \mathsf{coNP}$: To show that $\langle G, k \rangle \in \mathsf{MAXCLIQUE}$, it suffices to demonstrate the existence of a larger clique in G of size greater than k, so the NP algorithm for $\mathsf{MAXCLIQUE}$ just guesses the larger clique.

13. It is known that $\mathsf{PRIMES} = \{\langle p \rangle : p \text{ is a prime number}\}$ is in P; i.e., given integer p, there is an algorithm that decides whether or not p is prime in time that is polynomial in $\log(p)$[1]. Way before this algorithm was discovered, it was shown that $\mathsf{PRIMES} \in \mathsf{NP} \cap \mathsf{coNP}$. Given these facts, consider the following problem:

- **Input:** Positive integer N;
- **Output:** Prime factorization of N; $N = p_1^{i_1} p_2^{i_2} \cdots p_k^{i_k}$, where k is a non-negative integer, $p_1, \ldots, p_k$ are positive prime numbers, and $i_1, \ldots, i_k$ are positive integers.

(a) (Medium) Formulate this problem as a decision problem and explicitly write down the language corresponding to it;

(b) (Medium) How can you use an algorithm that is polynomial (in $\log N$) for the decision version to solve the original (function) problem in time $\text{poly}(\log N)$?

(c) (Easy) Show that your decision version is in NP;

(d) (Medium) Show that your decision version is in coNP.

[1] "M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. Annals of Mathematics, 160:781–793, 2004."