Exercise sheet 4

Exercise 4.1. Tautology is the problem of deciding whether *every* truth assignment satisfies a given propositional formula:

$$\text{Tautology} = \{\langle\phi\rangle \mid \phi \text{ is a logically true propositional formula}\}.$$

▷ Show that Tautology is coNP-complete.

Exercise 4.2.

(a) Show that P is closed under complement.

(b) Show that if P = NP, then NP = coNP.

Exercise 4.3. A complexity class $\mathcal{C}$ is said to be *closed under polynomial-time many-one reductions* if

$$A \leq_p B \text{ and } B \in \mathcal{C} \implies A \in \mathcal{C}.$$

(In Lecture 2, we showed that P is closed under polynomial-time many-one reductions.)

(a) Show that NP is closed under closed under polynomial-time many-one reductions.

(b) Show that coNP is closed under closed under polynomial-time many-one reductions.

Exercise 4.4. Let A be an NP-complete problem. Show that if $A \in$ coNP, then NP = coNP.

Exercise 4.5. Suppose that $A \in$ NP and $B \in$ coNP. Assume for the purpose of the exercise that NP $\neq$ coNP. Prove each of the following, or give a counterexample.

(a) $\overline{A} \cup B \in$ coNP

(b) $A \cap \overline{B} \in$ coNP

(c) $A \cup B \in$ coNP

(d) $A \cap B \in$ coNP

Exercise 4.6. Show that Factoring is in NP $\cap$ coNP.

Exercise 4.7. What is wrong with the following proof of P $\neq$ NP?

Assume that P = NP. Then, there exist an algorithm A and a polynomial $p(n)$ such that SAT is decided by A in time $O(p(n))$. Assume that $p(n) = O(n^{37})$. By the Time Hierarchy Theorem, there exists a problem $P \in \text{DTIME}(n^{38})$ such that $P \notin \text{DTIME}(n^{37} \log n^{37}) = \text{DTIME}(n^{37} \log n)$. Since SAT is NP-complete, we can reduce P to SAT and decide it in time $O(n^{37})$. But we have just shown that P requires time $\omega(n^{37} \log n)$. This leads to a contradiction, hence the assumption P = NP must be false. □

Exercise 4.8. Show that $\Pi_k^p = \text{co}\Sigma_k^p$, for every $k \geq 0$.

Exercise 4.9. Show that if $\Sigma_k^p \subseteq \Pi_k^p$, then $\Sigma_k^p = \Pi_k^p$.

Exercise 4.10. Show that if P = NP, then P = PH.

Exercise 4.11. Show that if there exists a PH-complete problem, then PH $= \Sigma_k^p$ for some k.

Exercise 4.12. Suppose that you are given a formula ϕ on DNF and that you want to know whether there exists a smaller equivalent DNF formula ψ. The following definition formalizes this problem:

$$\textsf{Min-DNF} = \{\langle \phi, k \rangle \mid \exists \text{ DNF } \psi \text{ s.t. } \psi \equiv \phi \text{ and } \psi \text{ has} \leq k \text{ occurrences of literals}\}$$

For example, the DNF formula $(x_1 \wedge \bar{x}_2 \wedge x_3) \vee (\bar{x}_1 \wedge x_3)$ has 5 occurrences of literals.

▷ Show that $\textsf{Min-DNF} \in \Sigma_2^p$.

Exercise 4.13. The complexity class DP is defined as those decision problems that can be written as an intersection of an NP problem and a coNP problem. Formally,

$$\text{DP} = \{A_1 \cap A_2 \mid A_1 \in \text{NP}, A_2 = \text{coNP}\}.$$

(a) Show that $\textsf{3SAT-3UNSAT} = \{(\phi, \psi) \mid \phi \in \textsf{3SAT}, \psi \notin \textsf{3SAT}\}$ is DP-complete.

(b) Let $\alpha(G)$ be the *independence number of the graph* G, that is, the size of a maximum independent set in G. Show that $\textsf{Exact IndSet} \in$ DP, where $\textsf{Exact IndSet}$ is the language of all inputs $\langle G, k \rangle$ such that $\alpha(G) = k$.

(c) For two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the *lexicographic product of* G_1 *and* G_2 is defined as the graph G with vertices $V_1 \times V_2$ and an edge between (u_1, u_2) and (v_1, v_2) if $\{u_1, v_1\} \in E_1$ or if $u_1 = v_1$ and $\{u_2, v_2\} \in E_2$. Show that $\alpha(G) = \alpha(G_1) \cdot \alpha(G_2)$.

(d) Show that every problem in DP is polynomial-time many-one reducible to $\textsf{Exact IndSet}$, i.e., $\textsf{Exact IndSet}$ is DP-complete.

(e) Show that NP $\cup$ coNP $\subseteq$ DP $\subseteq \Sigma_2^p \cap \Pi_2^p$.