Exercise II, Computational Complexity 2024

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

1 Prove the following basic properties of polynomial-time reductions.

(a) $A \leq_p A$. (*reflexivity*)

(b) If $A \leq_p B$ and $B \leq_p C$, then $A \leq_p C$. (*transitivity*)

(c) If $A \leq_p B$ then $\bar{A} \leq_p \bar{B}$.

(d) If $A \leq_p B$ and $B \in \mathsf{NP}$, then $A \in \mathsf{NP}$.

(e) If $A \in \mathsf{P}$ then $A \leq_p \{1\}$.

2 What is wrong with the following "proof" that 2-SAT is NP-complete? Since 3-SAT is in NP, so is 2-SAT. We then prove that 3-SAT $\leq_p$ 2-SAT by giving a polynomial time computable reduction. We define a function $\varphi \mapsto f(\varphi)$ which maps boolean 2-CNF fomulas to 3-CNF formulas. For a formula $\varphi = (a_1 \vee b_1) \wedge (a_2 \vee b_2) \wedge \cdots \wedge (a_n \vee b_n)$, where a_i, b_i are literals, we define $f(\varphi) = (a_1 \vee b_1 \vee b_1) \wedge (a_2 \vee b_2 \vee b_2) \wedge \cdots \wedge (a_n \vee b_n \vee b_n)$. Note that

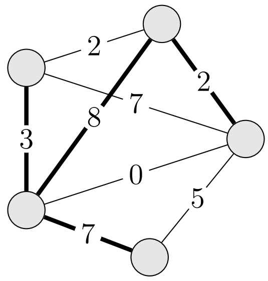
$$\varphi \text{ is satisfiable} \Leftrightarrow f(\varphi) \text{ is satisfiable.}$$

Clearly, f is polynomial time computable given φ. Hence 3-SAT $\leq_p$ 2-SAT, and in particular it follows that 2-SAT is NP-complete.

3 Show that if any NP-complete problem lies in coNP, then NP $=$ coNP.

4 Prove that the following problem is NP-complete: Given a set S, a collection $\mathcal{C}$ of subsets of S and a number k, is there a subset $T \subseteq S$ of size k such that $T \cap C_i \neq \emptyset$ for all $C_i \in \mathcal{C}$? *(Hint: reduce from* VertexCover*.)*

5 Let $G = (V, E)$ be a graph and $w \colon E \to \mathbb{N}$ an assignment of non-negative integer weights to the edges. A subset of edges $E' \subseteq E$ is a *spanning tree* if the subgraph (V, E') is a tree (no cycles) that connects all vertices. The weight of the spanning tree is $\sum_{e \in E'} w(e)$. For example, the bold edges below form a spanning tree of weight $3 + 7 + 8 + 2 = 20$.



In the Exact Spanning Tree problem (ESP for short), the input consists of a graph $G = (V, E)$, edge weights $w\colon E \to \mathbb{N}$, and a target $k \in \mathbb{N}$. The goal is to decide whether G contains a spanning tree of weight exactly k. That is,

$$\text{ESP} \;=\; \{\langle G, w, k\rangle : G \text{ contains a spanning tree of weight exactly } k\}.$$

Prove that ESP is NP-complete by finding a reduction Subset-Sum $\leq_p$ ESP.

6 (*) A multi-variate polynomial equation with integer coefficients is called a *Diophantine equation*; for example, $x^2y - y^2 + 6 = 0$. Let DEQ be the language consisting of all (binary encodings of) Diophantine equations that admit an integer solution; for example, $x = 1$, $y = -2$ for the above equation. Find a reduction 3-SAT $\leq_p$ DEQ. Do we have DEQ $\in \mathsf{NP}$?