

**Homework II** (due Tuesday, March 21st)

**Problem 1** (NP-hardness). Prove that the following problems are NP-hard

- The halting problem $\text{Halt} = \{ (\alpha, x) : M_\alpha \text{ halts on input } x \}$.
- The problem Not-All-Equal SAT

  $\text{NAE-3-SAT} = \{ (c_1, \ldots, c_n) \text{ where each } c_i \text{ is composed of 3 literals } c_i = (x_{i1}, x_{i2}, x_{i3}) :$

  there is an assignment s.t. the values assigned to the literals of every $c_i$ are not all equal$\}$

  For example, $(c_1, c_2) = ((x_1, x_2, x_3), (x_1, \neg x_2, \neg x_3))$ is in the language because I can assign $x_1 = 0, x_2 = 0, x_3 = 1$ and then for the two triples, there is a literal taking a value 0 and another taking value 1. However, the assignment $x_1 = 1, x_2 = 1, x_3 = 1$ is not a satisfying one in the NAE-3-SAT sense, even though it is a satisfying assignment if we were interpreting the instance $(c_1, c_2)$ as a 3-SAT instance.

- The problem Integer Linear Programming

  $\text{ILP} = \{ (A, b) : A \text{ integer } n \times m \text{ matrix and } b \text{ column vector of } m \text{ integers}$

  s.t. there exists a vector $x$ of $n$ integers such that $Ax \geq b$ $\}$

  We use the notation $v \geq w$ for vectors $v$ and $w$ when $v_i \geq w_i$ for all coordinates $i$.

  Are they NP-complete? For this, you may choose one of three options with justification: not in NP, in NP, seems to be in NP but there is a difficulty and say what the difficulty is.

**Problem 2** (Reductions). We saw that a language $L$ is polynomial-time Karp reducible to $L'$ if there is a polynomial-time computable functions $f$ such that $x \in L$ iff $f(x) \in L'$. Give an example of a class that is closed under polynomial-time Karp reductions (i.e., if $L$ is reducible to $L'$ and $L' \in C$, then $L \in C$) and one that is not.

**Problem 3** (Difference of NP problems). Let $\text{DP} = \{ L_1 \setminus L_2 : L_1 \in \text{NP and } L_2 \in \text{NP} \}$. Show that the problem $\text{ExactIndSet} = \{ (G, k) : \text{ the largest independent set of } G \text{ has size exactly } k \}$ is DP-complete (for the usual polynomial-time Karp reductions).

**Problem 4** (Graph isomorphism – Bonus). Two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ on the same vertex set are said to be isomorphic if there exists a permutation $\pi$ of $V$ such that $(u, v) \in E_1 \iff (\pi(u), \pi(v)) \in E_2$. Suppose that there is a polynomial-time algorithm $A$ that decides whether two given graphs $G_1$ and $G_2$ are isomorphic. Using $A$ as a subroutine, describe a polynomial-time algorithm that finds the permutation $\pi$ when the graphs are isomorphic.