Computability - Exercise 11

All questions should be answered *formally* and *rigorously* unless stated otherwise

Due: June 25, 2020

Question 1

Classify the following languages to P, NP-hard, or coNP-hard. Prove your answer.
Reminder: a language L is coNP-hard if for every language $L' \in$ coNP, it holds that $L' \leq_p L$.

1. (**Not for submission**) A *2-dominating set* in an undirected graph $G = \langle V, E\rangle$ is a set $S \subseteq V$ such that for every $u \in V$, either $u \in S$, or there is $v \in S$ such that $\{u, v\} \in E$, or there are $v \in S$ and $w \in V$ such that $\{u, w\}, \{w, v\} \in E$.
   Now, 2-DS = $\{\langle G, k\rangle : G$ is an undirected graph containing a 2-dominating set of size $k\}$.
2. $SPATH = \{\langle G, s, t, k\rangle : G$ is an undirected graph containing a simple path of length at most k from s to $t\}$
3. $U\text{-}ST\text{-}HAMPATH^{\leq} = \{\langle G, s, t, 1^k\rangle : G$ is an undirected graph that has no more than k Hamiltonian paths from s to $t\}$
4. $U\text{-}ST\text{-}HAMPATH^{\geq k} = \{\langle G, s, t\rangle : G$ is an undirected graph that has at least k Hamiltonian paths from s to $t\}$
5. (**Not for submission**) A weighted graph is $G = \langle V, E, w\rangle$, where $w : E \to \mathbb{N}$ assigns a weight to each edge. The *weight* of a simple path $P \subseteq E$ is $w(P) = \sum_{e \in P} w(e)$.

   Now, TSP = $\{\langle G, w, k\rangle : G$ is a weighted undirected graph with a Hamiltonian cycle of weight at most $k\}$.

Question 2

Classify the language L below to P or NP-complete. Prove your answer.
$L = \{\langle G\rangle \mid G$ is an undirected graph and there exist 3 cliques $V_1, V_2, V_3 \subseteq V$ such that $V = V_1 \cup V_2 \cup V_3\}$.
Note that the sets V_1, V_2, and V_3 need not be disjoint.

Question 3

In the *set-cover* problem, we are given a set U of n elements, a collection $S_1, \dots, S_m$ of m subsets of U ($S_i \subseteq U$ for all $1 \leq i \leq m$), and an integer k. The goal is to decide whether there is a collection C of at most k sets such that their union is equal to U. That is, $C \subseteq \{S_1, ..., S_m\}$ is such that $|C| \leq k$ and $\bigcup_{S_i \in C} S_i = U$. As a motivating example, assume that U is a set of characters like "tall", "pregnant", "young",..., and that there are m candidates who want to join a journey to the moon. Each candidate $1 \leq i \leq m$ has a set S_i of his or her characters. We want to make sure that we cover all characters when we select the lucky passengers, but we have only k seats in our spaceship.
Prove that the set-cover problem is NP-complete.

Question 4

For every natural k, a *k-coloring* of a graph $G = \langle V, E\rangle$ is a function $c : V \to \{1, \dots, k\}$, such that for every $\{u, v\} \in E$, we have that $c(u) \neq c(v)$. Thus, vertices connected by an edge are mapped to different colors. We say that a graph G *admits* a k-coloring if such a coloring of G exists. Let $k\text{-}COLOR = \{\langle G\rangle : G$ admits a k-coloring$\}$.

1. Show that 1-$COLOR$ and 2-$COLOR$ are in P (hint for 2: bipartite graphs).

2. Show that 3-$COLOR$ is NP-complete.

   Instruction: You may use the following construction (a reduction from 3-SAT to 3-$COLOR$) and prove its correctness and its running time, aside from proving that the language is in NP.

   Construction: Let φ be a 3-CNF formula. We output the graph $G = \langle V, E\rangle$ constructed as follows. First, G has 3 special vertices that consist the *palette* gadget, which is simply a 3-clique. We call these vertices T, F, and U. Next, for every variable x, we have two vertices x and $\overline{x}$, and the edge $\{x, \overline{x}\}$ (this is the variable gadget). We also add, for every variable x, the edges $\{U, x\}$ and $\{U, \overline{x}\}$. Then, we define a clause gadget to be a composition of two OR-gadgets, one on top of the other, as depicted in Figure 1. We designate 3 vertices in the clause gadget as i_1, i_2, i_3, and the top vertex as o, as depicted. For each clause $(l_1 \vee l_2 \vee l_3)$ we have a clause-gadget. We connect the top vertex o to U and F from the palette. The vertices i_1, i_2 and i_3 are the literal vertices l_1, l_2 and l_3 respectively (that is, they are the same vertices, and the clause gadget is "composed" on top of them). This completes the construction.

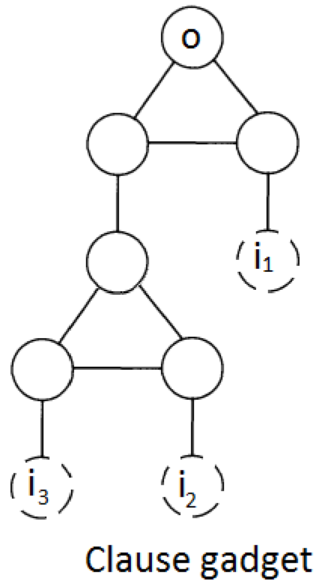

Figure 1: Clause gadget

3. Show that for every natural k, if k-$COLOR$ is NP-complete, then so is $(k+1)$-$COLOR$.
   Remark: It follows inductively that k-$COLOR$ is NP-Complete for every $k > 2$.