Schedule S1, (Computer Science, CS and Philosophy, Maths and CS)
Schedule B, MSc in Advanced Computer Science

Hilary Term 2022

Computational Complexity

Exercise class 2: NP, co-NP, reductions

1. Recall from lectures that Tautology is co-NP-complete. Classify the computational complexity of the problems DNF-Tautology and CNF-Tautology, which are the special cases of Tautology in which the input formula is in DNF and CNF, respectively.

2. Given undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ a homomorphism from G_1 to G_2 is a function $h : V_1 \longrightarrow V_2$ satisfying the following property: for every edge $\{u, v\} \in E_1$ we have that $\{h(u), h(v)\} \in E_2$. The language Homomorphism is defined as follows:

$$\text{Homomorphism} = \{\langle G_1, G_2 \rangle \ : \ \text{there is a homomorphism from } G_1 \text{ to } G_2\}$$

   A *vertex colouring* of a graph G with k colours is a function

$$c : V(G) \longrightarrow \{1, \dots, k\}$$

   such that adjacent nodes in G have different colours, i.e., $\{u, v\} \in E(G)$ implies $c(u) \neq c(v)$. k-Colouring is the problem of determining if a given graph G has a vertex colouring with k colours. The language k-Colouring is defined as follows:

$$k\text{-Colouring} = \{G \ : \ G \text{ has a vertex colouring with } k \text{ colours}\}.$$

   Let G be an undirected graph and let k be an integer. G contains a clique of order k if there exists some subset $S \subseteq V(G)$ with $|S| = k$ such that there exists an edge $\{x, y\}$ for every pair of distinct vertices $x, y \in S$. The language Clique is then defined as follows:

$$\text{Clique} = \{\langle G, k \rangle \ : \ G \text{ is an undirected graph containing a clique of order } \geq k\}$$

   Do the following:

   (a) Show that k-Colouring is polynomial-time reducible to Homomorphism.

   (b) Assuming that Clique is NP-complete, show that Homomorphism is NP-hard.

   (c) Show that if a language $\mathcal{L}$ is NP-complete, then its complement $\overline{\mathcal{L}}$ is co-NP-complete.

3. A *vertex cover* of a graph G is a set $X \subseteq V(G)$ such that for all edges $\{u, v\} \in E(G)$ at least one of u, v is in X. The problem Vertex Cover is the problem of deciding for a given graph G and number $k \geq 1$ if G contains a vertex cover of size $\leq k$.

   A *dominating set* of a graph G is a set $X \subseteq V(G)$ such that for any vertex v that does not belong to X, v has a neighbour in X. The problem Dominating Set is the problem of determining for a given graph G and $k \in \mathbb{N}$ if G has a dominating set of size at most k.

   Using the fact that Vertex Cover is NP-complete, show that Dominating Set is also NP-complete.

4. Given an undirected graph $G = (V, E)$, a *spanning tree* for G is a subgraph of G that is a tree and includes all vertices of G. We define the minimum-leaf spanning tree problem as follows.

   $\textsc{MinLeafST} = \{\langle G, k\rangle \ :\ G$ an undirected graph containing a spanning tree having at most k leaves$\}$

   A *simple path* in G is a path with no repeated nodes. Consider also the following problems:

   $$\textsc{HamPath} = \{G \ :\ G \text{ has a simple path involving all nodes in } G\}$$
   $$\textsc{LongestSimplePath} = \{\langle G, k\rangle \ :\ G \text{ has a simple path of length} \geq k\}$$

   Do the following:

   (a) Using the fact that HamPath is NP-complete, show that both MinLeafST and LongestSimplePath are also NP-complete.

   (b) Identify the error in the following: "Let

   $$\textsc{ShortestSimplePath} = \{\langle G, k\rangle \ :\ G \text{ has a simple path of length} \leq k\}$$

   LongestSimplePath is NP-complete and the complement of ShortestSimplePath; hence, ShortestSimplePath is co-NP-complete."