Computational Models, Spring 2013 Exrecise #6

$\mathcal{NP}$-Complete languages

By "graph", we mean undirected graph. By "number", we mean natural number.

1. For the following decision problems, determine whether they are in $\mathcal{P}$ or in $\mathcal{NPC}$ (assuming $\mathcal{P} \neq \mathcal{NP}$). Prove your answer.

   (a) Input: sets $A_1 \dots A_n$, and a number k.
   Question: does there exist a set C of size k, such that for every $1 \leq i \leq n$ $A_i \cap C \neq \emptyset$?

   NPC. Reduction from VC. Given input G = (V, E), k for VC, we construct a set A_i for every $\{v_1 v_2\}$ ∈E and let $A_i = \{v_1 v_2\}$ We keep the same k.

   (b) Input: a 3CNF formula ψ Question: does there exist an assignment that satisfies ψ and gives **True** for exactly 10 variables?

   P. We check all possible sets of 10 variables.

   (c) Input: a 3CNF formula ψ
   Question: do there exist at least two assignments that satisfy ψ?

   NPC. Reduction from 3SAT. Given input ψ , we add a clause $(z \vee \bar{z} \vee \bar{z})$ where z is a new variable.

   (d) Input: graph G.
   Question: does there exist a Hamiltonian path in G (between any pair of vertices)?

   Reduction from UHAMPATH. Given input G, s, t we construct a graph G' by adding two new vertices u_1, u_2 to G and connecting u_1 to s and u_2 to t.

   (e) Input: graph G and a number k.
   Question: does there exist a simple path in G of length $\geq k$?

   NPC. Reduction from previous item. Given input $G = (V, E)$, we construct the input $G, |V| - 1$.

   (f) Input: graph G and a number k.
   Question: is there a Vertex-Cover S in G of size k and an IndependentSet, T, of size $\frac{k}{2}$, such that $T \subseteq S$?

   NPC. Reduction from VC. Given input G, k for VC, we construct input G', k' for our problem. G' is obtained from G by adding it a distinct simple cycle of length 2k. k' = 2k.

2. Prove that if $\mathcal{P} = \mathcal{NP}$ then every language in $\mathcal{P}$, except $\emptyset$ and Σ^* is $\mathcal{NPC}$. Why can't $\emptyset$ and Σ^* be $\mathcal{NPC}$?

   Let A be any language in NP and let B be another language not equal to $\emptyset$ or Σ^*. Then there exist strings $x \in B$ and $y \notin B$. To reduce an instance w of A to that of B, we just check in polynomial time if $w \in A$. If yes, we output x and y when $w \notin A$.
   The languages $\emptyset$ and Σ^* cannot be NP-complete, because to reduce a language A to a language B, we need to map instances in A to instances in B and those outside A to outside B. However, for $B = \emptyset$, there are no instances in B (and none outside B for $B = \Sigma^*$) which means there cannot be such a reduction from any language $A \neq \emptyset, \sigma^*$.

3. A c-coloring of an undirected graph $G = (V, E)$ is a map $\chi : V \to \{1 \ldots c\}$ such that adjacent vertices (u, v) get different colors i.e. $\chi(u) \neq \chi(v)$. Let

$$c - COLORING = \{G \ : \ G \text{ is a graph that can be } c\text{-colored}\}$$

   (a) Prove that 3-coloring is $\mathcal{NPC}$.

   In order to show that 3-coloring is NP-complete we will reduce 3SAT to it. This is a classic reduction, and a nice exposition of it can be found at `http://www.ugrad.cs.ubc.ca/~cs320/2010W1/handouts/3color.html`.

   (b) Prove that 2-coloring is in $\mathcal{P}$.

   It is clearly enough to show that any connected component of a graph G is 2-colorable. In a connected component C, let v be an arbitrary vertex and color it red. As long as there are uncolored vertices in C, do the following: for any $v \in C$, if v is colored red then color its uncolored neighbors blue; and if v is colored blue color its uncolored neighbors red. If there is an edge with two endpoints colored the same the graph is not 2-colorable, and otherwise this will produce a 2-coloring of C. Note that this coloring of C is unique up to renaming the colors. In an intuitive level, the reason 2-coloring is easy is because given a color of a vertex there is a unique way to color its neighbors in a consistent way.

4. Define $SETCOVER$ to be

$$\{(U, S_1, \ldots S_m, k) \text{ s.t. } \forall i S_i \subset U \text{ and there is } I \subseteq \{1 \ldots m\} \text{ with } |I| = k \text{ and } U = \bigcup_{i \in I} \{S_i\}\}$$

   Show that $SETCOVER$ is $\mathcal{NPC}$.

   Solution: SET-COVER can be seen as a generalization of VERTEX COVER. For a given graph G, consider each vertex as a set of the edges incident upon it. Formally, we define the sets $S_u = \{(u, v) \in E \mid v \in V\}$ for all $u \; inV$. Also, we take $U = E$ and let k be the same as in the vertex cover instance. Then a SET COVER of this family of sets corresponds exactly to picking vertices (sets) such that at least one vertex corresponding to each edge is picked (i.e. at least one set containing every element is picked). Hence, the graph G has a vertex cover of size at most k if and only if the above instance has a set cover of size k. (Note that if there is a cover of size less than k, then there is also one of size exactly k since we can always add a few extra sets.) To show that the problem is in NP, it suffices to note that given an $I \subseteq \{1, \ldots, m\}$, we can verify in polynomial time that $|I| = k$ and $U = \bigcup_{i \in I} \{S_i\}$.