Assignment 25

NP-completeness

1. SAT_2 is the language $\{< \phi > | \phi \text{ is a boolean formula with at least 2 satisfying assignments}\}$.
    1. Provide an element of the language.
    2. Provide an element of the same type but that is not in the language.
    3. Show that SAT_2 is in NP.
    4. Show that SAT is polynomial-time reducible to SAT_2. **Hint:** Given a boolean expression, tack on another expression at the end that can be satisfied in 2 ways, such that the overall expression is satisfiable in at least 2 ways if and only if the original expression was satisfiable, otherwise the final result should not satisfiable at all.
    5. Given that SAT can be reduced in polynomial time to SAT_2 and that SAT_2 is in NP, what does this say about SAT_2?
2. Given that L is NP-complete, why is $\overline{L}$ not also necessarily NP-complete?

Assignment 26

NP-completeness pt. 2

Recall the G_{ISO} language from a few homeworks ago:
$\{< G_1, G_2 > | \exists \text{ an isomorphism between the two graphs}\}$. An isomorphism between two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is a bijection $f : V_1 \to V_2$ such that
$(u, v) \in E_1 \iff (f(u), f(v)) \in E_2$.

A similar, related language is the G_{SUB_ISO} language, which is defined as follows:
$\{< G_{small}, G_{big} > | \exists \text{ an isomorphism between } G_{small} \text{ and a subgraph of } G_{big}\}$.

Terminology: A *subgraph* of a graph is any subset of its vertices and edges, but note that the vertex subset must include all endpoints of the edge subset (it may also include additional vertices).

Show that G_{SUB_ISO} is NP-complete by showing it is in NP and reducing the clique problem to it.

Hint: Recall that the input to the clique problem is a graph and a non-negative integer. Use that number to construct another graph such that the original graph and the constructed graph are in G_{SUB_ISO} (but not necessarily in that order!) if and only if there is a clique of that size in the original graph.

Assignment 27

NP-completeness part 3

$STANDALONE$ is the language
$\{< G, k > | k \in \mathbb{N} \wedge G \text{ is an undirected graph with at least } k \text{ vertices that share no edge}\}$.

$SUBSETS\text{-}ENVELOP$ is the language
$\{< U, S, k > | k \in \mathbb{N} \wedge S \subset P(U) \text{ such that the union of at most } k \text{ elements of } S \text{ equals } U\}$.

For both of these languages:

1. Provide an example of an element in the language.
2. Provide an example of the same type of data but that is not in the language.
3. Show that the language is in NP
4. Show that the language is NP-complete by reducing the vertex cover problem to it.