CSC373H1F 2013 **Assignment A3**
Due: Friday, November 29, 10PM Total Score: 10 points

Please follow the instructions provided on the course website to submit your assignment.

In the following questions you will be given a computational problem and you must provide a verifier algorithm or a reduction algorithm.

When giving a polynomial-time verifier algorithm to show a problem is in NP problem, you must provide both **a short high level explanation of how your algorithm works in plain English** and **the pseudo-code of your algorithm** similar to those we have seen in the lectures. You must also provide a bound on the size of the certificates in the size of the original input. State the running time of your algorithm with a brief argument supporting your claim. You must prove the correctness of your algorithm.

When giving a polynomial-time reduction between two problems you must state which computational problem is being reduced to which one, define a reduction function between them, and show that your reduction function is computable in polynomial time. You must also prove the correctness of your reduction.

Questions

Reminders and Definitions

- **Graph k-Coloring**
  Let G be a graph and k be an integer. A k-coloring of a graph G is an assignment of colors $\{1, 2, \ldots, k\}$ to the nodes in G such that for each edge in G the colors assigned to its endpoints are distinct. More formally, a k-coloring of graph G is a function $c : V(G) \to \{1, 2, \ldots, k\}$ s.t. for all $vu \in E(G)$, $c(v) \neq c(u)$.
  The k-coloring decision problem is defined as follows:

$$k\text{-}Coloring = \{\langle G, k\rangle \mid G \text{ has a } k\text{-coloring}\}$$

- **Graph Homomorphism**
  Let G and H be two graphs. A graph homomorphism from G to H is a mapping of G to H which preserves edges, i.e. maps edges in G to edges in H. More formally, a graph homomorphism from G to H is a function $f : V(G) \to V(H)$ such that for all $vu \in E(G)$, $f(v)f(u) \in E(H)$.
  The graph homomorphism decision problem is defined as follows:

$$GraphHomo = \{\langle G, H\rangle \mid \text{there is a homomorphism from } G \text{ to } H\}$$

- **NP certificate verifier**

  Let Q be a decision problem in NP, let V be a polynomial-time certificate verifier for Q, and let q be a polynomial bounding the size of certificates. So for any input x of size $n = |x|$ we have: $Q(x) = Yes$ if and only if there is a certificate c of size at most $q(n)$ such that $V(x, c) = Yes$.

Example The verifier for $k - Coloring$:

A certificate c is a list of length $|V|$ of numbers from $\{1, \ldots, k\}$ giving the colors for the vertices.

The verifier checks if for all edges vu in $E(G)$, $c(v) \neq c(u)$.

The size of input is $n = \theta(|V| + |E| + \lg k)$ (assuming that the graph is given by its list of edges).

The size of c is $O(|V| \lg k) = O(n^2)$.

The running-time is $O(E) = O(n)$.

- **Search Problem**

  The *search problem* for Q based on V is defined as follows: given input x, find a certificate c of size at most $q(n)$ such that $V(x, c)$.

Example The search problem for $k - Coloring$: given G and k, find a k-coloring of G.

- **Certificate Extension Problem** The *certificate extension decision problem* for Q based on V is defined as follows: given input x and a string y (intended to be a partial solution for the search problem) is there certificate c "extending" y such that $V(x, c) = Yes$?

  More formally, $QExt(x, y) = Yes$ if and only if there is a certificate c of size at most $q(n)$ such that $V(x, c) = Yes$ and $y \sqsubseteq c$, where $y \sqsubseteq c$ means c "extends" y.

Example If Q is k-coloring, a *partial* solution can be a list of number of length $l \leq |V|$ providing colors from $\{1, \ldots, k\}$ to the first l vertices.

We say (in fact we define) certificate c "extends" certificate y if and only if c assigns the same colors to vertices as y does. c might also assign colors to vertices which y does not assign any color, but for any vertex that y assigns a color to c has to assign the same color.

The certificate extension problem for k-coloring is then: given a graph G and an assignment of colors to some of its vertices y, is there a valid k-coloring of G which extends y?

1. *GraphHomo* $\in$ NP [20]

   Show that the graph homomorphism problem is in NP by providing a polynomial-time certificate verifier algorithm for it.

   a. Design a verification algorithm. Briefly explain the high-level idea behind your algorithm and provide its pseudo-code. State what is the intended meaning of your certificates.

   b. Give a *tight* bound (i.e. Θ) for the size of input (n) in terms of the parameters for the input ($|V(G)|$, $|E(G)|$, $|V(H)|$, $|E(H)|$).

   c. Give a good upper-bound on the size of the certificates in terms of the size of the input. Briefly justify your answer.

   d. Give a good upper-bound on the worst-case running time of your algorithm in terms of the size of the input. Briefly justify your answer.

   e. Briefly argue for the correctness of your verification algorithm. (Show that for any Yes instances there is some certificate accepted by your verifier, and for any No instances there is no certificate accepted by your verifier.)

2. *GraphHomo* **and** *GraphHomoExt* [25]

   In this question we will look at the search and extension problems for the graph homomorphism problem based on the verifier you have defined in the previous question.

   a. Define the certificate extension problem for the graph homomorphism problem.

   b. Give a polynomial-time reduction from the graph homomorphism problem to the graph homomorphism certificate extension problem.

   c. Briefly argue that your reduction is correct.

   d. Give a polynomial-time reduction from the graph homomorphism certificate extension problem to the graph homomorphism problem.

   e. Briefly argue that your reduction is correct.

3. *GraphHomoSearch* **and** *GraphHomoExt* [25]

   In this question we will look at the search and extension problems for the graph homomorphism problem based on the verifier you have defined in the previous question.

   a. Define the search problem for the graph homomorphism problem.

   b. Give a polynomial-time reduction from the graph homomorphism search problem to the graph homomorphism certificate extension problem. In other words, assume that we are given a black-box that solves the graph homomorphism certificate extension problem. Give a polynomial-time algorithm that solves the graph homomorphism search problem using that black-box.

   c. Briefly argue that your reduction is correct.

d. Give a polynomial-time reduction from the graph homomorphism certificate extension problem to the graph homomorphism search problem. In other words, assume that we are given a black-box that solves the graph homomorphism search problem. Give a polynomial-time algorithm that solves the graph homomorphism certificate extension problem using that black-box.

e. Briefly argue that your reduction is correct.

[15] 4. $GraphHomo \in$ NP-hard

Prove that the graph homomorphism problem is NP-hard by reducing the graph k-coloring problem to it.

a. Give a polynomial-time many-one reduction from the graph k-coloring problem to the graph homomorphism problem.

Hint: Given a graph G and integer k you have to construct two graphs such that G has a k-coloring if and only if there is a homomorphism between your constructed graphs.

b. Briefly argue that your reduction is correct.

[15] 5. $GraphHomo$ **and** SAT

From the previous questions we know that the graph homomorphism problem is NP. Since SAT is in NP-hard there is a polynomial-time many-one reduction from the graph homomorphism problem to SAT. In this question you are going to give a *direct* reduction explicitly (i.e. without using the fact that SAT is NP-hard).

a. Give a polynomial-time many-one reduction from the graph homomorphism problem to SAT.

Hint: Given two graphs G and H you have to defined a propositional formula φ such that φ is satisfiable if and only if there is a homomorphism from G to H. Think about what variables you are going to have in your propositional formula.

b. Briefly argue that your reduction is correct.