Hsiang (Sean) Lo

CS 325 - Homework 7

1. ***(6 points)*** **Let X and Y be two decision problems. Suppose we know that X reduces to Y in polynomial time. Which of the following can we infer? Explain**

a. If Y is NP-complete then so is X.

Reducibility only satisfies condition 1, therefore the most a function can be reduced to is NP-Hard at this stage, so no.

b. If X is NP-complete then so is Y.

Because X reduces to Y, this would be true if Y is NP-Hard, but because it's not, its not true.

c. If Y is NP-complete and X is in NP then X is NP-complete.

This is also not true because NP-Complete is both NP-Hard and NP.You can also ever deduce to NP-Hard using reducibility.

d. If X is NP-complete and Y is in NP then Y is NP-complete.

NO it is not possible for Y to be both NP and NP-Complete since X reduces to Y. The most it can be proved is that its NP-Hard

e. If X is in P, then Y is in P.

Because X reduces to Y, therefore Y is also P, true.

f. If Y is in P, then X is in P.

Because X reduces to Y, we can only infer what Y becomes. Not what X is.

2. ***(6 points)*** **A Hamiltonian path in a graph is a simple path that visits every vertex exactly once. Show that HAM-PATH = { (G, u, v): there is a Hamiltonian path from u to v in G } is NP-complete. You may use the fact that HAM-CYCLE is NP-complete.**

Given that HAM-Cycle is NP-Complete, and we want to use reducibility to prove that Hamiltonian path is NP-Complete as well.

1) Show that Hamiltonian Path is a subset of NP to prove NP-Hard

Suppose we have a digraph G = (V,E), Construct a new graph G' = (V',E') as follows: Pick an arbitrary vertex v in V and split it into two vertices: Vo and V1. Let V' = (V-v) U {Vo,V1}. For all edges (v,u) subset of E, add an edge (vo, u) is a subset of E'. For all other edges in E,c opy them to E'. We claim that G'' has a Hamiltonian path from V0 to V1 if and only if G has a Hamilton cycle.

2) A ≤P B for some NP-complete problem A.

For each edge {u, v} create a new graph by deleting this edge and adding vertex x onto u and vertex y onto v. Let the resulting graph be called G'. Invoke Hamiltonian Path subroutine to see whether G' has a Hamiltonian path. If it does, then it must start at x to u, and end with v to y (or vice versa). Then we know that the original graph had a Hamiltonian cycle (starting at u and ending at y). If this fails for all edges, then we report that the original graph has no Hamiltonian cycle.

3. ***(5 points)*** **LONG-PATH is the problem of, given (G, u, v, k) where G is a graph, u and v vertices and k an integer, determining if there is a simple path in G from u to v of length at least k. Show that LONG- PATH is NP-complete.**

Long Path is in NP since the path is the certificate (we can easily check in polynomial time that it is a path, and that its length is k or more. Traversing through the list, since a Long-Path is a simple path with the longest possible length, it is then possible to traverse from vertices u to vertices v of length at least k. Since this possible to be done in polynomial time, it is at least NP-Hard. (Further explanation below)

Now we need to prove NP-Complete such that a variety of X <= p Long-Path for R is a subset of NP-Complete. Since Hamiltonian Path (the variant where we specify a start and end node) is a special case of Long Path, namely where k equals the number of vertices of G minus 1. X is a Ham-Path and also a known NP-Complete. And to transform Ham-Path to a Long-Path, provided that we have a graph G such that has a Hamilton path if and only if its longest path has length of n -1.

This reduction shows that the decision version of the Long-Path is also NP-Complete. Now suppose you add a weight to each of the vertex, provided G has a Ham-Path if and only if G’ has a long path, we can see that with or without the weight, long-path result is the same as the ham-result, therefore, the Long-Path is NP-complete as well.

4. ***(8 points)*** **K-COLOR. Given a graph G = (V,E), a k-coloring is a function c: V -> {1, 2, ... , k} such that c(u) □ c(v) for every edge (u,v) □ E. In other words the number 1, 2, .., k represent the k colors and adjacent vertices must have different colors. The decision problems K-COLOR asks if a graph can be colored with at most K colors.**

g. The2-COLORdecisionproblemisinP. Describeanefficientalgorithmtodetermineifagraph has a 2-coloring. What is the running time of your algorithm?

```
Check Color(Graph G){

    While not the last vertex{

        For(All vertexes connected to current)

            If Current vertex -> V == vertex.adjacet

                Return false
```

```
Else

      Return true

This can be done in linear time or O(n)
```

h. The 3-COLOR decision problem is NP-complete by using a reduction from SAT. Use the fact that 3-COLOR is NP-complete to prove that 4-COLOR is NP-complete.

To reduce from 3-Color to 4-Color, we find that there is a reduction function f that takes input a graph G=(V,E) and produce another graph G'=(V',E') such that it introduces a vertex w that is new and not in V.

Let that V' = V U {w} and E' = E U {{v,w}:v is a subset of V}

$V \rightarrow \{1,2,3\}$ is 3-coloring of G

$V' \rightarrow \{1,2,3,4\}$ is a 4-coloring of G'.

Assuming assigned w is 4. So removal of 4 wont affect V, therefore it shows a 4-color graph is also a 3 color graph