Benjamin Fondell

HW 7

1.

a. X may be complete but we cannot assume that it is just based on this property.

b. This cannot be assumed either. It is possible for Y to be NP-hard but not NP-complete

c. We cannot assume this either. X may be in NP and reduce from Y and not be NP complete.

d. This can be inferred by the properties of NP complete if x is np complete and y is in np and and x reduces o y then y is NP-Complete

e. Yes they can, one NP-Complete problem may reduce to another, vice versa. therefore we cannot infer this.

f. This cannot be inferred either. X may be in p but may also reduce from an np-problem.

g. This can be inferred. If x reduces to y and y is in p then x must also be in p.

2.

a.   This does not follow from COMPOSITE is in NP and SUBSET_SUM is in NP-Complete. An NP-Complete algorithm does not reduce to an NP problem unless that problem is also NP-Complete or NP-Hard.

b.   O(n^3) is a polynomial time algorithm for SUBSET_SUM. Because SUBSET_SUM is NP-Complete, all problems in np reduce to SUBSET_SUM. This implies that there   exists a polynomial time algorithm for all problems in NP and because COMPOSITE is in NP then the claim follows.

c. This would be true if COMPOSITE was NP-Complete but we have for this fact that COMPOSITE is NP therefore we cannot assume that P = NP.

d. This does not follow. Because P is in the set NP and every problem in P may be solved by a polynomial time algorithm. The implication of this statement is that no NP-Complete algorithm can be solved in polynomial time because the sets P and NP do not intersect.

3.

a. true. Because both problems are NP-Complete and because NP-Complete requires that all NP problems reduce to NP-Complete problems then an NP-Complete problem must also reduce to another NP-Complete problem therefore 3-SAT ≤p TSP .

b. assume this is true that is p!= NP. Then if there if 3-SAT $\leq_p$ 2-SAT then 2-sat out be NP-Complete. But 2-SAT is also polynomial then NP ⊆ P and P ⊆ NP so, P = NP which is a contradiction to the statement so it is false.

c. This is true as because all np problems can reduce to an NP-Complete problem therefore the two sets NP and NP-Complete intersect but are not equal. Also P is a subset of NP though it is not the same set as NP and thus never intersects with NP-Complete by these observations no NP-Complete problem can be solved in P time.

4.

Hamiltonian Path: A simple open path that contains each vertex in a graph exactly once.

Show that HAM-PATH belongs to NP:

For a graph G{E,V} we are given a set of nondeterministically chosen set of edges E of G that are included in the path. Then traversing the adjacency list we can make sure we visit each vertex exactly one time. This traversal would require polynomial time therefore we can assume HAM-PATH is NP.

Show that HAM-CYCLE (NP-Complete problem) can reduce to HAM-PATH:

- Again, given a graph G we must find a hamiltonian cycle for at minimum one edge e = {x,y}. Create a copy of graph G(prime) adding vertices x(prime) and y(prime) such that x(prime) is connected by an edge only to x and y(prime) is connected by an edge only to y.
- G(prime) has a hamiltonian path if and only if G has a hamiltonian cycle with edge {x,y}
- Run the HAM-PATH algorithm for each copy graph G(prime) at each edge in G. If all the copy graphs have no hamiltonian path then G has no hamiltonian cycle.

We've shown that HAM-PATH belongs to NP and that an NP-Complete algorithm can reduce to HAM-PATH therefore HAM-PATH is NP-Complete.

5.

Prove that LONG-PATH is in NP-Complete

1) Show that LONG-PATH is in NP.

An example long path algorithm might traverse all vertexes of a graph. At each vertex it check if {v(1), v(2), v(3), …} are reachable. If true then sum the current edge weight to the path length. If not continue to the next permutation. If the path length to a reachable vertices is greater than the maximum path length then store the path of vertices and longest path value to some variable. This algorithm would have the runtime O(n * n^n). As this runtime is polynomial we can determine that the LONG-PATH problem is in NP.

2) Show that R ≤P LONG-PATH for some R is inNP-Complete

a. Consider the HAM-PATH problem for R

Benjamin Fondell

b. Given an instance $G' = \{V', E'\}$ for HAM-PATH, count the $|V'|$ of vertices in G' and pass the graph such that $G = G'$, $K = |V'|$ for LONG-PATH. Then G' has a simple path of length $|V'|$ if and only if G' has a Hamiltonian path. It is implicit in a long path that the path is simple meaning it does not use the same vertices twice.

c. Prove that R(x) = yes if and only if Q(x') = yes. That is

   a. If x is a "yes" solution of R then x' is a "yes" solution of Q. And

      Consider the contrapositive that is, consider that no sub-graph $G' = \{V', E'\}$ is found in HAM-PATH for graph G to satisfy the problem. Then there exists no hamiltonian path in G for any subgraph $G' = \{V', E'\}$. Hence there exists no simple path and therefore will not satisfy LONG-PATH. By the contrapositive then, if x is a solution for HAM-PATH then x is solution for LONG-PATH.

   b. If x' is a "yes" solution of Q then x is a "yes" solution of R.

      Consider a graph $G = \{V, E\}$ for LONG-PATH. If there exists a solution for long path then a sub-graph $G' = \{V', E'\}$ with $\{V', E'\} = \{V, E\}$ of the LONG-PATH, will also satisfy the HAM-PATH problem. This is because a longest path is inherently a path without repeat vertices. Therefore if x is a solution for LONG_PATH then x is a solution for HAM-PATH.

Now, because both 1 and 2 are true that is LONG-PATH is in NP and LONG-PATH and an NP-Complete algorithm might reduce to LONG-PATH then LONG-PATH is NP-Complete which was to be shown.