Zuhair Ahmed
CS325_400
Spring 2020

Homework #7

1. (7 pts) 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
False. To prove for NP complete for X, we need to show that X has to be the subset of NP complete which is unknown.

b. If X is NP-complete then so is Y
False. Just because X is reducible to Y, but this doesn't mean Y is reducible to X. So X may or may not be NP-complete.

c. If Y is NP-complete and X is in NP then X is NP-complete
False. Since X is reducible to Y, this means X can be solved in polynomial time by Y, but the converse may or may not be true.

d. If X is NP-complete and Y is in NP then Y is NP-complete
True. Since X reduces to Y so there is a means to solve Y in polynomial time, then that can be used to solve the problem X in polynomial time too. Hence Y is both in NP and NP Hard implying that Y is also NP-complete.

e. X and Y can't both be NP-complete
False. X can be NP-complete if an only if Y is NP-complete

f. If X is in P, then Y is in P
False. Just because X is reducible to Y, doesn't mean the converse so Y may or may not be in P

g. If Y is in P, then X is in P
True. Since X reduces to Y so there is a means to solve Y in polynomial time, which then that can be used to solve the problem X also in polynomial time

Hence, we can infer D and G to be true.

2. (4 pts) Consider the problem COMPOSITE: given an integer y, does y have any factors other than one and itself? For this exercise, you may assume that COMPOSITE is in NP, and you will be comparing it to the well-known NP-complete problem SUBSET-SUM: given a set S of n integers and an integer target t, is there a subset of S whose sum is exactly t? Clearly explain whether or not each of the following statements follows from that fact that COMPOSITE is in NP and SUBSET-SUM is NP-complete:

a. SUBSET-SUM ≤p COMPOSITE
False. Since the SUBSET-SUM is NP-Complete, it can be reduced to any other NP-Complete problem. However this does not carry over to broader NP class that COMPOSITE is part of. Hence SUBSET-SUM (NP-Complete) may or may not reduce down to COMPOSITE (NP).

b. If there is an $O(n^3)$ algorithm for SUBSET-SUM, then there is a polynomial time algorithm for COMPOSITE

True. If there is a polynomial time algorithm for SUBSET-SUM (NP-Complete) then this mean that there is a polynomial algorithm for all NP-Complete problems. This would also prove that P = NP which implies that a polynomial time algorithm must also exist for COMPOSITE (NP).

c. If there is a polynomial algorithm for COMPOSITE, then P = NP

False. COMPOSITE (NP) having a polynomial time algorithm does nothing to show that P = NP. This is because COMPOSITE could be NP Easy or NP-Composite or NP-Hard or some other subset of NP. So just because we can reduce one specific algorithm called COMPOSIE down to polynomial does not prove either P = NP or P ≠ NP.

d. If P ≠ NP, then no problem in NP can be solved in polynomial time

False. By definition P is a subset of NP and if P ≠ NP were true, then there can still be plenty of algorithms in P that are solvable in polynomial times. At the same time this does nothing to show that NP-Complete problems, like SUBSET-SUM, can still only be solvable in exponential time and not faster.

3. (3 pts) Two well-known NP-complete problems are 3-SAT and TSP, the traveling salesman problem. The 2-SAT problem is a SAT variant in which each clause contains at most two literals. 2-SAT is known to have a polynomial-time algorithm. Is each of the following statements true or false? Justify your answer.

a. 3-SAT ≤p TSP

True. This goes back to showing that if one NP-Complete problem can be solved in polynomial time, then all NP problems can be solved in polynomial time. In this example, since 3-SAT is NP-Complete and Traveling Salesman Problem (TSP) is also NP Problem therefore if it could be showed that 3-SAT could be solved in polynomial time then TSP can also be solvable in polynomial time.

b. If P ≠ NP, then 3-SAT ≤p 2-SAT

False**.** If *P≠NP* and there is a polynomial-time reduction from 3-SAT to 2-SAT. This implies that 2-SAT is NP-complete, since it is already been proved that 3-SAT is indeed NP-Complete. If this were true we can reduce all NP-Complete problems down to NP meaning that *P=NP.* This is of course a contradiction with original statement.

c. If P ≠ NP, then no NP-complete problem can be solved in polynomial time

True. This is because if one NP-Complete problem can be solved in polynomial time, then ALL NP-Complete problems can be solved in polynomial time. This also then means that P = NP. However, if we assume P ≠ NP as stated, then by definition no NP-complete problem can be solved in polynomial time.

4. (6 pts) 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

By definition, a Hamiltonian path is a simple and open path that contains each vertex in a graph only once. To demonstrate that Hamiltonian path problem is NP-complete we first need to 1) prove that it is in NP and 2) then show that it is in NP-Complete. The latter can be shown by finding a known NP-complete algorithm that then can be reduced to Hamiltonian Path.

For a given graph G, we can show that a Hamiltonian Path exists by choosing arbitrary edges from G that are to be included in the path. Then we traverse the path and make sure that we visit each vertex exactly/only once. This can be done in polynomial time or as shown in this example O(N) time, so the problem belongs to NP. Hence first part of proof is complete.

To show NP-completeness, this problem can be reframed to simplify. A similar problem is whether a graph contains a Hamiltonian cycle or more precisely a Hamiltonian path that begin and end in the same vertex. In addition, it well known that the Hamiltonian Cycle problem belongs to NP-complete class. So if we can show that a Hamilton Cycle can reduce down to Hamiltonian Path then therefore the Hamiltonian Path must also belong to NP-Complete. Put another away, if HAM-PATH G contains a Hamiltonian Cycle if and only if G contains a Hamiltonian Path.

Firstly, suppose that G contains a Hamiltonian <u>cycle</u>. This must mean that we visit each vertex exactly once but with added restraint that this is also starts and ends in same vertex. This later is an added constraint, so if first part is true then if we assume G contains Hamiltonian Cycle then it must contain a Hamiltonian Path.

Secondly, suppose a different G contains a Hamiltonian <u>path</u>. In that case, we by definition have a path that visits each vertex exactly once. In order to make this graph into a cycle all we need to do is remove (or add) a vertex such that we can redirect path back to starting position. This will then classify this particular but arbitrary Hamiltonian path as a Hamiltonian cycle.

Hence, we have shown that HAM-PATH G contains a Hamiltonian Cycle if and only if G contains a Hamiltonian Path which then satisfies second condition of proof.

In conclusion, we have shown that G is in NP and that G is NP-Complete. This demonstrates that Hamiltonian Path is NP-Complete.

5. (5 pts) 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. Prove that LONGPATH is NP-complete.

Same as with problem 4, to show that an algorithm is NP-Complete we must show that 1) it is in NP and 2) that it can be reduced to another NP-Complete problem.

We can clearly see that LONG-PATH is in NP as by definition it is a simple path. This means that it can traverse nodes in graph from points u to v at worst case in O(N) time which is in NP. To show that it is part of NP-Complete we just need to reduce it down a Hamilton Path (which as shown in problem 4) has already been proved NP-Complete. If we assume vertex u is starting point of graph G and v is ending point (i.e. G-1 number of points total) such that all points in graph G are traversed in path then G is indeed a Hamiltonian path. This implies that a Hamilton Path is a subset/special case of LONG-PATH. Hence, since Hamilton Path is NP-Complete, LONG-PATH is also NP-Complete.