CS325-HW5

1. **Let X and Y be two decision problems. Suppose we know that X reduces to Y in polynomial time. State whether the following statements are true or false and give a brief explanation.**

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

   False. From the question, we get X reduces to Y in polynomial time, which mean is X is not harder than NP-complete problem. X could be NP or other. We cannot verify that X is NP-complete too.

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

   False. Because X reduces to Y in polynomial time, Y could be NP-hard. We cannot verify that Y is NP-complete.

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

   False. similar question with a). If Y is NP-complete, there is a polynomial time reduction from every problem A $\in$ NP to Y. We cannot verify that x is NP-complete.

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

   True. We know X $\leq_P$ Y and X is NP-complete. Y $\in$ NP. We can verify that Y is NP-complete.

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

   False. We know X $\leq_P$ Y => X is "no harder to solver" than B. So, Y could be in P or NP. We cannot sure Y is in P.

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

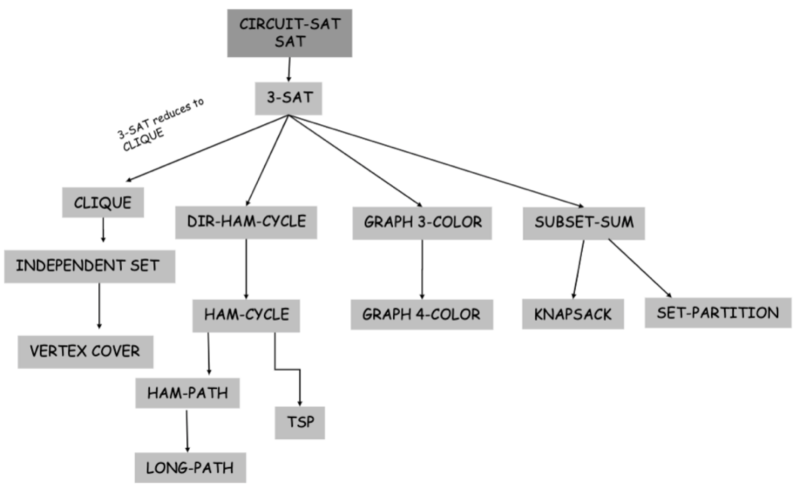
True. We know X $\leq_P$ Y => X is "no harder to solver" than B**.** Hence, if X is in P, Y is in P too.

g) **X and Y can't both be in NP.**

False. In the question d), that statement is true. So, we can proof that X and Y can both be in NP.

2. **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. State whether the following statements are true or false and give a brief explanation.**

a) **3-SAT $\leq_P$ TSP.**



True. According to the NP-Completeness tree graph. We can see the TSP is the subtree of 3-SAT, which mean is that 3-SAT can reduces to TSP in polynomial time.

b) **If P $\neq$ NP, then 3-SAT $\leq_P$ 2-SAT.**

False. From this question, we know that 3-SAT is NP-Complete, and 2-SAT is known to have a polynomial-time algorithm. So, 2-SAT should be in NP. Hence, 3-SAT cannot reduce to 2-SAT in polynomial time.

c) **If TSP $\leq_P$ 2-SAT, then P=NP.**

True. TSP is in NP-Complete, and 2-SAT is in P. This statement mean is that TSP can reduce to 2-SAT in polynomial time. We get NP-Complete can be solved in polynomial time. Hence, all NP be solved in polynomial time. Then we can get P = NP.

3. **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.**

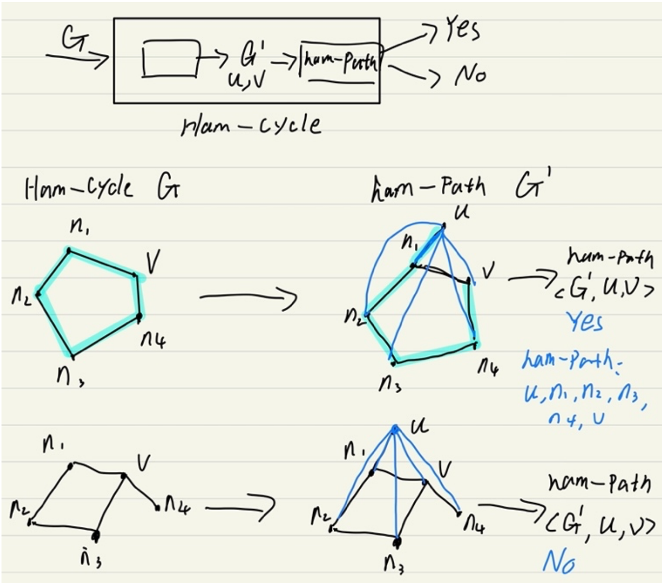
Ham-Path is the problem of, given a directed or undirected graph with n vertices {(G, u, v)}, find a path which from u to v and connects every vertex exactly once.

a) Show that HAM-PATH $\in$ NP:

Given a graph G = (V, E) with n vertices which start from u, end at v and a "certificate solution" HAM-PATH. PATH = (u, … v). we can verify in polynomial time that if path has all of vertices in the G. Also, we need check if each point appears exactly once, in this step, we can just determine that if the number of vertices in PATH equal to the number of vertices in the G. This takes time at most O (V + 1) = O(V) which is polynomial time.

b) Show that HAM-CYCLE $\leq_P$ HAM-PATH:

i. Given a graph G we produce a new graph G' such that G has a ham-cycle if and only if G' has an HAM-PATH. G' is created by adding one new vertices u. And connect it to every vertex in G. This transformation of G into G' can be done in polymodal time by adding one vertex and some edges.



ii. Prove the HAM-CYCLE can be solved by HAM-PATH:

Show that the graph G has HAM-CYCLE if and only if graph G' has an HAM-PATH.

1. If G has a HAM-CYCLE<v1, v2… vn>, G' has a HAM-PATH: <u, v1, … vn>
2. If G' has a HAM-PATH, it has a path through all vertices exactly once. This path will have a start point u, end point v. we just need check if there have a line between u and v or just adding an edge between u and v. Then we will get the HAM-CYCLE.

iii. Since HAM-CYCLE is in NP-Complete then HAM-PATH is in NP-Hard

Acceding to the a) and b), we proved HAM-PATH is in NP and NP-Hard. HAM-PATH is NP-Complete

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

   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.

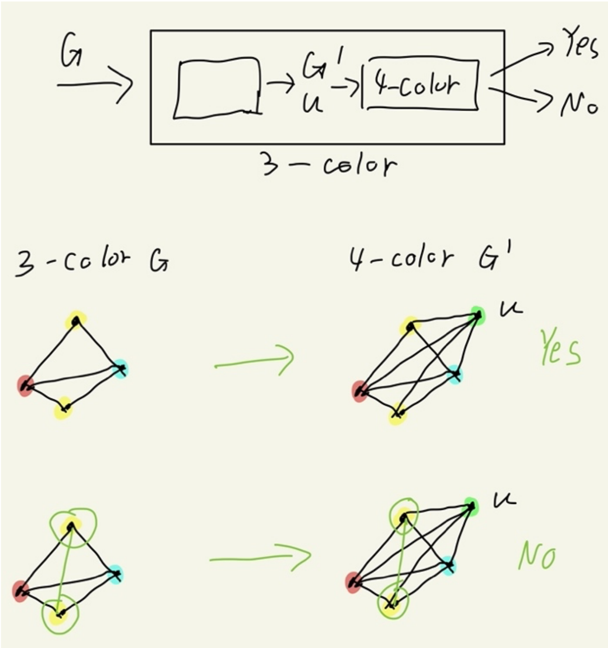
   4-COLOR is the problem of, given an undirected graph with n vertices. Assignment some colors to every vertex of the graph such that no two adjacent vertices have the same color, and at most 4 colors are used to complete color the graph.

   a) Show that 4-COLOR $\in$ NP:

      Given a graph G = (V, E) that every node is assigned by 4 different color and a "certificate solution" 4-COLOR. For each edge {u, v} in graph G verify that the color c(u)≠c(v). In this way, we can check if this solution is correct. This takes time at most O(V+E) which is polynomial time.

   b) Show that 3-COLOR $\leq_P$ 4-COLOR:

      i. Given a graph G we produce a new graph G' such that G has a 3-COLOR if and only if G' has a 4-COLOR. G' is created by adding a new vertex u and connecting u to all verities in the graph. This transformation of G into G’ can be done in polynomial time by adding one vertex and some edges.



ii. Prove 3-COLOR can be solved by using 4-COLOR, and 4-COLOR is harder than 3-COLOR.

Show that the graph G has a 3-COLOR if and only if graph G’ has a 4-COLOR.

1. If G has a 3-COLOR cr = {c1, c2, c3}and for each edge {u, v} c(u)! = c(v). G’ has a 4-COLOR cr= {c1, c2, c3, c4} and each edge {u, v} c(u)! = c(v).
2. If G’ has a 4-COLOR, then there will have a vertex u which has a unique color. Also, this G’ is meet for each edge {u, v} c(u)! = c(v). Hence, we can remove a vertex u and some edges, and it will produce a 3-COLOR in G.

iii. Since 3-COLOR is in NP-Complete then 3-COLOR is in NP-Hard

Acceding to the a) and b), we proved 4-COLOR is in NP and NP-Hard. Hence, 4-COLOR is NP-Complete.