Aashish Yadavally
Spring 2021

Homework 5

Question 1

Let F be a disjunctive normal form (DNF). Let P_i be an elementary product of F.

The following is an algorithm to decide the boolean satisfiability of a DNF: Loop through all the elementary products P_i of F. Let F_0 and F_1 be the remainder formulae when $P_i = 0$ or when $P_i = 1$. For each P_i in F, if at least one of F_0 and F_1 are satisfiable, F is satisfiable. This algorithm has polynomial time complexity. Thus this problem is polynomial-time solvable.

Question 2

Given two undirected graphs G_1 and G_2, the SUBGRAPH-ISOMORPHISM problem checks if G_1 is a isomorphic to a subgraph of G_2. Two graphs are considered to be isomorphic if there is a bijection between the vertex sets of both the graphs, thus retaining the edge connectivity. A problem is NP-complete if it is both in NP and in NP-Hard.

Case I (SUBGRAPH-ISOMORPHISM problem is in NP):

Let the subgraph of G_2 be G, and M be the mapping of the vertices between G_1 and G. If we verify that the mapping M is a bijection, and if for every edge in G_1, there is a corresponding edge in G, then the subgraph-isomorphism condition is satisfied. Verifying each of these conditions takes polynomial time. Thus, the subgraph-isomorphism problem has polynomial-time verifiability. Hence, it is in NP.

Case II (SUBGRAPH-ISOMORPHISM problem is in NP-Hard):

Consider the Clique-Decision (C) problem which checks whether a graph G contains a clique of size k. We know that it is in NP-Complete. In order to reduce it to Subgraph-Isomorphism (S) problem, let us assume that the input graph G_1 in S is a complete graph of k vertices. Also, let G_2 in S correspond to G in C. When k is greater than the number of vertices in G, a clique of size k cannot be a subgraph of G. Thus, $k \leq n$. Since G_1 is a subgraph of G_2 and every graph is isomorphic to itself, S is true. Also, G has a clique of size k, if and only if G1 is a subgraph of G2. Hence, if C is true, G is true as well. Thus, the clique decision problem can be reduced to the subgraph-isomorphism problem in polynomial time for a particular instance. This shows that an NP-Complete problem can be reduced to subgraph-isomorphism problem in polynomial time. Hence, it is NP-Hard.

Therefore, SUBGRAPH-ISOMORPHISM problem is NP-Complete.

Question 3

The decision version of the longest simple-cycle problem (S) can be stated as follows: Given a graph G and a positive integer k, determine whether G has a cycle with length at least k.

Consider the Hamiltonian Cycle (H) problem. For an input graph $G = (V, E)$ in H, set

$k = |V|$ where k is the minimum length of the cycle in S. If a graph G contains a Hamiltonian cycle, then this is a simple cycle with $|V|$ vertices. If there is no Hamiltonian cycle, the length of the longest simple cycle is 0. Thus, G has a Hamiltonian cycle if and only if G has a simple cycle with the length at leaves. This reduction only involves a simple counting and is polynomial-time solvable. Thus, H can be reduced to S in polynomial time. Hence, the longest simple-cycle problem is in NP-Hard.

Question 4

Given a graph $G = (V, E)$, the Hamiltonian Path (H_P) problem determines if the graph contains a path containing all vertices $v \in V$ exactly once. A problem is NP-Complete if it is both in NP and NP-Hard.

Case I (H_P is in NP):

Consider a random ordering of vertices and verify whether it is a Hamiltonian Path by traversing through the set of vertices in the ordering. One such ordering of vertices where it is true can be thought of as a certificate. Additionally, it can be seen that the verification for Hamiltonian path can be done in polynomial time. This gives us a non-deterministic polynomial-time algorithm for the Hamiltonian-Path problem. Therefore, H_P is in NP.

Case II (H_P is in NP-Hard):

Consider the Hamiltonian Cycle (H_C) problem. For an input graph $G = (V, E)$, H_C checks if G has a cycle which contains a Hamiltonian Path. For an arbitrary vertex $u \in V$, add a copy of it with all it's edges. Add two vertices v and v' and connect v to u and v' to u'. Such a construction ensures that the graph constructed G' has a Hamiltonian Path if and only if G has a Hamiltonian Cycle. Conversely, suppose G' contains a Hamiltonian path. In that case, the path must necessarily have endpoints in v and v'. This path can be transformed to a cycle in G by disregarding the vertices v and v', and closing the path back to u instead of u'. Hence we have shown that G contains a Hamiltonian cycle if and only if G' contains a Hamiltonian path. Thus, H_C has been reduced to H_P. Hence, H_P is NP-Hard.

Since H_P is both in NP and NP-Hard, therefore, it is NP-Complete.

Question 5

The decision version of the Spanning-Tree (S) problem can be stated as follows: Given a graph $G = (V, E)$ and a positive integer k, determine whether G has a spanning tree with at least k leaves.

Consider the Vertex-Cover V_C problem which for an input graph $G = (V, E)$, V_C finds a minimum set of vertices that covers all the edges in the graph.

Let $E' = \{(u, e), (v, e) | e = (u, v) \in E'\} \cup \{(s, t)\} \cup \{(s, u) | u \in V\}$.

$V' = V \cup E' \cup \{s, t\}$ where $s, t \notin V \cup E'$.

Construct a graph $G' =< (V', E'), k >$ where $k = 1 + |E'| + |V| - h$. A Hamiltonian path exists in G' if and only if the minimum-leaf spanning tree has exactly two leaves. The problem of determining whether a Hamiltonian path exists in a graph is an NP-Hard problem. Thus, we have reduced H_P to S in polynomial time. Hence, S is NP-Hard.

Question 6

Half-Clique (H_C) problem deals with determining whether a graph G with n vertices has a Clique of size at least $\frac{n}{2}$. A problem is NP-Complete if it is both in NP and NP-Hard.

Case I (H_C is in NP):

Consider a set of $k = \frac{|V|}{2}$ vertices to test if they are fully connected. One such set of vertices for which they are fully connected is the certificate, verification of which can be done in $O(|V|^2)$. Since it is polynomial-time solvable, H_C is in NP.

Case II (H_C is in NP-Hard):

A graph has a clique of size $\frac{n}{2}$ if and only if it's complement has an independent set of size $\frac{n}{2}$, which in turn is possible only if a vertex cover of size at most $\frac{n}{2}$ is feasible. Thus, H_C can be reduced from the Vertex-Cover (V_C problem.

Now, consider the reduction from 3SAT (S_3) problem to V_C.

For each 3CNF F with n variables and m clauses. Construct a graph G which contains $2n + 3m$ vertices. F is satisfiable only if G has vertex-cover of size $n + 2m$. By choosing a node for each of the literals, we cover all the edges between them, and subsequently satisfy the 3SAT formula.

If we assume a vertex cover of G with $n + 2m$ vertices or less, at least n vertices are covering the literal pairs since the edges between them can't be covered in any other way. The other $2m$ vertices must be covering the traingles since each triangle requires two vertices to be covered. This assignment has to satisfy the formula since every clause is satisfied by our construction. This construction takes as input the m clauses and n literals and creates a graph with exactly $2n + 3m$ nodes. This can be done in polynomial time. Thus reduction from V_C to S_3 can be done in polynomial time, and that from H_C to V_C can be done in polynomial time. Therefore, H_C is NP-Hard.

Since H_C is both in NP and in NP-Hard, it is NP-Complete.

Question 7

A dominating set is a subset of vertices in which every vertex not in the subset is adjacent to a vertex in the subset. Consider an input graph $G = (V, E)$. To build an instance of dominating set (H, k) for G, remove all the isolated vertices in V, and for every edge $(u, v) \in E$, add an additional vertex x connected to u and v.

The dominating set of G obtained by removing the isolated vertices, it can be observe that the new vertices are also dominated. Thus the vertex cover of G is a dominating set of H. Thus, if G has a smaller k, then H has a dominating set smaller than k. Additionally, everything that x connected to the edge (u, v) dominates, is also dominated by u and v. So we can assume that the dominating set D contains only vertices from G. Now for every edge (u, v) in G the new vertex x is dominated by D. Consequently, either u or v is also in D. Thus, D is the vertex cover of G. Therefore, G has a vertex cover smaller than k if and only if H has a dominating set smaller than k. This reduction from the vertex cover problem to the dominating set problem can be done in polynomial time. Thus, the dominating set problem is in NP-Hard.