CSCE 629-601 Homework 11
28th November, 2016

Name: Adil Hamid Malla **UIN:** 425008306

Problem 1. Textbook page 110, Exercise 34-5.1.

Question: Sub Graph Isomorphism Problem - Given two graphs G and H, check whether the H is isomorphic to subgraph of G. Show that this decision problem is NP Complete.
Solution:
Since the question is a decision problem asking whether the graph H is isomorphic to the subgraph of G. To prove that this problem, which we will use as sub graph isomorphic problem no one is NP complete.
First we must know that what the problem is. Lets us assume that we have two graphs, $G = (V, E)$ and $H = (V', E')$. Now we have to check whether there exists a sub graph of G as $G_0 = (V_0, E_0)$ such that

$$V_0 \in V$$
$$E_0 \in E \text{ and } E_0 \cap (V_0 \times V_0)$$

Such that $G_0 \cong H$, i.e there exists an $f : V_0 \to V'$ such that $(u, v) \in E_0 \Leftrightarrow (f(u), f(v)) \in E'$.
To show that it is NP complete we will follow three steps:

1. **We need to prove that sub graph isomorphism is NP**.
To prove that we need a certificate, and we should be able to say whether the given certificate gets a yes or no as our problem is a decision problem. To see that the problem is in NP, we observe that a certificate is a mapping ϕ from the nodes of G_0 (a subset of G) to the nodes of H, describing which vertices of H correspond to vertices of G_0. The certifier then needs to make sure that for each edge e = (u, v) in G_0, the edge (ϕ(u), ϕ(v)) is also in H, and whenever (u, v) is not an edge of G_0, then (ϕ(u), ϕ(v)) is not an edge of H. This can be done with two simple nested loops, and takes at most $O(n^2)$ time, i.e., polynomial. So this proves that we can say whether a given certificate is solution of our problem or not.

2. **We need to have a mapping from this problem to any known NP Complete problem.** The easiest problem we can think of is Reduction from the Clique Problem. Since we have already proved in the class that the Clique problem is NP Complete, then if we prove that we can reduce Clique problem to our sub graph isomorphism problem in polynomial time then we can prove the second step of showing that the given problem is NP complete. We can show that, for instance, the Clique problem is a special case. Given an instance of Clique, consisting of a graph G_{new} and a number k, we generate an instance of Subgraph Isomorphism by setting $G = G_{new}$, and H a clique on k vertices. This reduction clearly takes polynomial time, since all we do is copy a graph, and write down a complete graph in time $O(k^2)$. So thus the reduction of the given problem takes place in polynomial time.

3. **We need to prove that the reduction/mapping to Clique problem preserves the Yes and No answer.** We can prove that by getting a yes from both the sides. To prove correctness of the reduction, first assume that G contains a clique of size k. Then, those k nodes of G = G_0 are isomorphic to H, because H is a clique of size k. So this is equivalent to having the same number of edges in H as well as G_0 which is k-subset of G and also the correspondence is maintained. Conversely, if G contains a subgraph G_0 isomorphic to H, because H is a k-clique, G must contain a k-clique. Thus, we have the **Yes** preservance from both the sides.

Hence the sub graph isomorphism is a NP Complete Problem.

Problem 2. Textbook page 110, Exercise 34-5.2.

Given, a $m * n$ matrix A and a m- vector b, we have to prove that checking whether there exists an integer n-vector x with elements in the set $\{0, 1\}$ such that $Ax \leq b$ is an NP Complete problem.
If we can reduce this problem to one of the other NP Complete problem , then we according to the property of NPC problems, we have proved that $0 - 1$ Integer programming problem is NP Complete. A reduction from 3-SAT finishes the proof.

- Given an integer vector x with elements from the set $\{0, 1\}$ as a certificate, we can evaluate every constraint in above problem by substituting the values of x to verify if it satisfies all constraints. This can be done in $O(mn)$ time, which is polynomial time algorithm. Hence, IP $\in$ NP.

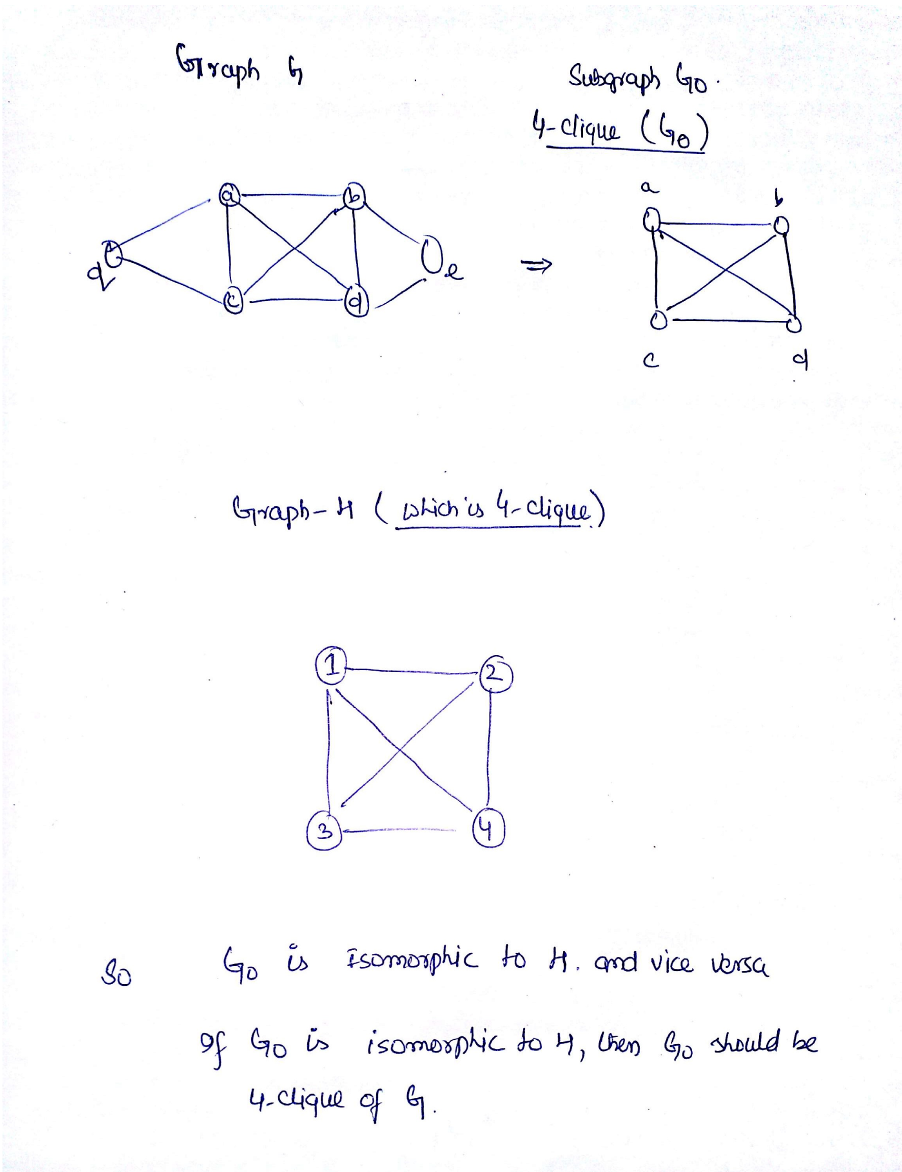

Figure 1: Example of the Sub Graph Isomorphism

- To show: Now we show that 3-CNF-SAT $\leq_p$ IP.

  Consider a 3-CNF formula ϕ with n variables $x_1, x_2, x_3, \ldots x_n$ and m clauses $C_1, C_2 \ldots C_m$.

  For example, consider the clause

$$\phi = (x_1 \vee x_2 \vee \bar{x}_4) \ \wedge \ (\bar{x}_2 \vee \bar{x}_3 \vee x_4)$$

Let us define the mapping from the 3-CNF formula to IP problem. Every clause is converted into a constraint by replacing $\bar{x}_i$ with $1 - x_i$, $\vee$ by a $+$ operator, leaving x_i as it is, and the constraint to be ≥ 1. In the above example, we can convert the formula into three inequalities as follows:

$$\begin{aligned} x_1 + x_2 + (1 - x_4) \geq 1 &\implies -x_1 - x_2 + x_4 \leq 0 \\ (1 - x_2) + (1 - x_3) + x_4 \geq 1 &\implies +x_2 + x_3 - x_4 \leq 1 \end{aligned}$$

The above inequalities can be written in $Ax \leq b$ with $A = \begin{bmatrix} -1 & -1 & 0 & +1 \\ 0 & -1 & +1 & -1 \end{bmatrix}$ and $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Notice that the element $a_{i,j}$ of A matrix can be formed using the following rule:

$$a_{i,j} = \begin{cases} +1 & \text{if } x_j \in C_i \\ -1 & \text{if } \bar{x}_j \in C_i \\ 0 & \text{otherwise} \end{cases}$$

Preservance of Yes/No and Correctness:

1. This ϕ is satisfiable if and only if there exists a 0-1 assignment to variables $x_1, x_2, ..., x_n$ such that $Ax \leq b$. This reduction is straightforward to compute, requiring only linear time in the number of variables to translate each clause into an inequality, and polynomial time overall to do this for all m clauses. As for a clause i there exist at least one variable with the value 1 for the direct variable(x_j) or for the inverted variable($1 - x_j$) and since all the variables are non-negative the sum is always greater than 1, and hence satisfying its equivalent inequality corresponding to the clause C_i.
2. Given 0-1 Integer Programming solution, then each clause's inequality is satisfied, so the value of each clause is 1 and the original instance of 3-CNF-SAT problem is satisfied. Conversely, if we have 3-SAT-CNF problem has a satisfying assignment to its variable, then we have value of each clause to be 1 and this also satisfies corresponding inequalities - so the 0-1 Integer Programming instance has the same solution. Consider a solution to the IP problem. Notice that the inequalities in the original form (x_i or (1-x_i)) form, to satisfy each inequality (≥ 1) at least one of the variables must be 1. If one of the variables is one, in its equivalent 3-SAT form the clause output is 1 if at least one of the variables is 1 and hence the solution satisfies the 3-SAT problem if it satisfies the IP problem.