CSC 373 - Algorithm Design, Analysis, and Complexity **Summer 2014**

Lalla Mouatadid

Introduction to Complexity Theory: CLIQUE is NP-complete

In this lecture, we prove that the CLIQUE problem is NP-complete. A clique is a set of pairwise adjacent vertices; so what's the CLIQUE problem:

CLIQUE: Given a graph $G(V, E)$ and a positive integer k, return 1 if and only if there exists a set of vertices $S \subseteq V$ such that $|S| \geq k$ and for all $u, v \in S$ $(u, v) \in E$.

We'll prove the theorem below by first showing CLIQUE is in NP, then giving a Karp reduction from 3-SAT to CLIQUE. (3-SAT $\leq_P$ CLIQUE).

Theorem 1. *CLIQUE is NP-complete.*

Proof. 1. To show CLIQUE is in NP, our verifier takes a graph $G(V, E)$, k, and a set S and checks if $|S| \geq k$ then checks whether $(u, v) \in E$ for every $u, v \in S$. Thus the verification is done in $\mathcal{O}(n^2)$ time.

2. Next we need to show that CLIQUE is NP-hard; that is we need to show that CLIQUE is at least as hard any other problem in NP. To do so, we give a reduction **from** 3-SAT (which we've shown is NP-complete) **to** CLIQUE. Our goal is the following:

Given an instance ϕ of 3-SAT, we will produce a graph $G(V, E)$ and an integer k such that G has a clique of size at least k **if and only if** ϕ is satisfiable.

Let ϕ be a 3-SAT instance and $C_1, C_2, ..., C_m$ be the clauses of ϕ defined over the variables $\{x_1, x_2, ..., x_n\}$. What we need to do is construct an instance of CLIQUE (a graph) that would somehow capture the satisfiability of the clauses of ϕ.

We will represent every clause C_i as $C_i = \{z_{i1}, z_{i2}, ..., z_{it}\}$ where each z_{ij} represents a literal in C_i. Since ϕ is a 3-SAT instance, we know that $t \leq 3$.

We construct a graph $G(V, E)$ by adding t vertices for every clause $C_i = \{z_{i1}, z_{i2}, ..., z_{it}\}$. In total this takes $\mathcal{O}(t \cdot m) = \mathcal{O}(m)$ time since $t \leq 3$. Then for every pair of vertices v_{ab}, v_{cd} in G, we will add the edge (v_{ab}, v_{cd}) if and only if we satisfy two conditions:

$$a \neq c \tag{1}$$

$$z_{ab} \neq \neg z_{cd} \tag{2}$$

What do these two conditions mean? Well (1) implies that the literals z_{ab}, z_{cd} corresponding to the vertices v_{ab}, v_{cd} respectively belong to different clauses $C_a \neq C_c$ in ϕ. The second condition implies that both literals can be satisfied simultaneously. This step of the construction takes $\mathcal{O}(m^2)$ time. The final step is to determine the value of k; we will set k to be m, the number of clauses in ϕ.

Now I claim that ϕ is satisfiable if and only G as constructed above has a clique of size at least $k = m$.

First suppose ϕ is satisfiable. Then there exists a satisfying assignment $(x_1^*, x_2^*, ..., x_n^*)$ such that every clause C_i in ϕ is satisfied. Notice that to satisfy a clause C_i, we just need one of its literals in $\{z_{i1}, z_{i2}, ..., z_{it}\}$ to be satisfied. We iterate through the clauses and choose one satisfied literal from every clause which we denote by $(z_1^*, z_2^*, ..., z_m^*)$. Let $v_1, v_2, ..., v_m$ be the corresponding vertices in G to the satisfied literals we selected.

The set $S = \{v_1, v_2, ...v_m\}$ must form an m-clique in G. Why? Well notice that $z_1^*, z_2^*, ..., z_m^*$ all have the same truth assignment, since otherwise $z_i^* = \neg z_j^*$ for some $i, j \in [1, m]$ thus implying that one of z_i^* and z_j^* is not the satisfying literal of C_i, C_j, a contradiction to our choice of the z^* literals. Notice also that the z^*'s belong to different clauses, that's how we chose them. Therefore, by the construction of G, every pair of $v_1, ..., v_n$ must have a connecting edge and thus $S = \{v_1, v_2, ...v_m\}$ forms an m-clique in G.

Conversely, suppose G has a clique of size at least $m = k$. Let $v_1, v_2, ..., v_q$ be a clique in G of size $q \geq m$, then the first m vertices $v_1, ..., v_m$ must also form a clique in G. Since there are no edges connecting vertices from the same clause, every v_i corresponds to a literal z_i from exactly one clause C_i. Moreover, since $v_1, ..., v_m$ is a clique, the corresponding literals z_i, z_j of any pair $v_i, v_j \in \{v_1, ..., v_m\}$ can be satisfied simultaneously (by construction). Now, to construct a satisfying assignment $x_1, ..., x_n$ for ϕ, we just need to satisfy all of $z_1, ..., z_m$ and assign the remaining variables arbitrarily. Every C_i contains one z_i, and every z_i is satisfied thus every C_i is satisfied and so ϕ is satisfied.

To conclude, we've shown that CLIQUE is in NP and that it is NP-hard by giving a reduction from 3-SAT. Therefore CLIQUE is NP-complete. $\square$