CSE 3500 Algorithms and Complexity – Fall 2016
Lecture 26: December 1, 2016

Clique is $\mathcal{NP}$-complete

- We have shown that Clique is a member of $\mathcal{NP}$.
- We will show that SAT $\propto$ Clique.
- Let $F = C_1 \wedge C_2 \wedge \cdots \wedge C_k$ be any Boolean formula in CNF. Here $C_1, C_2, \ldots, C_k$ are clauses, where each clause is a disjunction of literals. (A literal is either a variable or its negation). The input for the Clique problem will be a graph and an integer.
- We generate the following input for Clique: $(G(V, E); k)$, where there is a node in G for every literal in every clause of F. If x_q is a literal in C_i, then the node corresponding to this literal is denoted as (x_q, i).
- Two nodes (x_q, i) and (x_r, j) will be connected by an edge if and only if $i \neq j$ and $x_q \neq \bar{x}_r$.
- There are $O(|F|)$ nodes in G. Thus G can be constructed in $O(|F|^2)$ time, which is clearly a polynomial in the input size.
- Now we have to show that the reduction is correct.
- Assume that F is satisfiable. In this case we will show that G has a clique of size k. If F is satisfiable, then, in the satisfying assignment, there will be at least one literal in each clause whose value is T. Let σ_i be a literal in C_i whose value is T in the satisfying assignment, for $1 \leq i \leq k$. Then, consider the nodes $(\sigma_1, 1), (\sigma_2, 2), \ldots, (\sigma_k, k)$. Every pair of these nodes is connected by an edge in G since each node corresponds to a distinct clause and no two of the literals $\sigma_1, \sigma_2, \ldots, \sigma_k$ negate each other. In other words, G has a clique of size k.
- Now assume that G has a clique of size k. Let the nodes that form a clique be $(\sigma_1, 1), (\sigma_2, 2), \ldots, (\sigma_k, k)$. Then, the following assignment will satisfy F: $\sigma_i = T$, for $1 \leq i \leq k$. This is because σ_i is in C_i, for $1 \leq i \leq k$. Also, no two of the literals $\sigma_1, \sigma_2, \ldots, \sigma_k$ can negate each other and hence all of these literals can be set to T and there will not be any conflicts.
- In summary, we have shown that $SAT \propto$ Clique and hence Clique is also $\mathcal{NP}$-complete. $\square$

Optimization and Decision Versions of Problems

- If there exists an efficient algorithm to solve the decision version of a problem, then we can also devise an efficient algorithm for solving the optimization version.

- As an example, consider the Clique problem. Assume that there exists a deterministic polynomial time algorithm CLQ to solve the decision version of the problem. In particular, given a graph $G(V, E)$ and an integer k ($1 \leq k \leq |V|$), CLQ decides in $p(n)$ time if G has a clique of size k or not. Here $p(n)$ is a polynomial of constant degree. We can use CLQ to solve the optimization problem also in a deterministic polynomial time. Specifically, this algorithm will identify the largest k such that G has a clique of size k, given any graph $G(V, E)$ as the input. Here is such an algorithm:

  for $k = |V|$ **downto** 1 **do**
      if CLQ($G(V, E); k$) **then** output k and quit;

  The run time of the above algorithm is $O(n\ p(n))$ which is a constant degree polynomial.

- As another example, consider the SUBSET SUM problem. This problem takes as input a set $S = \{a_1, a_2, \ldots, a_n\}$ of real numbers and another real number A. The problem is to check if there exists a subset S' of S such that the sum of all the elements in S' is A. An optimization version of the problem will also have a set S and a number A as the input. If there exists a subset S' of S whose elements sum to A, then we are required to find one such subset.

- Assume that SSET is a deterministic algorithm for solving the decision version of the SUBSET SUM problem whose run time is a constant degree polynomial in n. Consider the following algorithm that finds a subset S' of S whose elements sum to A (if there is one such subset):

  for $i = 1$ **to** n **do**
      if the elements in S sum to A **then** output S and quit;
      if SSET($S - \{a_i\}$) **then** $S = S - \{a_i\}$;

  Let $p(n)$ be the run time of SSET. Then SSET is called at most n times. These calls will cost $O(n\ p(n))$ time. Also, the elements in S are summed at most n times. The total time needed for these sums will be $O(n^2)$. Thus the total run time of this algorithm is $O(n\ p(n) + n^2)$ which is a constant degree polynomial.

Node Cover Decision Problem (NCDP) is $\mathcal{NP}$-complete

- The NCDP takes as input an undirected graph $G(V, E)$ and an integer k (for some k, $1 \leq k \leq |V|$). The problem is to check if G has a node cover of size k.

- A node cover for an undirected graph $G(V, E)$ is a subset V' of V such that for every edge $(a, b) \in E$, either a is in V' or b is in V'. As an example, if $V = \{1, 2, 3, 4\}$ and $E = \{(1, 2), (1, 4), (2, 4), (2, 3), (4, 3), (3, 5), (4, 5)\}$, then, the following are two possible node covers: $\{1, 3, 4\}$ and $\{2, 4, 5\}$.

- **Lemma:** NCDP is $\mathcal{NP}$-complete.

  Proof: It is easy to show that NCDP in in $\mathcal{NP}$. We'll show that Clique $\propto$ NCDP. Let $(G(V, E); k)$ be any input for the Clique problem and let $n = |V|$. We generate the following instance for the NCDP: $(G'(V, E'); n - k)$, where $(a, b) \in E'$ if $(a, b) \notin E$ and $(a, b) \notin E'$ if $(a, b) \in E$ (for every pair of distinct nodes a and b in V). $G'(V, E')$ is known as the complement of $G(V, E)$. Clearly, G' can be constructed in $O(n^2)$ time.

  Assume that G has a clique of size k. Let the set of nodes that form a clique be $A = \{v_1, v_2, \ldots, v_k\}$. No two of these nodes will be connected by an edge in G'. This means in any node cover for G' there is no need to include any of these nodes. In other words, $V - A$ is a node cover for G' and $|V - A| = n - k$.

  Now assume that G' has a node cover of size $n - k$. Let the set of nodes that form a cover be $B = \{u_1, u_2, \ldots, u_{n-k}\}$. Let $C = V - B = \{w_1, w_2, \ldots, w_k\}$. No two nodes in C will be connected by an edge in G'. This is beacuse if (w_i, w_j) is an edge in G' (for some i and j), then at least one of these nodes should be in the node cover. This means that each pair of nodes in C will be connected by an edge in G. This means that G has a clique of size k.

  In summary, Clique $\propto$ NCDP and hence NCDP is also $\mathcal{NP}$-complete. $\square$