Tutorial 5-6

Proof sketches.

a) Vertex Cover

Proof sketch: To see that Vertex Cover is in NP, given a graph (V, E), guess a $V' \subseteq V$, and verify that (i) $|V'| \leq k$, and (ii) for all $(v, w) \in E$, at least one of v, w is in V'. If the verification is successful, then accept; otherwise reject.

To show that Vertex Cover is NP-hard, consider the following reduction from 3SAT.

Suppose $U = \{x_1, x_2, \ldots, x_n\}$ is the set of variables and $C = \{c_1, c_2, \ldots, c_m\}$ is the set of clauses, where $c_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$.

Then form the vertex cover instance $G = (V, E)$, where

$V = \{u_i, w_i : 1 \leq i \leq n\} \cup \{z_{j,1}, z_{j,2}, z_{j,3} : 1 \leq j \leq m\}$.

Let $E = \{(u_i, w_i) : 1 \leq i \leq n\} \cup \{(z_{j,1}, z_{j,2}), (z_{j,2}, z_{j,3}), (z_{j,1}, z_{j,3}) : 1 \leq j \leq m\} \cup \{(z_{j,r}, u_i) : l_{j,r} = x_i\} \cup \{(z_{j,r}, w_i) : l_{j,r} = \neg x_i\}$.

Let $k = 2m + n$

Intuitively, u_i represents x_i and w_i represents $\neg x_i$. $z_{j,r}$ represents the literal $l_{j,r}$. Clearly the above reduction can be done in polynomial time.

It is easy to verify that in any vertex cover, one must have (i) at least one of u_i, w_i for each i, $1 \leq i \leq n$ and (ii) at least two of $z_{j,1}, z_{j,2}, z_{j,3}$, for each j, $1 \leq j \leq m$. Thus, any vertex cover for G of size at most $2m + n$ must contain exactly one of u_i, w_i for each i, $1 \leq i \leq n$ and exactly two of $z_{j,1}, z_{j,2}, z_{j,3}$, for each j, $1 \leq j \leq m$.

If the 3SAT problem (U, C) has a satisfying assignment, then by correspondingly choosing u_i in V' iff x_i is true, w_i in V' iff x_i is false, and choosing two of $z_{j,1}, z_{j,2}, z_{j,3}$ to be in V' such that if $z_{j,r}$ is left out of V' then the literal $l_{j,r}$ is true, we can easily verify that V' is a vertex cover of G.

If the Vertex Cover problem (V, E) has a vertex cover, then consider the truth assignment: x_i is true iff u_i is in the vertex cover. It can now be shown that if $z_{j,r}$ is not in the vertex cover then, $l_{j,r}$ must be true (otherwise, both the vertices of the edge $(z_{j,r}, s_i)$ are not in the vertex cover, where s_i is u_i, if $l_{j,r} = x_i$, and s_i is w_i, if $l_{j,r} = \neg x_i$.)

b) Clique:

Suppose $G = (V, E)$ is a graph. Then, one can show that $G = (V, E)$ has a vertex cover of size k iff $G = (V, E)$ has an independent set of size $|V| - k$ iff $G = (V, E^c)$ has a clique of size $|V| - k$. Here $E^c = \{(u, v) : u, v \in V, u \neq v\} - E$.

This proves that Clique and independent set are NP-complete.

c) X3C: It is easy to verify that X3C is in NP. Just guess a cover $C' \subseteq C$ of size q, and check that $\bigcup_{S \in C'} S = A$.

NP Hardness: By reduction from 3DM. Let (X, Y, Z, S) be a 3DM instance. Without loss of generality assume that X, Y, Z are pairwise disjoint. Then construct a X3C instance (A, C) as follows (here A is the set, and C is the collection of subsets of A):

$A = X \cup Y \cup Z$.

$C = \{\{x, y, z\} : (x, y, z) \in S\}$.

It is easy to verify that X3C instance above has a solution iff there is a matching in the 3DM instance and the reduction can be done in polynomial time.

d) 3-Colorability.

Proof sketch: To show that graph 3-colorability problem is in NP, just guess a coloring using colors $0, 1, 2$ and verify that for all edges (u, v) in the graph, u and v have different colors.

To show that it is NP-hard, consider reduction from 3-SAT.

Let (U, C) be a 3-SAT problem.

Suppose $U = \{x_1, x_2, \ldots, x_n\}$ and $C = \{c_1, c_2, \ldots, c_m\}$, where $c_i = (\ell_i^1 \vee \ell_i^2 \vee \ell_i^3)$.

Then, construct the following 3-color problem:

$V = \{u_i, w_i : 1 \leq i \leq n\} \cup \{C0, C2\} \cup \{a_i^1, a_i^2, a_i^3, a_i^4, a_i^5, a_i^6 : 1 \leq i \leq m\}$.

Let $b_i^r = u_j$, if $\ell_i^r = x_j$, and $b_i^r = w_j$, if $\ell_i^r = \neg x_j$.

$E = \{(C0, C2)\} \cup E1 \cup E2 \cup E3 \cup E4$, where

$E1 = \{(u_i, w_i), (u_i, C2), (w_i, C2) : 1 \leq i \leq n\}$

$E2 = \{(a_i^6, C0), (a_i^6, C2) : 1 \leq i \leq m\}$

$E3 = \{(a_i^1, a_i^2), (a_i^1, a_i^4), (a_i^2, a_i^4), (a_i^4, a_i^5), (a_i^3, a_i^5), (a_i^3, a_i^6), (a_i^5, a_i^6) : 1 \leq i \leq m\}$

$E4 = \{(b_i^r, a_i^r) : 1 \leq i \leq m, 1 \leq r \leq 3\}$.

Clearly the above reduction can be done in polynomial time.

Now, if the 3-SAT problem (U, C) is satisfiable, then fix one such satisfying assignment. Color the vertices in V as follows.

u_i is colored 1, and w_i is colored 0 iff x_i is true.

u_i is colored 0, and w_i is colored 1 iff x_i is false.

$C0$ is colored 0.

$C2$ is colored 2.

a_i^4 is colored as $(\mathit{color}(b_i^1)$ OR $\mathit{color}(b_i^2))$ and a_i^6 is colored as 1 (that is $(\mathit{color}(a_i^4)$ OR $\mathit{color}(b_i^3))$.

One of a_i^1, a_i^2 is colored 2 and the other is colored $1 - \mathit{color}(a_i^4)$ (note that this can be done as at least one of b_i^1, b_i^2 is the same color as a_i^4). Similarly, one of a_i^3, a_i^5 is colored 2 and the other is colored $1 - \mathit{color}(a_i^6)$ (note that this can be done as at least one of a_i^4, b_i^3 is the same color as a_i^6).

On the other hand, if the coloring is possible, then without loss of generality assume that the color of $C2$ is 2, and color of $C0$ is 0. This implies that color of one of u_i, w_i is 0 and the other 1. Consider the truth assignment $Q(x_i)$ =true iff u_i is colored 1.

Now, if b_i^r, $1 \leq r \leq 3$ are all colored 0, then a_i^4 and a_i^6 must also be colored 0. But this will cause a conflict with the edge $(a_i^6, C0)$. Thus, at least one of b_i^r, $1 \leq r \leq 3$ is colored 1, and thus at least one literal in c_i is true.

e) Not-All-Equal SAT (NAESAT).

Proof sketch: It is easy to see that NAESAT is in NP: Certificates would be truth assignment which make each clause have at least one true and at least one false literal. Verification checks if the truth assignment indeed makes at least one literal true and one literal false in each clause.

To show that NAESAT is NP-hard, we reduce 3-SAT to NAESAT.

Suppose (U, C) is an instance of 3SAT. Suppose the clauses in C are $c_1, \ldots, c_m$.

Let $U' = U \cup \{w_i, r_i : 1 \leq i \leq m\} \cup \{T\}$, where r_i, w_i and T are new variables.

Suppose the i-th clause c_i is $(l_1^i \vee l_2^i \vee l_3^i)$. Then, let $C' = \{(l_1^i \vee l_2^i \vee w_i), (l_1^i \vee l_3^i \vee r_i), (r_i \vee w_i \vee T) : 1 \leq i \leq m\}$.

Clearly the above reduction can be done in polynomial time.

We claim that (U, C) is satisfiable iff (U', C') is in NAESAT.

Suppose (U, C) is satisfiable: Fix a truth assignment $A(\cdot)$ to variables in U which satisfies all clauses in C.

Consider the following truth assignment A' to variables in U'. $A'(v) = A(v)$, for $v \in U$. $A'(T)$ =True. For, $1 \leq i \leq m$, let $A'(w_i)$ be false iff at least one of l_1^i or l_2^i is true. For, $1 \leq i \leq m$, let $A'(r_i)$ be false iff at least one of l_1^i or l_3^i is true. Note that at least one of r_i and w_i is false. It is now easy to verify that the truth assignment A' witnesses that each clause in C' has at least one true literal and at least one false literal.

Now suppose that there exists a truth assignment A' to variables in U' such that each clause in C' has at least one true literal and at least one false literal. Without loss of generality assume that $A'(T)$ is true (otherwise just flip the truth assignment of each variable). We claim that A' restricted to variables in U is a satisfying truth assignment for C. To see this note that both r_i and w_i cannot be true (otherwise all three literals in (r_i, w_i, T) are true). If w_i is false, then at least one of l_1^i, l_2^i is true. If r_i is false, then at least one of l_1^i, l_3^i is true. Thus C is satisfiable.

(f) MAX2SAT

Proof sketch: It is easy to see that MAX2SAT is in NP: Certificates would be truth assignment which makes at least k of the clauses true. For verification, check if the truth assignment indeed makes at least one literal true in at least k of the clauses.

To show that MAX2SAT is NP-hard, we reduce 3-SAT to MAX2SAT.

Suppose (U, C) is an instance of 3SAT. Suppose the clauses in C are $c_1, \ldots, c_m$, where clause c_i is $(l_{i,1}, l_{i,2}, l_{i,3})$.

Then the instance of MAX2SAT (U', C', k') is created as follows.

$k' = 7m$.

$V' = V \cup \{w_i : 1 \leq i \leq m\}$.

$C' = \bigcup_{1 \leq i \leq m} C'_i$, where C'_i consists of 10 clauses as follows:

$(l_{i,1}), (l_{i,2}), (l_{i,3}), (w_i)$, $(\neg l_{i,1} \vee \neg l_{i,2})$, $(\neg l_{i,2} \vee \neg l_{i,3})$, $(\neg l_{i,1} \vee \neg l_{i,3})$, $(l_{i,1} \vee \neg w_i)$, $(l_{i,2} \vee \neg w_i)$, $(l_{i,3} \vee \neg w_i)$.

Clearly the above reduction can be done in polynomial time.

Note that if all of $l_{i,1}, l_{i,2}, l_{i,3}$ are false, then one can make at most six of the above clauses true.

If at least one of $l_{i,1}, l_{i,2}, l_{i,3}$ is true, then we can make 7 clauses true by making some truth assignment to w_i (by setting w_i to be true if all of x, y, z are true, and w_i to be false otherwise). Moreover, there is no assignment to w_i which will make more than 7 clauses true in all the cases.

It follows that one can satisfy at least $7m$ of the clauses in C' iff there is a satisfying assignment which makes all the clauses in C true.