Lecture 15: A few more NP-complete problems

Valentine Kabanets

November 15, 2016

1 Clique, IndependentSet, VertexCover, and HamCycle

A *clique* in a graph G is a subset of vertices of G such that every pair of nodes in the subset are connected by an edge. An *independent set* in a graph is a set of vertices such that no two of them are connected by an edge.

A *vertex cover* for a graph G is a set S of vertices such that every edge of G has at least one of its endpoints in S (every edge of G is covered by S).

An st-path in a graph G is a path from a specified vertex s to a specified vertex t of G. A *Hamiltonian* st-path in G is an st-path that visits each vertex of G exactly once. A *Hamiltonian cycle* is a cycle in a graph that visits each vertex exactly once.

Define

$$Clique = \{\langle G, k\rangle \mid \text{ graph } G \text{ has a clique of size } \geq k\},$$

$$IS = \{\langle G, k\rangle \mid \text{ graph } G \text{ has independent set of size } \geq k\},$$

$$VC = \{\langle G, k\rangle \mid \text{graph } G \text{ has vertex cover of size } \leq k\},$$

$$st\text{-}HamPath = \{\langle G\rangle \mid G \text{ has a Hamiltonian } st\text{-path}\},$$

$$HamCycle = \{\langle G\rangle \mid G \text{ has a Hamiltonian cycle}\}$$

All of these problems are NP-complete. We'll give reductions:

$$3SAT \leq_p CLIQUE \leq_p IS \leq_p VC.$$

Recall the problem **3SAT:** Given a 3cnf formula $\phi(x_1, \ldots, x_n)$ (which is a conjunctions of clauses, where each clause is an OR of 3 literals; each literal is a variable or the negation of a variable), decide if ϕ is satisfiable.

Theorem 1. $3SAT \leq_p CLIQUE$

Proof. Given a 3cnf $\phi(x_1, \ldots, x_n)$, let m be the number of clauses in ϕ. Define a graph G to be on $3m$ vertices, one triple of vertices for each clause of ϕ. Also, imagine labeling the vertices in each triple by the literals of the corresponding clause. Add edges between every pair of vertices of the graph *except*

- nodes in the same triple (corresponding to the same clause), and

- nodes labeled by the complementary literals (e.g., x and $\bar{x}$).

We claim that if ϕ is satisfiable, then G has a clique of size m. Indeed, pick one true literal from each clause. The corresponding set of m vertices forms a clique in G (because no complementary literals are chosen, as we choose true literals only). For the other direction, if G has a clique of size m, it must be the case that exactly one vertex from each triple/clause is in the clique. Moreover, no conflicting literals can be in the clique. Therefore, it is possible to assign True to every literal/vertex in the clique. But such an assignment will satisfy all clauses of ϕ, so ϕ is satisfiable. $\square$

For the other reductions, we rely on the following easy observations. Recall that the *complement* G^c of a graph G is the graph on the same set of vertices such that edges of G are non-edges of G^c and, conversely, non-edges of G are edges of G^c.

Lemma 1. *Graph G has an clique of size k iff G^c has an independent set of size k.*

Lemma 2. *Graph G has an independent set of size k iff G has a vertex cover of size $n-k$.*

For $CLIQUE \leq_p IS$: the reduction maps $\langle G, k\rangle$ to $\langle G^c, k\rangle$, where G^c is the complement of the graph G (the graph on the same set of vertices as G but such that (u, v) is an edge of G^c iff (u, v) is not an edge of G). Correctness of the reduction is by Lemma 1.

For $IS \leq_p VC$: the reduction maps $\langle G, k\rangle$ to $\langle G, n-k\rangle$, where $n =$ the number of vertices in G. Correctness is by Lemma 2.

The textbook shows that *st*-HamPath is NP-complete. Using this, we'll show that HamCycle is also NP-complete, via the reduction *st-HamPath* $\leq_p$ *HamCycle*. The reduction maps a graph G on n nodes to a graph G' which is the same as G plus a new vertex a with two edges (s, a) and (t, a). It is not hard to see that this is a correct reduction.

2 NP-completeness of SubsetSum

Recall **SubsetSum:** Given natural numbers $a_1, \ldots, a_n, T$, decide if there is a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i\in S} a_i = T$.

Theorem 2. *SubsetSum is* NP*-complete.*

Proof. Clearly, SubsetSum is in NP. We will show that $3SAT \leq_p SubsetSum$.

The idea is to define a table whose rows will represent the numbers a_i, in the decimal notation: each position of the row holds a digit (between 0 and 9).

Given a 3-cnf on n variables and m clauses, we specify $2n$ "literal" rows (one row for each literal), and $2m$ "clause" rows (two rows for each clause). There will be $n+m$ columns in the table. The first n columns correspond to the n variables $x_1, \ldots, x_n$ (variable columns), and the remaining m columns correspond to clauses $c_1, \ldots, c_m$ (clause columns).

The row corresponding to variable x_i has 1 in variable column i, and 1 in clause column c_j for every clause j containing literal x_i. The row corresponding to literal $\bar{x}_i$ has 1 in variable column i, and 1 in clause column c_j for every clause j containing literal $\bar{x}_i$. For each clause j, we'll have two clause rows that have 1 in the clause column j, and zero everywhere else. All unspecified entries are 0.

Finally, we define the target row T to be 1 in each variable column i, $1 \leq i \leq n$, and 3 in each clause column j, $1 \leq j \leq m$.

We claim that the input 3-cnf is satisfiable iff the constructed instance of SubsetSum has a solution subset S.

For one direction, suppose 3-cnf is satisfiable. Let a be a satisfying assignment for the 3-cnf. Define S to consist of those literal rows that are true under assignment a, plus add to S 0, 1, or 2 clause rows for each clause c_j depending on whether the assignment a makes true 3, 2, or 1 literal in c_j. Observe that the defined set S of rows will add up to have 1 in each variable column, and 3 in each clause column, as required.

For the opposite direction, suppose we have a subset S of rows that adds up to T. The fact that each variable column of T is 1 forces S to contain exactly one literal row for each variable x_i (either x_i or $\bar{x}_i$). Thus, S encodes a legal truth assignment: we assign x_i True if S contains row x_i, and False if S contains row $\bar{x}_i$.

We claim that this assignment is satisfying for each clause c_j. Indeed, since S must add up to 3 in every clause column c_j, we get that each clause column c_j must contain at least one literal row chosen in S (otherwise, we can't get to the sum 3 in that column). But the latter means that c_j contains a literal that is assigned True by the assignment we extracted from the set S. Thus, each clause contains at least one true literal, and so our 3-cnf is satisfiable.

Finally, it is easy to see that the described reduction is polytime. This concludes the proof. $\square$

3 Traveling Salesman Problem

Finally, define the *Traveling Salesman Problem (TSP)*: **given** a complete graph on n vertices, an assignment of positive integer weights to all edges, and an integer W, **decide** if there exists a Hamiltonian cycle in G of weight at most W, where the weight of the cycle is the sum of the weights of its edges.

We'll show TSP is NP-complete, via the reduction $HamCycle \leq_p TSP$. The reduction maps a graph G on n nodes to a complete graph on n nodes, with the weights 1 for all edges of G, and weights $n+1$ for all non-edges of G. We also set $W = n$.

Observe that if G has a Hamiltonian cycle, then this cycle is a tour of weight n in our TSP instance. Conversely, if there is a tour of weight at most n, that tour can't use any non-edges of G (which would make its total weight at least $n+1$). Thus, the original graph G must have a Hamiltonian cycle.