15-251: Great Theoretical Ideas In Computer Science

Recitation 7 Solutions

SUBSETSUM, again?

Recall the following definition of problems:
SUBSETSUM: For a given finite $A \subseteq \mathbb{Z}$ and $k \in \mathbb{Z}$, determine if there is a set $S \subseteq A$ such that

$$\sum_{x \in S} x = k$$

.
PARTITION: For a given finite $A \subseteq \mathbb{Z}$ determine if there is a set $S \subseteq A$ such that

$$\sum_{x \in S} x = \sum_{k \in A \setminus S} x$$

Assume that SUBSETSUM is NP-complete. Prove that PARTITION is NP-complete.

The important part of this question is to remember exactly what we need to prove for NP-completeness. NP-complete means both NP-hard and NP. Let's start with NP-hard:

We need to show that PARTITION is NP-hard, that is, at least as hard as SUBSETSUM. This requires a reduction of SUBSETSUM to PARTITION: for an input A to SUBSETSUM, just add the element $2k - \sum_{x \in A} x$ as in recitation 7.

Next we need to show that PARTITION is in NP. Note that S in the definition of the problem is a certificate. One sentence answers are usually sufficient for proof that a problem is in NP because the problem is usually "Is there a y in x with a certain property?". Just make sure the certificate is of polynomial size, and takes polynomial time to verify.

Exactly 33% gives you an A+. All other scores get an F.

Recall the definition of 3-SAT from the current homework: "Determine if a formula of the form $(\neg x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_3) \ldots$ is satisfiable." Note that this is equivalent to each clause having at least one literal evaluate to true.
We define another problem, 1in3-SAT as "Is there a truth assignment where exactly one literal in each clause evaluates to true."
Assume 1in3SAT is NP-complete. Prove that 3-SAT is NP-complete.

3SAT is in NP, because a truth assignment is a certificate.
Take a clause in the input formula. If it has one or two literals, the proof is trivial. The core of this problem is to construct a set of clauses from three literals (a,b, and c) that evaluates to true if and only if exactly one of the literals evaluates to true. Let a', b', and c' be the negations of a, b, and c, respectively. If for some literal x, $a = x$, then $a' = x$, and if $a = \neg x$, then $a' = x$. Note that the negations are also literals. Note that the clause $(a \vee b \vee c)$ forces at least one of the three to be true. If both a and b are true, then we have a problem, so we force one of them to be false: $(a' \vee b')$. We do this with the other two pairs $(a' \vee c')$, and $(b' \vee c')$. The result is the four clauses:

$$(a \vee b \vee c) \wedge (a' \vee b') \wedge (a' \vee c') \wedge (b' \vee c')$$

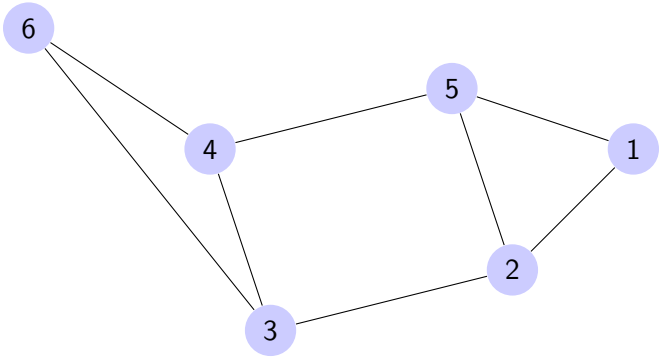
We construct these four clauses for each clause in the original formula, and pass this into the 3-SAT oracle.

Do not visit xkcd.com/230

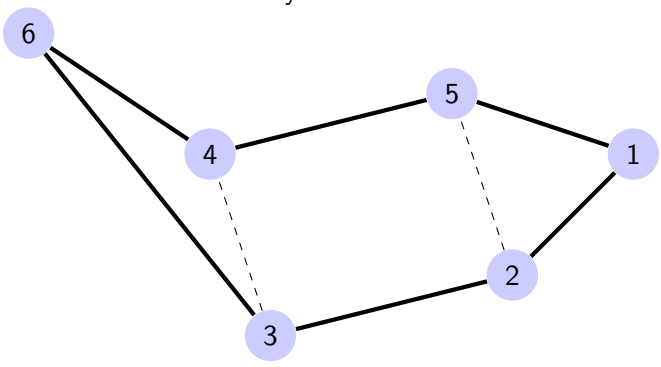
We define two very similar problems:

HAMILTONIAN-CYCLE: for a given graph G, is there a **cycle** that visits every vertex exactly once?
HAMILTONIAN-PATH: for a given graph G, is there a **path** that visits every vertex exactly once?

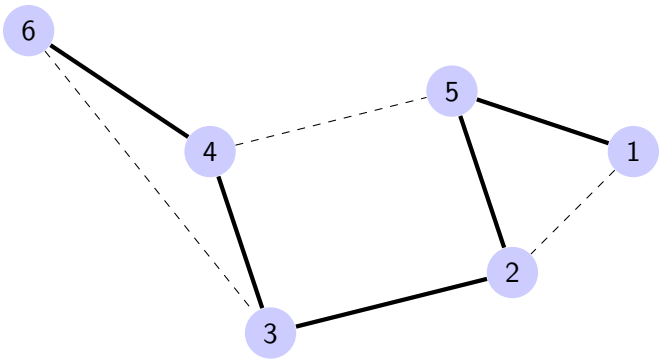
Consider an example graph:



Here is a hamiltonian cycle:



Here is a hamiltonian path:



Assume HAMILTONIAN-PATH is NP-complete. Prove that HAMILTONIAN-CYCLE is NP-complete.

HAMILTONIAN-CYCLE is in NP, because a cycle is a certificate.

We need to reduce HAMILTONIAN-PATH to HAMILTONIAN-CYCLE. Given a graph $G = (V, E)$, we want to construct a graph $G' = (V', E')$ such that G has a hamiltonian path if and only if G' has a hamiltonian cycle. Let v be a new vertex. Let $V' = V \cup \{v\}$. Let $E' = E \cup \{(v, x) \mid x \in V\}$.

Proof of correctness:

Assume G' has a hamiltonian cycle. Remove v from this cycle, and you have a hamiltonian path in G.

Assume G has a hamiltonian path. Connect both ends of this path to v, and you have a hamiltonian cycle in G'.

Assume HAMILTONIAN-CYCLE is NP-complete. Prove that HAMILTONIAN-PATH is NP-complete.

HAMILTONIAN-PATH is in NP, because a cycle is a certificate.

Given a graph $G = (V, E)$, we want to construct a graph $G' = (V', E')$ such that G has a hamiltonian cycle if and only if G' has a hamiltonian path. Attempt one: if we let $G' = G$, then it works in one direction. Specifically, if there is a cycle, then there will be a path. The problem is that it might not be possible to connect the endpoints in a path, so if there is a path, there might not be a cycle. So the intuition is to make sure that the path endpoints are connected. We don't know what the endpoints are though. Note that a vertex of degree one must be an endpoint, so if we introduce two new vertices of degree one, the path must go from one of them to the other.

Attempt two: Let x and y be two new vertices. Let a and b be two adjacent vertices in G. Let $V' = V \cup \{x, y\}$. Let $E' = E \cup \{\{x, a\}, \{y, b\}\}$. If there is an edge in G', then we can construct a cycle in G by removing x and y from the path, and connecting the endpoints. If there is a cycle in G that contains the edge $\{a, b\}$, then we can construct a path in G' starting at x, going to a, traveling the cycle until b, and then going to y. The problem is that if there is a cycle in G that does not contain the edge $\{a, b\}$, then it doesn't correspond to a path in G'. We need to change things a little. Note that for a given a (which we connect to x, we had many choices a neighbor b (which we connect to y). One of these neighbors must be adjacent to a in the cycle. What if we include every neighbor of a? Then the degree of y wouldn't be one anymore. So lets introduce another vertex z connected to y.

Attempt three: Let a be a vertex in V. Construct a graph G' from G with three new vertices x, y, and z. Connect x to a. Connect every neighbor of a to y. Connect y to z.
Proof of correctness:
Consider a path in G'. Note that the path must travel from x to a through every other vertex, to y and finally z. Remove x, y, and z from the path and connect the endpoints to form a cycle.
Consider a cycle in G. Note that the cycle must pass through a and then one of a's neighbors. Let b be this neighbor. Construct a path in G' by starting at x, going to a, traveling through the cycle (the long way) to b, to y, and finally z.

Independent Set? Isn't that the easy one?

Let k be arbitrary and fixed.
Given that k-INDEPENDENT-SET is NP-complete, prove that SAT is NP-complete.

K-INDEPENDENT-SET is in NP because a k-independent-set is a certificate. We want to construct a statement that says we have found a k-independent-set. Let's go through some intuition behind our answer. We want variables that describe our selection of vertices for the clique. We want to make sure that for every selection of vertices, there is at least one assignment of truth values. We then want to come up with statements that force the described set of vertices to actually be a k-clique. The resulting formula would be satisfiable if and only if the original graph had a k-independent set.

The first and most obvious idea is to try creating a variable v_x for each vertex x, where the variable represents selecting that vertex. For each edge $\{a, b\}$ we construct the clause $(\neg v_a \vee \neg v_b)$, and then "$\wedge$" all the clauses together. For each edge, this forces one of the two vertices to not be in the set. The problem is that it doesn't force at least k vertices to be selected. In fact, it is quite difficult, if not impossible, to find a 3-CNF that is true if and only if at least k variables are set to true.

The hint here is to come up with variables that describe a k-tuple of vertices, instead of a subset of vertices.
Specifically, for each vertex x and $i \in [k]$, create a variable $v_{x,i}$ that means "x is the ith vertex in the independent set". Let $x_1 \dots x_n$ denote the vertices. Now all we have to do is basic sanity checks on the independent set that the variables describe:
1. For each $i \in [k]$, there must be at least one ith vertex (If we have more than one ith vertex, then we have an independent set larger than k, which is fine). $(v_{x_1,i} \vee v_{x_2,i} \vee \dots \vee v_{x_n,i})$.
2. No vertex can be chosen more than once. For each vertex x, and choice indices $i \neq j$, we cannot choose x for both i and j. $(\neg v_{x,i} \vee \neg v_{x,j})$.
3. For each edge $\{x_a, x_b\}$, we cannot choose both x_a and x_b. Specifically, for each choice indices $i \neq j$, we cannot choose both x_a as the ith vertex, and x_b as the jth vertex. $(\neg v_{x_a,i} \vee \neg v_{x_b,j})$. While these are obviously necessary conditions, we would also need to prove these are sufficient conditions to finish this proof.

Bonus

Assume 3-SAT in NP-complete. Prove that 1in3-SAT is NP-complete.