NP-completeness IV

We are now going to give many examples of NP-complete problems, and we are going to show that they are NP-complete. In this part of the course the goal is to learn to recognize which problems are NP-complete, and to learn to demonstrate that they are indeed NP-complete. Recall that by showing that a certain problem is NP-complete, we show that it does not have a polytime solution (i.e., an efficient solution)—under the assumption that P $\neq$ NP.

The practical use of this is that once we know that a given problem is NP-complete, we can concentrate on finding approximate solutions, or partial solutions, or solutions to restricted instances of our problem, as a complete efficient solution is *not* possible (again, assuming that P $\neq$ NP).

One more thing: make sure that you understand the distinction between $\leq_p$ and $\xrightarrow{p}$. We use $\leq_p$ to reduce (in polytime) one decision problem to another, usually to show that a problem is NP-hard. We use $\xrightarrow{p}$ to reduce (in polytime) a search problem to its related decision problem. In the case of $\leq_p$, we reduce instances of one problem to instances of another problem; in the case of $\xrightarrow{p}$, we design an algorithm which uses a "black-box" solver of one problem to solve another, by repeatedly querying the "black-box".

CircuitSAT

We first define a "Boolean, combinational circuit" (or for conciseness, just "Circuit"). A circuit is just a "hardwired" algorithm that works on a fixed-length string of n Boolean inputs. Each gate is either an "and" gate, an "or" gate, or a "not" gate; a gate has either one or two inputs, each of which is either an input bit, or the output of a previous gate. The last gate is considered the output gate. There is no "feedback", so a circuit is essentially a directed, acyclic graph, where n nodes are input nodes and all the other nodes are labelled with a Boolean function.

More formally, we can view a circuit C with n input bits as follows. We view the i-th gate as computing a bit value into a variable x_i. It is convenient to view the inputs as gates, so that the first n gate values are $x_1, x_2, \ldots, x_n$. Each of the other gates compute bit values into variables $x_{n+1}, x_{n+2}, \ldots, x_m$. For $n < i \leq m$, the i-th gate will perform one of the following computations:

$$x_i \leftarrow x_j \vee x_k$$
$$x_i \leftarrow x_j \wedge x_k$$
$$x_i \leftarrow \neg x_j$$

where $1 \leq j, k < i$. The output of C is the value of x_m, and we say that C *accepts* $a_1 a_2 \ldots a_n \in$

$\{0,1\}^n$ if, when $x_1, \ldots, x_n$ are assigned the values $a_1, \ldots, a_n$, x_m gets assigned the value 1. We can now define the decision problem (i.e., language) CircuitSAT as follows:

CircuitSAT:
Input: $\langle C \rangle$ where C is a circuit on n inputs.
Question: Does C accept some $a_1 a_2 \ldots a_n \in \{0,1\}^n$?

Lemma 1 CircuitSAT is NP-complete.

Proof: Exercise. (Below we give a detailed proof of the fact that the problem Independent Set (IS) is NP-complete; you can follows the steps of that proof to show that CircuitSAT is NP-complete, as well as the fat hints given in class). □

Independent Set

If $G = (V, E)$ and $V_0 \subseteq V$, then V_0 is an *independent set* iff no two nodes in V_0 are connected by an edge in E.

IS:
Input: $G = (V, E)$, $B \in \mathbb{N}$, $B \leq |V|$.
Question: Does G have an independent set of size B?
An equivalent question is: Does G have an independent set of size at least B?

Related search problems:
ISS: given a graph, find the size of a largest independent set.
ISS′: given a graph, find a largest independent set.

Exercise: Show that ISS $\xrightarrow{p}$ IS, and that ISS′ $\xrightarrow{p}$ IS

Lemma 2 IS is NP-complete.

Proof: First we show that IS $\in$ NP: guess a set of vertices $V_0 \subseteq V$, check that $|V_0| = B$ and that no edge in E connects two vertices in V_0.

Now we show that 3SAT $\leq_p$ IS.

Reduction: Given an input $I = \{C_1, \ldots, C_m\}$ to 3SAT, we must find an input $\langle G_I, B_I \rangle$ to IS such that $I \in$ 3SAT iff $\langle G_I, B_I \rangle \in$ IS.

In general, for each instance $I = \{C_1, \ldots, C_m\}$ of 3SAT, where $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$, $1 \leq i \leq m$, define an instance $G_I = (V_I, E_I)$, B_I of IS as follows: the set V_I of vertices is given by:

$$V_I = \{v_{ij} |\ 1 \leq i \leq m,\ 1 \leq j \leq 3\ \}$$

To construct E_I, connect any two nodes in the same clause, and connect complementary literals (e.g. x and $\bar{x}$). More precisely, $(v_{ij}, v_{kl}) \in E$ iff one of the following holds: $i = k$, that is l_{ij} and l_{kl} are literals in the same clause, OR, l_{ij} and l_{kl} are complementary literals (one is the negation of the other).

Finally, let $B_I = m$.

We have to demonstrate that our reduction is correct. The correctness of the reduction just given can be stated as follows (note that stating the correctness is just saying what the reduction is supposed to do; make sure that you understand this, as in tests you will be asked to state the correctness condition of your reductions—there is nothing complicated about this):

Correctness Condition: I as a YES instance of 3SAT iff $\langle G_I, B_I \rangle$ is a YES instance of IS.

Once you have stated the correctness condition, you have to demonstrate that it holds (note that the correctness condition is given as an "iff" statement; thus, your proof will have two directions):

Proof of the Correctness Condition: ("$\Longrightarrow$") If I has a satisfying truth assignment t, then G_I has an independent set of size m. To see this, select one literal from each clause made true by t. Two complementary literals cannot both be made true by t, so the result is an independent set.

("$\Longleftarrow$") Conversely, if G_I has an independent set V_0 of size m, then I has a satisfying truth assignment t. To see this, observe that V_0 selects one occurrence of a literal from each clause. Define t to make all of these literals true. Note that for each l, we cannot select both l and $\bar{l}$ in different clauses because they are connected by an edge.

This ends the proof of the Lemma that IS is NP-complete. Learn this proof well, as it is a standard proof for showing that problems are NP-complete; you will have to provide similar proofs in tests, assignments, and the final exam. Note that the proof has four well defined components: showing first that the problem is in NP, and then showing that it is NP-hard by giving a reduction from a previously known to be NP-complete problem (and unless it is clear, you have to explain why your reduction is polytime), giving the correctness condition of the reduction, and giving the proof of the correctness condition. $\square$

Vertex Cover

If $G = (V, E)$ and $V_0 \subseteq V$, then V_0 is a *vertex cover* in G iff every edge in E has at least one end point in V_0.

VC:
Input: $G = (V, E)$, $B \in \mathbb{N}$, $B \leq |V|$.
Question: Does G have a vertex cover of size B?
An equivalent question is: Does G have a vertex cover of size at most B?

Related search problem:
VCS: given a graph G, find a smallest vertex cover for G.

Exercise: Show that VCS $\xrightarrow{p}$ VC.

Lemma 3 VC is NP-complete.

PROOF: Exercise. A fat hint: it is easy to show that VC $\in$ NP. We can show that VC is NP-complete by defining a reduction from IS, i.e., IS $\leq_p$ VC (we now know, from the previous Lemma, that IS is NP-complete!). To do that, we need the following result:

Let $G = (V, E)$ be a graph, let $V_0 \subseteq V$. Then V_0 is a vertex cover for G iff $V - V_0$ is an independent set. (Make sure you prove this; it is just an easy observation.)

We define the reduction from IS to VC as follows: given $I = \langle G, B \rangle$, an input to IS, let the corresponding input to VC be $I' = \langle G, B' \rangle$, where $B' = n - B$, $n = |V|$. The correctness of this reduction follows from the result above. $\square$

Clique

If $G = (V, E)$ and $V_0 \subseteq V$, then V_0 is a *clique* in G iff every pair of distinct nodes in V_0 is connected by an edge in E.

CLIQUE:
Input: $G = (V, E)$, $B \in \mathbb{N}$, $B \leq |V|$.
Question: Does G have a clique of size B?
An equivalent question is: Does G have a clique of size at least B?

Related search problem:
CLIQUE-SEARCH: given a graph G, find a largest clique for G.

Exercise: Show that CLIQUE-SEARCH $\xrightarrow{p}$ CLIQUE.

Lemma 4 CLIQUE is NP-complete.

PROOF: Exercise. A fat hint: it is easy to show that CLIQUE $\in$ NP. We will show that CLIQUE is NP-complete by defining a reduction from IS, i.e., IS $\leq_p$ CLIQUE.

Given an instance $I = \langle G, B \rangle$ of IS, we have to find an instance $I' = \langle G_I, B_I \rangle$ of CLIQUE such that $I \in IS$ iff $I' \in CLIQUE$. Let $G_I = \bar{G}$, the complement of G. Let $B_I = B$.

The complement $\bar{G}$ of G is defined as follows. If $G = (V, E)$, then $\bar{G} = (V, \bar{E})$, where $(u, v) \in \bar{E}$ iff $(u, v) \notin E$ (i.e. $\bar{E} = (V \times V) - E$).

The correctness of the reduction above follows from the following result:

G has an independent set of size B iff $\bar{G}$ has a clique of size B.

This is true because V_0 is an independent set in G iff V_0 is a clique in $\bar{G}$. $\square$