How to prove a language is NP-complete

Let us take the example of Clique. A *clique* is a set of k vertices in a graph G such that there is an edge between each pair in the set. You may be asked to formulate the problem as a language:

$$\textsc{Clique} = \{\langle G, k\rangle : G \text{ is a graph with a clique of size } k\}$$

Prove the language is in NP

There are two parts to proving a language is NP-complete. You must show that it is in NP and that it is NP-hard. Do not get so caught up in the second part that you forget the first. For many languages this is simple:

> Clique is in NP because a certificate can be verified in polynomial time. The certificate is the list of the k vertices. To verify it, check each pair of vertices in the list and verify the presence of an edge connecting them.

Sometimes this part is more subtle, but for many cases it can be handled with a brief statement about certificates.

Prove the language is NP-hard

To do this, pick a known NP-complete language to reduce from. In our example, we use Vertex Cover. In a reduction, your task is to convert an instance of the known language (Vertex Cover) into an instance of the unknown language (Clique). Make sure you do not get this backwards. Start by stating that you will be doing a reduction and name the language you will reduce from:

> Clique is NP-hard because Vertex Cover $\leq_p$ Clique, i.e., we can reduce Vertex Cover to Clique in polynomial time.

Next, give the actual construction. Note that this should be a simple description of the transformation; do not try to motivate the steps as you go.

> Given an instance (G, k) of Vertex Cover, we create an instance (G', k') of Clique. G' is constructed to have the same vertices as G, and G' has an edge between two vertices x and y iff there is no edge between x and y in G. k' is $|G| - k$.

At this point it is helpful, but not necessary, to justify the construction informally. A picture like the one we used in lecture with a description of the intuition behind the construction would be helpful. If your construction is correct and precise, however, it will stand on its own.

Next, you need to show that your construction actually achieves the desired reduction. There are a few parts to this.

Show that the construction can be done in polynomial time

A construction that is allowed to take more than polynomial time could solve the problem directly (possibly using exponential time), then output one of two pre-selected instances (one in Clique and one not) depending on the result. This would be a correct reduction, but it would not prove that the language is NP-hard.

> G' can be constructed by running through each of the $< |G|^2$ possible edges that could connect the $|G|$ vertices in G. For each one, check if the corresponding edge exists in G, and if not, add it to G'. Computing k' is straightforward. This shows that the construction can be performed in polynomial time.

Show that the construction yields a correct reduction

This is typically done in two parts:

1. Show that if $(G, k) \in$ Vertex Cover, $(G', k') \in$ Clique. Note that it is not necessary to prove anything about arbitrary Clique instances, only those that result from the construction:

   > If G has a cover of size k, then there must not be any edges connecting a pair of vertices x and y where neither x nor y is part of the cover. If such an edge existed, the cover would not be valid. This implies that G' must have an edge connecting every such pair of vertices from G. There are $|G|-k$ such vertices in G' (by construction), and they make up the $k' = |G|-k$ vertices in the clique of (G', k'). Every such pair lacked an edge in G, and therefore has an edge in G', which is the definition of a clique.

2. Show that if $(G, k) \notin$ Vertex Cover, $(G', k') \notin$ Clique:

   > If G lacks a cover of size k, then by definition every set of $|G| - k$ vertices in G contains at least one edge connecting two vertices within the set. If a set without such an edge exists, then the remaining k vertices must constitute a cover. If $|G| - k < 2$, there cannot be any edges in the set, but in that case $k > |G| - 2$ and a cover of size k must necessarily exist for G, so this case does not apply. If no size $k' = |G| - k$ set of vertices with no edges exists in G, then no size k' set of vertices with edges between every pair exists in G', so $(G', k') \notin$ Clique.

These three parts are all necessary to show that the construction gives a valid polynomial-time reduction from Vertex Cover to Clique. Other variations are possible for the correctness part of the proof:

1. Show that if $(G, k) \in$ Vertex Cover, then $(G', k') \in$ Clique, and if $(G', k') \in$ Clique, then $(G, k) \in$ Vertex Cover.
2. Show that if $(G, k) \notin$ Vertex Cover, then $(G', k') \notin$ Clique, and if $(G', k') \notin$ Clique, then $(G, k) \notin$ Vertex Cover.
3. ...

Just be careful that you do not merely show that if $(G, k) \in$ Vertex Cover, then $(G', k') \in$ Clique, and if $(G', k') \notin$ Clique, then $(G, k) \notin$ Vertex Cover, since the contrapositive is always true and you have not proved your case.

The final step is to conclude by stating what you have proved:

> Clique is in NP and is NP-hard, therefore it is NP-complete.