Two NP-Completeness Exercises

Lane A. Hemaspaandra

2015/2/4 (last revised 2015/2/11/807pm)

1 Exercise 1

VertexCover is a well-known NP-complete problem. In the classic book of Garey and Johnson, they define it this way. (We won’t in this exercise worry at all about details of encodings or syntactically incorrect inputs.)

VertexCover
INSTANCE: An undirected graph \( G = (V, E) \) and a positive integer \( K \leq ||V|| \).
QUESTION: Is there a \textit{vertex cover} of size \( K \) or less for \( G \), that is, a subset \( V' \subseteq V \) with \( ||V'|| \leq K \) and, for each edge \( \{u, v\} \in E \), at least one of \( u \) or \( v \) belongs to \( V' \)?

And here, from the classic book of Garey and Johnson, is the definition of the problem known as HittingSet (which is often used in NP-completeness proofs regarding election systems, by the way!).

HittingSet
INSTANCE: Collection \( C \) of subsets of a set \( S \); positive integer \( K \).
QUESTION: Does \( S \) contain a \textit{hitting set} for \( C \) of size \( K \) or less, that is, a subset \( S' \subseteq S \) with \( ||S'|| \leq K \) such that \( S' \) contains at least one element from each subset in \( C \). (NOTE: \( ||S'|| \) denotes the cardinality of the set \( S' \).)

Prove that HittingSet is NP-complete. You may assume and use the fact that VertexCover is NP-complete. So you need to do two things.

(1) You need to argue that HittingSet is in NP.

(2) Also, though there are various ways of showing it is “NP-hard,” for this exercise, use this specific approach. You should give a many-one polynomial-time reduction from VertexCover to HittingSet. (Hint: Although these problems seem very different—one is about set and one is about graphs—if you think about it and about these problems for a while, you may realize that there is a very simple and correct reduction!) Do remember that after you state your reduction function \( g \), you must discuss three things: (a) that \( g \) is polynomial-time computable, that for each input \( X \), (b) that if \( X \) belongs to VertexCover then \( g(X) \) belongs to HittingSet, and (c) that if \( X \) does not belong to VertexCover then \( g(X) \) does not belong to HittingSet.
2 Exercise 2

Professor Foo says that the following is a correct many-one polynomial-time reduction from HittingSet to VertexCover. “On any input, output $G$ and $K$, where $K$ is 1 and $G$ is the undirected graph that has two vertices, and one edge, namely, it connects those two vertices. This mapping is clearly polynomial-time computable. (In fact, it is CONSTANT-time computable, since it can ignore its input and just output the same thing always.) And for each input that is in HittingSet, we map to a 2-node 1-edge graph and ask if there is some vertex that touches every edge, and indeed there is, and so each instance that is in HittingSet is mapped to an instance that is in VertexCover. Thus, we have shown that HittingSet many-one polynomial-time reduces to VertexCover by the given reduction.” (“By the way,” confides Foo with a wink, “this even proves that HittingSet many-one polynomial-time reduces to the 1-element set that contains just $(G, 1)$, where the $G$ is as above. But since all 1-element sets are in polynomial time, I’ve shown that a polynomial-time set is NP-hard, and so $P = NP$, and so I’ll soon collect the one million dollar prize from the Clay Foundation!”)

Unfortunately, you’ll have to disappoint Professor Foo. In particular, please explain why the given reduction is not a correct many-one polynomial-time reduction from HittingSet to VertexCover. Be very specific: Give a particular input on which the action of this reduction is clearly not the required action, and explain what Foo’s error was. (FYI, correct reductions do exist; but you need to explain why the above is not a correct reduction.)