CPSC 320 Notes: The Futility of Laying Pipe, Part 3

March 19, 2017

First, an apparently unrelated side-track: **How many edges must there be in a tree with n nodes?** So... **if you use at most k edges to form a connected graph, how many nodes can you connect?**

Now, we'll prove that your reduction is correct. Remember: your reduction is correct exactly when the answer to an instance of 3-SAT is `YES` if and only if the answer to the corresponding instance of SP is `YES`. So, your proof should proceed in two steps:

- Assume the answer to the 3-SAT instance is `YES`. Prove that the answer to the SP instance your reduction creates is also `YES`.
- Assume the answer to the SP instance (which your reduction creates from the 3-SAT instance) is `YES`. Prove that the answer to the corresponding 3-SAT instance is also `YES`.

1 If 3-SAT answer is `YES`, SP answer is `YES`

We'll start by assuming that the answer to the 3-SAT instance is `YES` and proving that the answer to the SP instance your reduction creates is also `YES`.

1. We usually want to go further than the assumption that the original instance's answer is `YES` and say that a solution to the orginal instance (i.e., a working certificate) exists.

   Consider what a solution for 3-SAT looks like and use it to finish the statement: "Since the answer to the 3-SAT instance is `YES`, there must be..."

2. To prove the underlying instance's answer is `YES`, we similarly try to show the existence of a solution (working certificate) to that instance and conclude that therefore the answer to the instance is `YES`.

   What does a solution for SP look like?

3. By assumption (because you have a certificate), you now know the truth value of each variable in the 3-SAT instance. What parts of the solution to (i.e., certificate for) the SP instance follow from these truth assignments?

4. By assumption (because your certificate works), every clause in the 3-SAT problem has at least one true literal. What parts of the solution to the SP instance follow from this fact?

5. Finish the proof to show that the SP certificate actually works, i.e., is a solution to the instance.

2 If SP answer is `YES`, 3-SAT answer is `YES`

Now go the other way. Assume that the answer to the SP instance (created by your reduction from the 3-SAT instance) is `YES` and prove that the answer to the 3-SAT instance is also `YES`.

Hints: Many pieces of this proof will be very similar to the previous one, but you'll also need to show that any solution (working certificate) for the SP instance has its edges where you intended them to be when you designed your reduction. It may help to think about the number of nodes you can possibly connect, given a maximum of k edges.

You've now shown that SP is in NP and is NP-hard (because 3-SAT—which is known to be NP-hard—is polynomial-time-reducible to SP). Therefore, SP is NP-complete!

3 Challenge (with two that EVERYONE should try)

1. We didn't actually show that our reduction runs in polynomial time. Briefly sketch a proof that it does. **Everyone should complete this challenge.**

2. Prove that weighted SP is NP-complete. **Everyone should complete this challenge.**

3. The "Euclidean Steiner Tree Problem" simply lists the coordinates of some points and requires connecting those points together with line segments using the shortest total length possible. Give and solve examples with three and four points, being sure to indicate why this is more like SP than like minimum spanning tree.

4. Play around with the simple-to-describe but fiendish Opaque Forest Problem: Given a convex polygon C in the Euclidean (or other??) plane, determine the minimal set of line segments T such that any line that intersects C also intersects T. How is it related to SP? What sort of solutions to this (unsolved!) problem can you find for even very simple shapes?