CS41 Lab 9: NP-Completeness

Thursday, November 3

This week, we've worked hard to understand the notion of problems in NP, escpecially those that are NP-Complete problems. In this lab, we'll look at two additional NPC problems. Recall that to show a problem $A \in \text{NPC}$, it suffices to:

- Prove that $A \in \text{NP}$.
- Choose a problem B known to be NP-Complete.
- Reduce $B \leq_{\text{P}} A$.

During this lab, focus initially on the reductions, and not the formal proofs.

1. Show that 3-Sat $\in$ NPC, by reducing from Sat. Given an instance X of Sat (i.e., a list of n variables and m clauses), you should create an instance Y of 3-Sat (i.e., a list of n' variables and m' clauses, each clause having three literals) such that $Y \in$ 3-Sat iff $X \in$ Sat.

2. In the third exercise, you will show that Three-Coloring is NP-Complete. Before getting there, it will be helpful to create some interesting three-colorable graphs. In all of the following exercises, you are to create a three-colorable graph (say the colors are red, blue, green) with certain special properties. The graphs you create should include three vertices marked a, b, c but can (and often will) include other vertices. Except for the properties specified, these vertices should be *unconstrained*. For example, unless the problem states that e.g. a cannot be red, it must be possible to color the graph in such a way that a is red. (You may fix colors for other vertices, just not a, b, c, and not in a way that constrains the colors of a, b, c.)

   - Create a graph such that a, b, c all have different colors.
   - Create a graph such that a, b, c all have the same color.
   - Create a graph such that a, b, c do NOT all have the same color.
   - Create a graph such that none of a, b, c can be green.
   - Create a graph such that none of a, b, c are green, and they cannot *all* be blue.

3. Show that Three-Coloring $\in$ NPC. Hints: reduce from 3-Sat. Associate the color red with True and the color blue with False.