CS 374 Lab 26: More NP-Completeness

Date: April 27, 2018.

Proving that a problem X is NP-hard requires several steps:

- Choose a problem Y that you already know is NP-hard.
- Describe a reduction f from Y to X, i.e., given input w for problem Y, $f(w)$ is an input to problem X.
- Prove that the function f is computable in polynomial time, by outlining an algorithm running in polynomial time that computes f.
- Prove that your reduction f is correct. This almost always requires two separate steps:
  - Prove that if $w \in Y$ then $f(w) \in X$, i.e., the reduction f transforms "yes" instances of Y into "yes" instances of X.
  - Prove that if $w \notin Y$ then $f(w) \notin X$, i.e., the reduction f transforms "no" instances of Y into "no" instances of X. Equivalently: Prove that if $f(w) \in X$ then $w \in Y$.

Proving that X is NP-Complete requires you to **additionally** prove that $X \in NP$ by describing a non-deterministic polynomial-time algorithm for X. Typically this is not hard for the problems we consider but it is not always obvious.

Problem 1. [Category: Proof] Recall the following kColor problem: Given an undirected graph G, can its vertices be colored with k colors, so that the endpoints of every edge get different colors?

1. Describe a direct polynomial-time reduction from 3Color to 4Color. *Hint:* Your reduction will take a graph G and output another graph G' such that G' is 4-colorable if and only if G is 3-colorable. You should think how an explicit 4-coloring for G' would enable you to obtain an explicit 3-coloring for G.
2. Prove that kColor problem is NP-hard for any $k \geq 3$, by showing that 3Color $\leq_P$ kColor, for $k \geq 3$.

Problem 2. [Category: Proof] Describe a polynomial-time reduction from 3Color to Sat. Can you generalize it to reduce kColor to Sat. *Hint:* Use a variable $x(v, i)$ to indicate that v is colored i and express the constraints using clauses in CNF form.

Problem 3. [Category: Proof] Let $G = (V, E)$ be a directed graph with edge lengths $\ell(e), e \in E$. The lengths can be positive or negative. The Zero-Length-Cycle Problem is to decide whether G has a cycle C of length *exactly* equal to 0. Prove that this problem is NP-Complete. *Hint: reduce Hamiltonian Path to Zero-Length-Cycle*