CS 170 HW 9

Due on 2018-04-02, at 11:59 pm

1 (★) Study Group

List the names and SIDs of the members in your study group.

2 (★) An introduction to 3-SAT

In this homework, we are going to delve deep into the properties of one of the integral problems in computer science, 3-**SAT**. Also, known as 3-*conjunctive normal form satisfiability*, 3-**SAT** is the decision problem of determining if a formula of a specific type has a solution.

A clause is a disjunction (an or) of literals (or their negations). A 3-**SAT** formula must be a conjunction (an and) of clauses with each clause containing *at most* 3 literals.

For example, the following is a 3-**SAT** clause.

$$\varphi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2).$$

It is satisfiable because $x_1 = 1, x_2 = 0, x_3 = 0$ is *a* satisfying assignment. It is not necessarily unique.

Show that 3-**SAT** is in NP.

3 (★★★) Reduction to (3,3)-SAT

As it turns out 3-**SAT** is also NP-complete, meaning that all NP problems have a polynomial time reduction to 3-**SAT**. Let's first explore a simple one, a toy problem called (3,3)-**SAT**. (3,3)-**SAT** is the decision problem of determining whether there is a satisfying assignment for a 3-**SAT** formula in which each literal or its negation appears *at most* 3 times across the entire formula. Notice that (3,3)-**SAT** is reducible to 3-**SAT** because every formula that satisfies the (3,3)-**SAT** constraints satisfies those for 3-**SAT**. We are interested in the other direction.

Show that 3-**SAT** is reducible to (3,3)-**SAT**. By doing so, we will have eliminated the notion that the "hardness" of 3-**SAT** was in the repetition of variables across the formula.

Describe your reduction precisely. Assert the correctness and argue that the reduction can be computed in polynomial time. No pseudocode.

4 (★★★★) Equivalence of Decision and Search

A powerful property of the class NP is the equivalence of decision and search. In the context of 3-**SAT**, it says that *finding* a satisfing assignment for a 3-**SAT** formula is polynomial-time

reducible to *deciding* if one exists. (Convince yourself that the opposite direction is definitionally true.)

Formally, show that Search 3-**SAT** is reducible to Decision 3-**SAT**.

- Search 3-**SAT**. Input: 3-**SAT** formula φ on variables $x_1, x_2, \ldots, x_n$. Output: A satisfying assignment (i.e. $x_1 = \alpha_1, x_2 = \alpha_2, \ldots$) if one exists. Othewise, $\perp$.
- Decision 3-**SAT**. Input: 3-**SAT** formula φ on variables $x_1, x_2, \ldots, x_n$. Output: 1 if a satisfying assignment exists and otherwise, 0.

Describe your reduction precisely. Assert the correctness and argue that the reduction can be computed in polynomial time. No pseudocode.

Hint: Assume you have an algorithm $\mathcal{O}$ that can solve Decision 3-**SAT**. Then given an input to Search 3-**SAT**, make $O(n)$ black-box queries to $\mathcal{O}$.

5 (★★★★★) Reduction to 3-Coloring

One of the beautiful properties of NP-complete problems is that they span most branches of theoretical computer science. For example, 3-**SAT**, as we have seen, is about boolean formulas; Traveling Salesman is about paths in graphs; Integer Programming is about optimization; the list goes on. Richard Karp's 1972 paper, *Reducability among Combinatorial Problems* laid the groundwork for this field by demonstrating 21 different problems which were all polynomial time reducible to 3-**SAT**. Today, we will explore one of these, namely 3-Coloring. The problem is stated as follows:

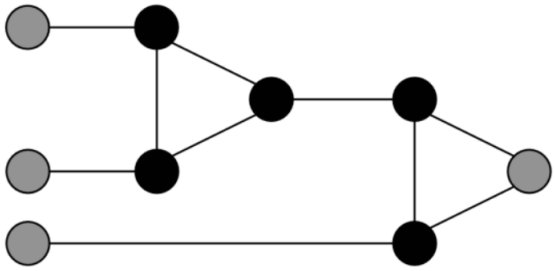
Input: A graph $G = (V, E)$. Output: 1 if there exists an assignment $\mathcal{C} : V \to \{0, 1, 2\}$ of one of three colors to each of the vertices such that no adjacent vertices are the same color if such an assignment exists. Otherwise, 0.

Recall, that vertices are adjacent if they share an edge. Provide a reduction from 3-**SAT** to 3-Coloring.

Describe your reduction precisely. Assert the correctness and argue that the reduction can be computed in polynomial time. No pseudocode.

Hint: Given a 3-**SAT** formula φ on literals $x_1, \ldots, x_n$, the goal is to produce a graph G. Your graph will consist of $\geq 2n + 3$ vertices where the $2n + 3$ vertices are: one vertex corresponding to each literal x_i and one corresponding to each literal $\neg x_i$ and three additional vertices that will form a triangle. Since, they form a triangle, they must each be colored differently if a 3-Coloring exists. Furthermore, one of the colors will represent true, i.e. 1. Another color will represent false, i.e. 0.

In addition, the following "gadget" will be useful. In the following graph,



if each of the grey nodes are colored with either 0 or 1, then it is possible to 3-Color the entire graph if and only if at least one of the grey nodes on the left has the same color as the one on the right. [In your proof, you make take this statement for granted without proving it explicitly.]