Problem 1: Power of Satisfiability

In class we proved that any NP problem can be reduced to Satisfiability (SAT). We did so by giving a series of tedious reductions beginning from WHILE-SAT and ending to CIRCUIT-SAT. Usually, when we start from a specific combinatorial problem, the reduction is much more straightforward and intuitive. This problem will hopefully give you more insight as to why SAT can be used to easily model any other efficient computation.

Consider the following problem: You are given a graph $G = (V, E)$. An assignment $a : V \rightarrow \mathbb{N}^1$ is called valid if $a(v) \neq a(u)$ $\forall u, v$ such that $(u, v) \in E$ (that is, the assignment should assign different numbers to adjacent nodes). Your goal is to (validly) assign to every node a number while using as few numbers as possible.

(a) Formulate the above problem as a decision problem (let’s call $GN_D$ this decision version). The decision problem should be equivalent to the search problem we just defined, in the sense that they can be related (in both ways) via polynomial time Turing reductions. However, you are not required to give such reductions here. (This point is further explored in part (c).) For part (a), just define an appropriate decision problem.

(b) Give a polynomial time (map) reduction from $GN_D$ to SAT. In particular describe how an instance of $GN_D$ can be transformed to a CNF formula $\phi$. State explicitly which and how many variables you are using in $\phi$. Prove that your transformation can be carried out in polynomial time and that it correctly reduces $GN_D$ to SAT.

(c) Using parts (a) and (b), explain how you would go about finding a valid assignment $a$ in your initial graph $G$ that uses the minimum possible number of integers, using an oracle for SAT. (In other words, give a polynomial time Turing reduction from the problem of finding an assignment for the graph to the decision problem SAT.)

Problem 2: NP Completeness

In the previous problem set we defined the following problem (Problem B) where instances consist of

- finite set $U = \{u_1, u_2, ..., u_n\}$
- a collection (or set) of subsets of $U$, $T = \{T_1, T_2, ..., T_m\}$ where $T_i \subseteq U$ $\forall i \in \{1, ..., m\}$ and
- a positive integer $k$

Decision Problem: Is there a subset $T' \subseteq T$ with $|T'| \leq k$ such that the union of the elements in $T'$ equals $U$?

- Give a map reduction (in pseudocode) from SAT (or 3-SAT) to B.
- Prove that your reduction can be performed in time polynomial in the size of the input and prove its correctness.
- Show that B is in NP and conclude that B is NP-Complete.

Note: You may use the fact that problem A, as defined in the previous problem set, reduces to problem B. You can therefore reduce SAT (or 3-SAT) to A and then use the previous problem set to conclude that SAT (or 3-SAT) $\leq_P$ B.

1For ease, you may assume that numbering starts at 1 and that the assignment uses consecutive integers.