Assignment 11

1 Search and Decision Problems

As discussed in class, NP is a class of decision problems, i.e., the answer is either "yes" or "no". This choice is convenient and often also captures the difficulty of searching for a solution. In our class, we have shown the reduction from search to decision for some problems. Here, you will show the search-to-decision reduction for two more examples.

(1) **3-SAT**: A 3-SAT formula consists of m clauses, each of which is a disjunction of three literals (where a literal is one of n variables or its negation). We say a 3-SAT formula F is satisfiable if there is an assignment of the n variables such that F evaluates to true. The 3-SAT problem asks us to decide whether a given F is satisfiable.

Suppose you are given a black box for a function 3SAT that determines whether a given 3-SAT formula is satisfiable with timing cost T_{3SAT}. Design an algorithm that finds a satisfying assignment for F (assuming it is satisfiable) using $O(n)$ calls to 3SAT and polynomially many other steps. Prove that your algorithm is correct and analyze its running time.

(2) **3-Coloring**: We say an undirected graph is 3-colorable if there is an assignment of the colors $\{r, g, b\}$ to the vertices (a coloring) such that no two adjacent vertices have the same color. The 3-Coloring problem asks us to decide whether a given graph is 3-colorable.

Suppose you are given a black box for a function 3Color that determines whether a given graph is 3-colorable with timing cost T_{3Color}. Design an algorithm that finds a coloring of a given graph using $O(n)$ calls to 3Color and polynomially many other steps, where n is the number of vertices in the input graph. Prove that your algorithm is correct and analyze its running time.

2 3-Dimensional Matching

In our class, we introduced how to solve the bipartite matching problem in polynomial time. In particular, the perfect bipartite matching problem can be written in the following way: Suppose we are given two disjoint sets A and B, each of size n, and a set P of pairs drawn from $A \times B$. The goal is to determine whether there exists a set of n pairs in P such that each element in $A \cup B$ is contained in exactly one of these pairs.

As a generalization, consider the following 3-dimensional matching problem: Suppose we are given three disjoint sets A, B, and C, each of size n, and a set T of triples drawn from $A \times B \times C$. The goal is to determine whether there exists a set of n triples in T such that each element in $A \cup B \cup C$ is contained in exactly one of these triples.

(1) Prove that 3-dimensional-matching $\leq_P$ Set-Cover. This implies that 3-dimensional-matching $\in$ NP.

(2) Prove that 3-SAT $\leq_P$ 3-dimensional-matching. Together with (1), this implies that the 3-dimensional-matching problem is NP-complete.

3 Restricted Monotone Satisfiability

Consider an instance of the Satisfiability problem, specified by clauses $C_1, \ldots, C_k$ over a set of Boolean variables $x_1, \ldots, x_n$. We say that the instance is *monotone* if each literal is a non-negated variable (i.e., x_i can appear as a literal but $\neg x_i$ cannot). Monotone instances are always satisfiable since we can simply set each variable to 1.

For example, suppose we have the three clauses

$$(x_1 \vee x_2), (x_1 \vee x_3), (x_2 \vee x_3)$$

This is monotone, and indeed the assignment that sets all three variables to 1 satisfies all the clauses. But we can observe that this is not the only satisfying assignment; we could also have set x_1 and x_2 to 1, and x_3 to 0. Indeed, for any monotone instance, it is natural to ask how few variables we need to set to 1 in order to satisfy it.

Given a monotone instance of Satisfiability, together with a number k, the problem of *Restricted Monotone Satisfiability* asks: is there a satisfying assignment for the instance in which at most k variables are set to 1? Prove this problem is NP-complete.

4 Do some Calculus!

For functions $g_1, \ldots, g_\ell$, we define the function $\max(g_1, \ldots, g_\ell)$ via

$$[\max(g_1, \ldots, g_\ell)](x) = \max(g_1(x), \ldots, g_\ell(x))$$

Consider the following problem. You are given n *piecewise linear, continuous* functions $f_1, \ldots, f_n$ defined over the interval $[0, t]$ for some integer t. You are also given an integer B. You want to decide: do there exist k of the functions $f_{i_1}, \ldots, f_{i_k}$ so that

$$\int_0^t [\max(f_{i_1}, \ldots, f_{i_k})](x)\, dx \geq B?$$

Prove that this problem is NP-complete.