CS 473: Fundamental Algorithms, Fall 2014

Discussion 12

November 18 / 19, 2014

12.1. From Set Cover to Montone SAT

Consider an instance ϕ of a CNF formula specified by clauses $C_1, C_2, \ldots, C_k$ over a set of boolean variables $x_1, x_2, \ldots, x_n$. We say ϕ is *monotone* if each term in each clause consists of a nonnegated variable; that is, each term is equal to x_i and not $\bar{x}_i$ - negations are not allowed. They could be easily satisfied by setting each variable to 1.

For example, the clause

$$\phi(x_1, x_2, x_3) = (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_3 \vee x_2)$$

could be satisfied by setting all thre variables to 1, or by setting x_1 and x_2 to 1 and x_3 to 0.

Given a monotone instance ϕ of a CNF formula and an integer $k \in \mathbf{N}$, the *Monotone Satisfiability* problem asks whether there is a satisfying assignment for the instance in which at most k variables are set to 1.

The *Set Cover* decision problem asks, given a collection F of subsets $S_1, S_2, \ldots, S_m$ of a ground set $U = \{1, \ldots, n\}$, and an integer k, whether U can be covered by k sets in F (i.e., whether there are k sets in F whose union is U).

Give a Karp reduction from the Set Cover decision problem to Monotone Satisfiability.

12.2. From 3 to 42!

The *3-Coloring* problem asks whether the vertices of a given graph G can be colored by three colors such that any two adjacent vertices have different colors. This problem is NP-Complete.

(a) The 42-Coloring coloring problem asks, given a graph G, whether the vertices of G can be colored by 42 different colors such that any two adjacent vertices are colored differently. Prove that the 42-Coloring problem is NP-Complete. *[Hint: Reduce from 3-Coloring.]*

(b) The 42-SAT problem asks, given a boolean formula ϕ in conjunctive normal form with exactly 42 literals per clause, determine whether ϕ has a satisfying assignment. Prove that the 42-SAT problem is NP-Complete. *[Hint: Reduce from 3SAT.]*

12.3. Self-Reduction for k-Coloring

In the *k-Coloring* problem, we are given a graph G and an integer k and have to decide whether the vertices of G can be colored by k-different colors such that no two adjacent vertices are colored the same.

Suppose we have a black-box that can answer the k-Coloring decision problem (where k is fixed). Design an algorithm that uses only a polynomial number of calls to the black-box that either generates a k-Coloring of a graph or determines that a k-Coloring does not exist.

12.4. In the Hitting Set Problem, we are given a collection C of subsets of a finite set S. That is, $C = \{C_1, ..., C_n\}$, where for each i we have $C_i \subseteq S$. We are also given a parameter k, and we want know whether there exists a subset S' of S such that S' contains at least one element from each C_i and $|S'| = k$. Prove that this problem is NP-Complete.