Theory of Computation

Midterm Examination on December 2, 2021
Fall Semester, 2021

Problem 1 (20 points) It is well-known that 3SAT is NP-complete. 1-IN-3 SAT asks if each clause contains exactly one true literal in a truth assignment. Prove that 1-IN-3 SAT is NP-Complete.

Proof: First, 1-IN-K SAT is in NP because a random guess of truth assignment ϕ can be verified in polynomial time. We then reduce 3SAT to 1-IN-K SAT as follows. For each clause $p \vee p' \vee p''$, create three clauses $\neg p \vee a \vee b$, $p' \vee b \vee c$, and $\neg p'' \vee c \vee d$. Call this new instance ϕ' as the input of 1-IN-3 SAT. Clearly, ϕ is satisfiable if and only if ϕ' is 1-in-3 satisfiable. So 1-IN-3 SAT is NP-complete. ■

Problem 2 (20 points) Given a graph $G(V, E)$ with a positive integer K, 2CLIQUE asks if there are two disjoint sets $C_1, C_2 \subseteq V$, each with exact K nodes, such that both are cliques. It is clear that 2CLIQUE is in NP. Prove that 2CLIQUE is NP-complete.

Proof: We only need to show that 2CLIQUE is NP-hard by reducing CLIQUE to 2CLIQUE. The reduction can be done by adding a complete graph (i.e., clique) of size K together with the input G, called G'. Clearly, G has a clique if and only if G' has two disjoint cliques. The claim holds. ■

Problem 3 (20 points) A cut in an undirected graph $G = (V, E)$ is a partition of the nodes into two nonempty sets S and $V - S$. MAX BISECTION asks if there is a cut so that $|S| = |V - S|$. Prove that MAX BISECTION is NP-complete by completing the argument in the lecture notes.

Proof: See pp. 413-414 of the slides. ■

Problem 4 (20 points) EDGE COVERING asks if there exists a set of edges C such that each vertex in G is incident with at least one edge in C. Prove that EDGE COVERING is NP-complete.

Proof: We first show that EDGE COVERING is in NP: guess a set of edges C and then check if C covers all vertices. It can be done in polynomial time. We proceed to reduce INDEPENDENT SET to EDGE COVERING. On the input graph $G(V, E)$ with the vertex set S, C collects all adjacency edges of all vertices in S. This reduction can be done in polynomial time. Clearly, S is an independent set of G if and only if C is an edge cover of G. Hence EDGE COVERING is NP-hard. ■

Problem 5 (20 points) IP asks whether a system of linear inequalities with integer coefficients has an integer solution. Reduce 3SAT to IP to show that IP is NP-hard.

Proof: The reduction can be done as follows. Add $0 \leq x_i \leq 1$ for each boolean variable x_i in the 3SAT formula ϕ. For each clause $\alpha \vee \beta \vee \gamma$, create the inequality $x_\alpha + x_\beta + x_\gamma \geq 1$, where a boolean variable x_i is mapped to x_i and a negated boolean variable $\neg x_i$ is mapped to $(1 - x_i)$. Clearly, ϕ is satisfiable if and only if the resulting system of linear inequalities has an integer solution. ■