Theory of Computation

Midterm Examination on November 29, 2016
Fall Semester, 2016

Problem 1 (25 points) Show that for $n > 3$, n-SAT is NP-complete. (You don't need to show that n-SAT is in NP.)

Proof: We reduce 3-SAT to n-SAT as follows. Let ϕ be an instance of 3-SAT. For any clause $(a \vee b \vee c)$ of ϕ, replace it with $(a \vee b \vee \underbrace{c \vee \cdots \vee c}_{n-2})$. By repeating this procedure for all clauses of ϕ, we derive a new boolean expression ϕ' for n-SAT. Then ϕ is satisfiable if and only if ϕ' is satisfiable. ∎

Problem 2 (25 points) Let $G = (V, E)$ be a graph and K be a positive integer. LONGEST PATH ask if there is a simple path which contains at least K edges in G. Show that LONGEST PATH is NP-complete. (You need to show that LONGEST PATH is in NP.)

Proof: First we show that LONGEST PATH is in NP. Given an instance G, we guess a set of edges of size at least K and at most $|G|$ and examine if it is a simple path in G. This can be done in polynomial time. We now proceed to show that LONGEST PATH is NP-hard by reducing HAMILTONIAN PATH to LONGEST PATH. Given an instance G of HAMILTONIAN PATH, we create an instance (G', K) of LONGEST PATH as follows: Take $G' = G$ and set $K = |V| - 1$. Then there exists a simple path of length K in G' if and only if G contains a Hamiltonian path. ∎

Problem 3 (25 points) Prove that the language Ψ is NP-complete, where

$$\Psi = \{(N, x, 1^t) \mid \text{a nondeterministic Turing Machine } N \text{ that accepts } x \text{ within time } t\}.$$

Recall that 1^k denotes the string consisting of k 1s. Do not forget to show Ψ is in NP.

Proof: We first show that Ψ is in NP. With the input $(N, x, 1^t)$, we simulate N nondeterministically on x up to t steps and accept if N accepts x. The algorithm obviously runs in polynomial time. Furthermore, $(N, x, 1^t)$ is in Ψ if and only if there is a path such that $N(x) =$ "yes" within t steps. We next show that Ψ is NP-hard. Let $L \in$ NP be accepted by a nondeterministic Turing Machine N that runs in polynomial time n^c for some constant c. To reduce L to Ψ, simply map the input x to the triple $(N, x, 1^{n^c})$. The reduction can evidently be performed in polynomial time. It is clear that $x \in L$ iff $(N, x, 1^{n^c}) \in \Psi$.

∎

Problem 4 (25 points) DNF NON-TAUTOLOGY asks if a DNF is *not* a tautology. Prove that this problem is NP-complete. (You need to show that DNF NON-TAUTOLOGY in NP.)

Proof: The problem is equivalent to asking if there exists a truth assignment that makes the DNF false. This problem is in NP because one can nondeterministically guess a truth assignment and accept the input DNF formula if it is not satisfied by the truth assignment. We shall reduce the NP-complete SAT to it. The reduction applies de Morgan's laws to convert the input CNF formula ϕ into a DNF ψ of about the same length. Then ϕ is satisfiable if and only if ψ is not a tautology. ■