Theory of Computation

Homework 3

Qing-Cheng Li
R01922024

November 19, 2012

1 Problem 1

We can write a nondeterministic polynomial-time algorithm which takes a $D{-}SAT$ instance and 2 proposed truth assignments as input, if the $D{-}SAT$ instance evelates 2 assignments to true, the algorithm outputs *yes*; otherwise outputs *no*. This runs in polynomial time, so $D-SAT$ is in *NP*. We reduce SAT to $D-SAT$ as follows. Let ϕ donates an instance of SAT, we add a new variable y, convert ϕ to $\phi' = \phi \wedge (y \vee \neg y)$, a $D-SAT$ instance. If ϕ has at least 1 statisfying assignment, then $\phi \wedge (y \vee \neg y)$ is true for $y = 1$ or $y = 0$, so ϕ' has at least 2 statisfying assignments. If ϕ has no statisfying assignment, then ϕ' is also has no statisfying assignment. So $D-SAT$ is *NP-Complete*.

2 Problem 2

We can write a nondeterministic polynomial-time algorithm which takes a undirected graph G and nodes n_s, n_e as input, nondeterministically choose path from n_s to n_e, if this path is a Hamiltonian path, this algorithm outputs *yes*, otherwise outputs *no*. This runs in polynomial time, so $SE-HamiltonianPath$ is in *NP*. We reduce $HamiltonianCycle$ to $SE{-}HamiltonianPath$ as follows. Given a undirected graph G, we can choose a node n from G, let this node n be n_s and n_e in $SE-HamiltonianPath$. If this graph G has a Hamiltonian Cycle, then there is a Hamiltonian Path from n to n on G. If G has no Hamiltonian Path from n to n, G has no Hamiltonian Cycle. So $SE-HamiltonianPath$ is *NP-Complete*.