CS 331: Theory of Computing

Problem Set 10

Iowa State University
Computer Science Department
April 18, 2013

Due by the midnight of April 25, 2013

Problem 1 (20 points)

Let G be undirected and connected graph. A graph is said to be connected if every pair of vertices in the graph are connected by a path. A path is called an Eulerian path if it starts and ends at a same vertex and all edges in G appear on the path exactly once. The Eulerian path problem is defined as follows.

$$\mathrm{EULERIAN{-}PATH} = \{ \langle G \rangle \mid G \textit{ has an Eulerian path} \}$$

Prove that $\mathrm{EULERIAN{-}PATH}$ is in P.

Hint: Prove that G has an Eulerian path if and only if every vertex in G has even number of degree, i.e., every vertex is in touch with an even number of edges. One direction is easy. Use induction to prove the other.

Problem 2 (20 points)

Let G be undirected graph whose every edge is associated with an integer (length). The **Traveling Salesman Problem** is defined as follows.

$$\mathrm{TSP} = \{\ \langle G, k\rangle \mid G \textit{ has a Hamiltonian path of length less than } k\ \}$$

Prove that TSP is *NP*-complete.

Hint: Prove that $\mathrm{HAMILTON{-}PATH} \leq_P \mathrm{TSP}$.

Problem 3 (20 points)

A **disqualifier** for a language $\mathscr{L}$ is DTM D, where

$$\mathscr{L} = \{ w \mid D \textit{ accepts } \langle w, e \rangle \textit{ for some word } e \}.$$

D is **polynomial time disqualifier** if D runs in polynomial time in the length of w. A language $\mathscr{L}$ is **polynomially disqualifiable** if it has a polynomial time disqualifier.

Prove that a language is in *coNP* iff it has a polynomial time disqualifier.

Problem 4 (20 points)

A language $\mathscr{L}$ is *coNP*-**complete** if

- $\mathscr{L} \in coNP$, and
- for any language $\mathscr{L}' \in coNP$, $\mathscr{L}' \leq_P \mathscr{L}$.

Prove the following statements.

- (10 points) The class *coNP* is closed under polynomial-time reductions; that is, if $\mathscr{L}_1 \leq_P \mathscr{L}_2$ and $\mathscr{L}_2 \in coNP$, then $\mathscr{L}_1 \in coNP$.
- (10 points) If a *coNP*-complete language $\mathscr{L}$ is in *NP*, then $coNP = NP$.

Problem 5 (20 points)

Let $P^{\mathscr{L}}$ be the class of languages recognized by polynomial time oracle Turing machines with an oracle for the language $\mathscr{L}$. Let $\mathcal{C}$ be a class of languages. Define $P^{\mathcal{C}} = \bigcup_{\mathscr{L} \in \mathcal{C}} P^{\mathscr{L}}$; that is, $P^{\mathcal{C}}$ is the class of languages recognized by polynomial time oracle Turing machines that use oracles for languages in $\mathcal{C}$.

Prove the following statements.

- (5 points) For any language class $\mathcal{C}$, $P^{\mathcal{C}}$ is closed under complementation; that is, $\mathscr{L} \in P^{\mathcal{C}}$ iff $\overline{\mathscr{L}} \in P^{\mathcal{C}}$.
- (5 points) For any language class $\mathcal{C}$ that is closed under complementation, $\mathcal{C} \subseteq coNP$ iff $\mathcal{C} \subseteq NP$.
- (5 points) $P^{P} \subseteq P$.
- (5 points) $NP^{P} \subseteq NP$.