CPSC 421/501 Intro to Theory of Computing (Term 1, 2012-13)
Assignment 5

Due: Wednesday, November 28th in class.

For any question (except the bonus question), if you write "I do not know the answer to this question", you will receive 20% of the marks for that question.

Question 1: Define the following computational problem. Let $x_1, ..., x_n$ be real variables. Let

$$f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n)$$

be polynomials, each of degree at most n. Let $C_1, ..., C_n \subseteq \mathbb{Z}$ be finite sets, each of size at most n. We wish to determine whether there exist values $\tilde{x}_i \in C_i$ for all $i = 1, ..., n$ such that

$$f_j(\tilde{x}_1, ..., \tilde{x}_n) = 0 \qquad \forall j = 1, ..., m.$$

The language $ROOTS$ consists of the encodings $\langle n, f_1, ..., f_m, C_1, ..., C_n \rangle$ of problem instances for which the desired $\tilde{x}_i$ values exist. Prove that $ROOTS$ is NP-hard.

Question 2: Define the following computational problem. Suppose there are n vitamins that one must ingest every day. (For simplicity, let's call those vitamins $v_1, ..., v_n$.) Suppose there are m different dinners that one could possibly eat, and that each dinner contains some subset of the vitamins. (We do not care about the amount of the vitamins in a dinner; we simply assume that the vitamin is either present or not present.) That is, each dinner corresponds to a subset $D_i \subseteq \{v_1, ..., v_n\}$. Given an integer k, we would like to know if there are k dinners which together contain all the required vitamins. (It is OK for the same vitamin to appear in multiple dinners.)

The language $DINNERS$ consists of all encodings $\langle n, k, D_1, ..., D_m \rangle$ of problem instances for which there exist k dinners that together contain all the vitamins. Prove that $DINNERS$ is NP-complete.

Question 3: Say that language A is in $IP_{1,\epsilon}[k]$ if some polynomial time function V (the verifier) and an arbitrary function P (the prover) exist, where for every function $\tilde{P}$ and string w

1: $w \in A$ implies $\Pr[V \leftrightarrow P \text{ accepts } w] = 1$, and

2: $w \notin A$ implies $\Pr\left[V \leftrightarrow \tilde{P} \text{ accepts } w\right] \leq \epsilon$

and the protocol proceeds for exactly k rounds. (I.e., k messages are sent between the prover and verifier.) Recall the definition:

$$NONISO \;=\; \{\, \langle G, H \rangle \,:\, G \text{ and } H \text{ are non-isomorphic graphs} \,\}.$$

Prove that, for every constant $\epsilon > 0$, $NONISO \in IP_{1,\epsilon}[2]$.