CPSC 421/501 Intro to Theory of Computing (Term 1, 2012-13)
Assignment 4

Due: Friday November 16th, 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: Describe the error in the following incorrect "proof" that $P \neq NP$. Assume towards contradiction that $P = NP$. If $P = NP$ then $SAT \in P$, so for some k, $SAT \in TIME(n^k)$. Because every language in NP is polynomial time reducible to SAT, you have $NP \subseteq TIME(n^k)$. Therefore $P \subseteq TIME(n^k)$. But, by the time hierachy theorem, $TIME(n^{4k})$ contains a language that isn't in $TIME(n^k)$, which contradicts $P \subseteq TIME(n^k)$. Therefore $P \neq NP$.

Question 2: For each of the following, determine whether such a problem is: known to exist, known not to exist, or neither. If you answer "neither", explain why discovering such a problem would make you rich. Otherwise, briefly justify your answer.

(a): An NP-complete problem which can be solved in exponential time.
(b): An NP-complete problem which cannot be solved in exponential time.
(c): An NP-hard problem which cannot be solved in exponential time.
(d): A problem in NP that is not NP-complete.

Question 3: Define

$$coNP \;=\; \{\, L : \bar{L} \in NP \,\},$$

where $\bar{L}$ denotes the complement of the language L. Suppose $L_1, L_2 \in NP \cap coNP$. Define

$$L_1 \oplus L_2 \;=\; \{\, x : x \text{ is in exactly one of } L_1, L_2 \,\}.$$

Prove that $L_1 \oplus L_2 \in NP \cap coNP$.

Question 4: Suppose that

- $A \subseteq \Sigma^*$ is NP-complete,
- $B \subseteq \Sigma^*$ is in P,
- $A \cap B = \emptyset$, and
- $A \cup B \neq \Sigma^*$

Prove that $A \cup B$ is NP-complete.

Question 5: Suppose that $P = NP$. Show that there exists a polynomial time algorithm that, given any 3-colorable graph, produces a valid 3-coloring.

Note: The algorithm you are asked to provide computes a function, but NP contains languages (i.e., decision problems), not functions. The $P = NP$ assumption implies that we can decide whether a graph is 3-colorable can be done in polynomial time. But that assumption doesn't say how this test is done, and the test might not reveal any satisfying assignments. You must show that you can find them anyways.

OPTIONAL BONUS QUESTION:

Question 6: Prove that $P = NP$ if and only $\exists k, \ell \geq 100$ such that $NTIME(n^k) \subseteq TIME(n^\ell)$.