CPSC 421/501 Intro to Theory of Computing (Term 1, 2013-14)
Assignment 5

Due: Monday November 4th, in class.

Question 1: [10 marks]

Let us consider a problem that generalizes Sudoku. (In the case $n = 3$, this is ordinary Sudoku.) A problem instance consists of a two-dimensional grid of cells, with n^2 rows and n^2 columns. In the initial problem instance, each cell is either blank or contains a number in $\{1, \ldots, n^2\}$.

The goal is to place a number into every blank cell such that:

- (1) Each column contain every number in $\{1, \ldots, n^2\}$ exactly once.
- (2) Each row contain every number in $\{1, \ldots, n^2\}$ exactly once.
- (3) For every $i, j \in \{0, \ldots, n-1\}$, the square at the intersection of rows $\{ni+1, \ldots, n(i+1)\}$ and columns $\{nj+1, \ldots, n(j+1)\}$ contains every number in $\{1, \ldots, n^2\}$ exactly once.

The computational problem of interest is: given an initial problem instance (in which each cell could be blank or contain a number), decide whether the blanks can be filled in such that conditions (1)-(3) are satisfied.

Show that this computational problem is in NP.

Question 2: [10 marks]

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 3: [10 marks]

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 QUESTIONS:

Question 4: [10 marks]

Motivated by Nancy's question in class, let's consider a different type of reduction. We say that A is **unusually reducible** to B if there exists a TM M that

- is nondeterministic
- runs in polynomial time
- has access to a subroutine that decides B
- has $L(M) = A$.

If this holds, then we write $A \leq_U B$.

Let us say that a language $L \in NP$ is **unusually NP-complete** if $A \leq_U L$ for all $A \in NP$. Prove that **every** $L \in NP$ is unusually NP-complete.

So "unusual NP-completeness" is not a very interesting concept.

Question 5: [20 marks]

Let us say that a boolean formula is a "four-occurrence CNF formula" if it is in conjunctive normal form and every variable appears at most four times. Define

$$CNF_4 = \{ \langle \phi \rangle : \phi \text{ is a satisfiable, four-occurrence CNF formula} \}.$$

It is known that CNF_4 is NP-complete.

Let us say that a boolean formula is a "four-occurrence 4CNF formula" if it is in conjunctive normal form, every variable appears at most four times, and every clause contains exactly four literals (no repetitions). Define

$$4CNF_4 = \{ \langle \phi \rangle : \phi \text{ is a satisfiable, four-occurrence 4CNF formula} \}.$$

Prove that $4CNF_4$ is in P.