Theory of Computation

Mid-Term Examination on November 6, 2012

Fall Semester, 2012

Note: You may use any results proved in class.

Problem 1 (25 points) It is known that 3-SAT is NP-complete. Show that 4-SAT is NP-complete. (Don't forget to show that it is in NP.)

Ans: To show that 4-SAT is NP-complete, we prove that 4-SAT is in NP and NP-hard.
First, 4-SAT is in NP, we can write a nondeterministic polynomial-time algorithm which takes a 4-SAT instance and a proposed truth assignment as input. This algorithm evaluates the 4-SAT instance with the truth assignment. If the 4-SAT instance evaluates to true, the algorithm outputs *yes*; otherwise, the algorithm outputs *no*. This runs in polynomial time.
To prove that 4-SAT is NP-hard, we reduce 3-SAT to 4-SAT as follows. Let ϕ denote an instance of 3-SAT. We convert ϕ to a 4-SAT instance ϕ' by turning each clause $(x \vee y \vee z)$ in ϕ to $(x \vee y \vee z \vee h) \wedge (x \vee y \vee z \vee \neg h)$, where h is a new variable. Clearly this is polynomial-time doable.

$\Rightarrow$ If a given clause $(x \vee y \vee z)$ is satisfied by a truth assignment, then $(x \vee y \vee z \vee h) \wedge (x \vee y \vee z \vee \neg h)$ is satisfied by the same truth assignment with h arbitrarily set. Thus if ϕ is satisfiable, ϕ' is satisfiable.

$\Leftarrow$ Suppose ϕ' is satisfied by a truth assignment T. Then $(x \vee y \vee z \vee h) \wedge (x \vee y \vee z \vee \neg h)$ must be true under T. As h and $\neg h$ assume different truth values, $x \vee y \vee z$ must be true under T as well. Thus ϕ is satisfiable.

■

Problem 2 (25 points) Show that if there exists a language $L \in$ NP not in P, then no NP-complete language is in P.

Ans: Suppose $L \in$ NP, $L \notin$ P. Now, if there is an $L' \in$ P which is NP-complete, then $L \in$ NP can be reduced to L', and hence $L \in$ P, a contradiction. ■

Problem 3 (25 points) Show that $L \neq P$ or $P \neq PSPACE$.

Ans: Suppose $L = P$ and $P = PSPACE$ instead. Then $L = PSPACE$. However, we know these two classes are different by the space hierarchy theorem, a contradiction. ■

Problem 4 (25 points) Show that $\{M : M \text{ halts on all inputs}\}$ is not recursive.

Ans: We reduce halting problem to this problem. Given $M; x$, we construct the following machine M':

$$M'(y): \text{ if } y = x \text{ then } M(x) \text{ else halt.}$$

Obviously, M' halts on all inputs if and only is M halts on x. ■