Assignment 1 **CAS705 Fall 2006**

Due on September 29
In class, at the beginning of the lecture

The work you submit must be your own. *You may discuss problems with each other; however, you should prepare written solutions alone. In particular, you should not leave with any written notes from such discussions. The style and clarity of your answers will be an important factor in the grade.*

Each question is worth 25%.

1. Let α be a 3CNF formula. A nae-assignment (nae=not all equal) to the variables of α is an assignment where each clause contains two literals with unequal truth values (i.e., one true and one false).

   (a) Show that the negation of any nae-assignment is also a nae-assignment ("negation" means that true and false are interchanged for all variables).

   (b) Let NAE3SAT be the set of 3CNF formulas with nae-assignments. Show that 3SAT $\leq_m$ NAE3SAT, and conclude that NAE3SAT is **NP**-complete.

2. Show that the problem DEG2POLY given by:

   - **Input:** m polynomials of degree 2 in n variables.
   - **Question:** Do they have a common zero?

   is **NP**-hard.

   Hint. 3SAT is **NP**-complete, and given a 3CNF formula we can convert it into a system of polynomials as follows: for each variable x_i that appears, add the polynomial $x_i^2 - x_i$, to ensure that only values $\{0, 1\}$ are taken. Then, for each clause $(l_1 \vee l_2 \vee l_3)$ add the polynomial $m(l_1) \cdot m(l_2) \cdot m(l_3)$, where $m(l) = (1 - x)$ if $l = x$ and $m(l) = x$ if $l = \bar{x}$. The resulting system has a common zero iff the original 3CNF formula is satisfiable. However, the problem is that the polynomials may be of degree 3. Use NAE3SAT instead (see previous question).

3. Consider languages over $\Sigma = \{0, 1\}$. Suppose that we have a set $T \subseteq \{1\}^*$ (i.e., T is a set consisting of strings of 1s) that is **NP**-hard (with respect to polynomial-time, many-one reducibility). Show that in that case $\mathbf{P} = \mathbf{NP}$.

   Hint. We know that SAT $\leq_m T$ (why?). So we have a polytime function g, such that $\alpha \in \text{SAT} \iff g(\alpha) \in T$. Now use g to give a polytime algorithm for deciding SAT. Note that SAT is "self-reducible" in the following sense: if $\alpha = \alpha(v_1, v_2, \ldots, v_n)$, i.e., $v_1, v_2, \ldots, v_n$ is the complete list of all the variables of α, then for any v_i, $\alpha \in \text{SAT} \iff (\alpha[v_i = T] \in \text{SAT} \vee \alpha[v_i = F] \in \text{SAT})$. Now put it all together.

4. Show that if $\mathbf{NP} \subseteq \mathbf{P/poly}$ then $\Sigma_2^p = \Pi_2^p$.