Homework I
CSCI-GA.3350-001
Computational Complexity

February 8, 2014

You are expected to solve all the problems. Due on Monday, Feb 24. Collaboration is allowed; please mention your collaborators.

1. (2-SAT vs MAX-2SAT) : An instance of 2-SAT consists of $n$ boolean variables $x_1, x_2, \ldots, x_n$ and $m$ clauses, and each clause contains at most 2 literals. We say that the instance is satisfiable if there is a \{TRUE, FALSE\}-assignment to $x_1, x_2, \ldots, x_n$ that satisfies every clause.

   - Show that deciding if an instance of 2-SAT is satisfiable is in P. 
     \textit{Hint: Given two clauses, one involving } x_i \text{ and the other involving } \overline{x_i}, \text{ try and replace them with a single clause.}

   - Show that deciding if there is an assignment that satisfies at least $k$ out of $m$ clauses is NP-complete (this problem is known as MAX-2SAT). Note that the parameter $k$ is now part of the input. 
     \textit{Hint: Use a reduction from Vertex Cover.}

2. The class EXP is defined as

   $$\text{EXP} = \bigcup_k \text{DTIME}(2^{n^k})$$

   Show that NP \subseteq EXP and co-NP \subseteq EXP.

3. (Decision vs Search) : Show that if P = NP, there is a polynomial time algorithm to find a satisfying assignment to a 3-SAT formula if such an assignment exists.
4. Let BIPARTITE denote the language of all (undirected) graphs which are bipartite. Show that BIPARTITE $\in$ NL.

5. A directed graph is **strongly connected** if for every pair of vertices $(u, v)$ there is a directed path from $u$ to $v$ in $G$. Show that the problem of deciding whether a graph is strongly connected is NL-complete.

6. The 0-1 knapsack problem is defined as follows: Let $\{a_i\}_{i=1}^n, b$ be positive integers (represented in binary). The knapsack problem asks whether there is an integer solution to

$$\sum_{i=1}^{n} a_i X_i = b \quad X_i \in \{0, 1\}$$

We know that this problem is NP-complete.

Show that if we remove the constraints that $X_i \in \{0, 1\}$ and allow $X_i$’s to be arbitrary (possibly negative) integers, then the problem is in P. In other words, deciding if the following equation has an integer solution is in P.

$$\sum_{i=1}^{n} a_i X_i = b$$

7. A problem $A$ is NP-hard if there is a polynomial time reduction to it from some NP-complete problem ($A$ itself need not be in NP).

- Show that the following problem is NP-hard. Given a polynomial $P(X_1, \cdots, X_n)$ with integer coefficients, the problem is to decide whether the following equation has an integer solution :

$$P(X_1, \cdots, X_n) = 0$$

*Hint*: Show that in fact the problem is NP-hard for polynomials of degree 2, using a reduction from knapsack.

- Can’t we simply guess a solution if it exists and verify it ? Doesn’t this mean that the problem is in NP ?

8. **(Padding)**: For a language $L \subseteq \{0, 1\}^*$, and a function $f(n)$ (assume that $f(n)$ is computable in time $O(f(n))$), let $L_f \subseteq \{0, 1, \#\}^*$ denote the following language :

$$L_f := \{ x\#^{f(|x|)} \mid x \in L \}$$
• Suppose that \( L \in \text{DTIME}(f(n)) \). Then show that \( L_f \in \text{DTIME}(O(n)) \). Show similar results for non-deterministic time classes and deterministic space classes.

• Show that if \( f(n) \) is a polynomial function, then \( L \in \text{P} \) iff \( L_f \in \text{P} \).

• Show that \( \text{P} \neq \text{DSPACE}(O(n)) \). *Hint: Assume an equality and arrive at a contradiction via suitable padding and the Deterministic Space Hierarchy Theorem.*

• Define the class NEXP as

\[
\text{NEXP} := \bigcup_k \text{NTIME}(2^{n^k})
\]

Prove that if \( \text{P} = \text{NP} \) then \( \text{EXP} = \text{NEXP} \).