Computational Complexity

Problem set #2
Due date: 22 Ordibehesht 98 (via lms.iut.ac.ir)

Solve at least 6 problems to get full marks.

Problem 0. Assume that A is **NP**-complete and $B \in \mathbf{P}$. Prove that if $A \cap B = \emptyset$, then $A \cup B$ is **NP**-complete. What can you say about the complexity of $A \cup B$ if A and B are not disjoint?

Problem 1. Show that 2SAT is in **NL**.

Problem 2. **(a)** Suppose that $L \in \mathbf{DSPACE}(n^2)$ and define $L' = \{x\$^{n^2-n} : \ x \in L\}$. Prove that $L' \in \mathbf{DSPACE}(n)$. Also, prove that if $L' \in \mathbf{NP}$ then $L \in \mathbf{NP}$.

(b) Prove that $\mathbf{NP} \neq \mathbf{DSPACE}(n)$. (Note that we do not know if either class is contained in the other.)

Problem 3. Let M be an oracle Turing machine and let us define

$$L(M, B) := \{x \mid M(x) = 1 \text{ when } M \text{ uses oracle } B\}.$$

Suppose that for every two languages B_1, B_2, we have $L(M, B_1) = L(M, B_2) = L_0$. Also, suppose that there is some language B_0 such that when M uses oracle B_0, its runtime is polynomial. Prove that $L_0 \in \mathbf{NP} \cap \mathbf{CoNP}$.

Problem 4. Show that there is a language $B \in \mathbf{EXP}$ such that $\mathbf{NP}^B \neq \mathbf{P}^B$.

Problem 5. An undirected graph is bipartite if its vertices can be divided into two sets such that all edges have exactly one endpoint in each set. It known that a graph is bipartite iff it does not contain a cycle with an odd number of vertices. Let $\textsc{Bipartite} = \{G \mid G \text{ is a bipartite graph}\}$. Show that $\textsc{Biparitite} \in \mathbf{NL}$.

Problem 6. Define $\textsc{Cycle} = \{G \mid G \text{ is an undirected graph that contains a cycle}\}$. Show that $\textsc{Cycle} \in \mathbf{L}$. (Note that G may not be connected.)

Problem 7. We define the product of two $n \times n$ Boolean (0-1) matrices A and B as an $n \times n$ Boolean (0-1) matrix AB such that $(AB)_{ij} = \vee_{k=1}^{n}(A_{ik} \wedge B_{kj})$. (We think of 0 as **false** and 1 as **true** for this problem.)

(a) Given Boolean matrices A, B and integers i, j, show that $(AB)_{ij}$ can be computed in space $O(\log n)$.

(b) Using repeated squaring, argue that $(A^p)_{ij}$ can be computed in space $O(\log n \log p)$.

(c) Show that if A is the adjacency matrix of a graph, then $((A+I)^k)_{ij} = 1$ if and only if there is a path of length at most k from the vertex i to the vertex j.

(d) Use the above to give an alternative proof that $\mathbf{NL} \subseteq \mathbf{DSPACE}(\log^2 n)$.

Problem 8. Define the language

$$\textsc{ShortestPath} = \{(G, k, s, t) \mid G \text{ is a directed graph and the shortest path from } s \text{ to } t \text{ in } G \text{ has length } k\}$$

Prove that ShortestPath is in **NL**.

Problem 9. Consider the function $\mathbf{pad} : \Sigma^* \times \mathbb{N} \to (\Sigma \cup \$)^*$ defined as $\mathbf{pad}(x, i) = x\$^j$,where $j = \min(0, i - |x|)$. Thus, $\mathbf{pad}(x, i)$ just adds enough number of symbol \$ to the end of the string x so that the length of the new string is at least i. For a language A and a function $f : \mathbb{N} \to \mathbb{N}$, define the language $\mathbf{pad}(A, f(n))$ to be

$$\mathbf{pad}(A, f(n)) = \{\mathbf{pad}(x, f(|x|)) \mid x \in A\}$$

(a) Prove that if $A \in \mathbf{DTIME}(n^6)$, then $\mathbf{pad}(A, n^2) \in \mathbf{DTIME}(n^3)$.

(b) Define $\mathbf{EXP} = \cup_{c \geq 1} \mathbf{DTIME}(2^{n^c})$ and $\mathbf{NEXP} = \cup_{c \geq 1} \mathbf{NTIME}(2^{n^c})$. Use the function **pad** to prove that

$$\mathbf{NEXP} \neq \mathbf{EXP} \Rightarrow \mathbf{P} \neq \mathbf{NP}$$

Good Luck.