Theory of Computation

Mid-Term Examination on April 17, 2008
Spring Semester, 2008

Problem 1 (15 points)**.** Please answer the following questions.

1. [5 points] Is EXP $\subseteq$ NEXP true?

2. [5 points] Does there exist a language in NP decidable in time linear in the input length?

3. [5 points] In 1975, Richard Ladner proved the famous theorem that if NP $\neq$ P, then there is a language in NP that is neither in P nor NP-complete. Is the converse of the theorem true? That is, if some language in NP is neither in P nor NP-complete, can we conclude NP $\neq$ P?

Solution. The questions are answered below.

1. Yes because a deterministic Turing machine is by definition a degenerate nondeterministic Turing machine.

2. Yes, the empty language belongs to NP $\cap$ TIME(n).

3. Yes.

$\square$

Problem 2 (20 points)**.** Show that there exists a language $L \subseteq \{0,1\}^*$ that belongs neither to RE nor to coRE.

Solution. Since there are countably many Turing machines, RE $\cup$ coRE is countable. Hence we only need to show that there are uncountably many subsets of $\{0,1\}^*$ to complete the proof.

Assume for contradiction that there are only countably many subsets of $\{0,1\}^*$ and use $L_0, L_1, \ldots$ to denote these subsets. Write $\{0,1\}^* = \{x_k \mid k \in \mathbb{N}\}$ where $x_i \neq x_j$ for $i \neq j$. Then the subset $\hat{L} = \{x_k \mid x_k \notin L_k, k \in \mathbb{N}\}$ of $\{0,1\}^*$ must equal L_t for some $t \in \mathbb{N}$. But $x_t \in \hat{L}$ and $x_t \notin \hat{L}$ imply each other, which is absurd. $\square$

Problem 3 (20 points)**.** Show that there exists a language in NPSPACE that is not decidable in time cubic in the input length.

Solution. The time hierarchy theorem implies $\mathrm{TIME}(n^3) \subsetneq \mathrm{P}$. This and the trivial fact that $\mathrm{P} \subseteq \mathrm{NPSPACE}$ complete the proof. $\square$

Problem 4 (20 points)**.** Does there exist a non-recursive language in NP?

Solution. No, every language in NP can be decided in at most exponential time. $\square$

Problem 5 (20 points)**.** Show that it is NP-hard to determine whether a Boolean expression in conjunctive normal form has at least two satisfying assignments. (Hint: Consider adding a clause C to a Boolean expression F in conjunctive normal form where the variables in C do not appear in F.)

Solution. We describe a logspace reduction from SAT to the problem in question. On input a Boolean expression F in conjunctive normal form, the reduction outputs $F' = F \wedge (x \vee y \vee z)$ where x, y, z are variables that do not appear in F. Now if F is satisfiable, then F' has at least 7 satisfying assignments because $(x \vee y \vee z)$ is satisfied by seven assignments to x, y and z. Conversely, if F' has at least two satisfying assignments, it is clear that F must have at least one. $\square$

Problem 6 (20 points)**.** Prove that $\mathrm{NSPACE}(\log^2 n) \subseteq \mathrm{TIME}(2^{(\log^5 n)})$.

Solution. A nondeterministic Turing machine M with space complexity $O(\log^2 n)$ goes through no more than $c^{\log^2 |x|}$ configurations on any input x, for some $c > 1$ depending on M. Consider a directed graph $G = (V, E)$ where V consists of those configurations of M on x and E consists of those pairs (c_1, c_2) of configurations such that c_1 yields c_2. It takes $\mathrm{poly}(c^{\log^2 |x|})$ time to enumerate the vertices and edges of G, and $\mathrm{poly}(c^{\log^2 |x|})$ time by breadth-first or depth-first search to determine whether the initial configuration of M on x yields an accepting configuration. Hence $\mathrm{NSPACE}(\log^2 n) \subseteq \mathrm{TIME}(O(1)^{(\log^2 n)})$. $\square$

Another solution. Savitch's theorem implies $\mathrm{NSPACE}(\log^2 n) \subseteq \mathrm{SPACE}(\log^4 n)$. Hence $\mathrm{NSPACE}(\log^2 n) \subseteq \mathrm{SPACE}(\log^4 n) \subseteq \mathrm{TIME}(O(1)^{(\log^4 n)})$. $\square$