Computer Science 2401F
St. George Campus

November 19, 2002
University of Toronto

Homework Assignment #3
Due: December 3, 2002, by 1:10PM

Question 1. (10 marks) Let L be the set of $\langle M_1, M_2, \ldots, M_k \rangle$ such that $M_1, M_2, \ldots, M_k$ are deterministic finite state machines over a common alphabet, with the property that there exists a string $x \in \mathcal{L}(M_1) \cap \mathcal{L}(M_2) \cap \cdots \cap \mathcal{L}(M_k)$.

Prove that L is complete for PSPACE. (This is Exercise 3.10 in the text.)
(Hint for showing PSPACE hardness: You want the intersection to be a set of accepting computations; consider each of the positions in a configuration.)

Question 2. (10 marks) We define an integer expression E to consist of nonnegative integer constants (written in binary), combined using the operators $+$ and $\cup$ (and parentheses). Every such expression E represents a set of nonnegative integers $\mathcal{L}(E)$ defined as follows:
If k is a nonnegative integer, then $\mathcal{L}(k) = \{k\}$;
$\mathcal{L}(E_1 + E_2) = \{m + n \mid m \in E_1 \text{ and } n \in E_2\}$;
$\mathcal{L}(E_1 \cup E_2) = \mathcal{L}(E_1) \cup \mathcal{L}(E_2)$.

For example, if $E_1 = 1 \cup 2 \cup 3$, $E_2 = (1 \cup 2) + 1$, $E_3 = 1 \cup ((1 \cup 2) + 1)$, then $\mathcal{L}(E_1) = \mathcal{L}(E_3) = \{1, 2, 3\}$ and $\mathcal{L}(E_2) = \{2, 3\}$.

We define the language L to consist of $\langle E, E' \rangle$ such that E and E' are integer expressions and $\mathcal{L}(E) = \mathcal{L}(E')$. (Continuing the above example, we see that $\langle E_1, E_3 \rangle \in L$, but $\langle E_1, E_2 \rangle \notin L$.)

a. (10 marks)
Prove that $L \in \Pi_2^P$.

b. (10 marks of **Extra Credit**)
Prove that L is hard for Π_2^P.

Question 3. (15 marks) Prove that the following language L is complete for Δ_2.
(Extra credit if you hand in this question by Nov. 26, before a hint is given.)

We say that $F \in L$ if F is a formula of the propositional calculus with the following property:
F has boolean variables $X_1, X_2, \cdots X_n$; we can view a truth assignment τ as an n-bit string, and hence as an n-bit integer, where the value given X_1 is the high-order bit and the value given X_n is the low-order bit; then we define
$F \in L \iff F$ is satisfiable, and the largest (when viewed as an integer) truth assignment that satisfies F is odd (that is, makes X_n true).

Question 4. (20 marks) For this question, we will define the *exponential time hierarchy*. For positive integer k, we say $L \in \Sigma_k^E$ if for some constant c and some polynomial time predicate R, for all strings x,
$x \in L \Leftrightarrow \exists y_1, |y_1| \leq 2^{c|x|}, \forall y_2, |y_2| \leq 2^{c|x|}, \exists y_3, |y_3| \leq 2^{c|x|}, \ldots, Q\, y_k, |y_k| \leq 2^{c|x|} R(x, y_1, y_2, \ldots, y_k)$.

a.
Prove that if **P**=**NP**, then for every $k \geq 1$, $\Sigma_k^E = \mathbf{E}$. (Recall that $\mathbf{E} = \cup_c \mathrm{DTIME}(2^{cn})$.)

b.
For every positive integer n, let $\max(n)$ be the worst case, over all subsets U of $\{0,1\}^n$, of the circuit complexity of U. For every positive integer n, let $L_n \subseteq \{0,1\}^n$ be the "lexicographically first" set of strings requiring circuit complexity $\max(n)$. Let $L = \cup_n L_n$.

Prove that for some k, $L \in \Sigma_k^E$.

Remark: These two parts imply that if **P**=**NP**, then there exists an $L \in \mathbf{E}$ that requires circuit size $\max(n)$ for every n. Contrapositively, if for every $L \in \mathbf{E}$ there is some n such that $L \cap \{0,1\}^n$ has circuit complexity at most $\max(n) - 1$, then $\mathbf{P} \neq \mathbf{NP}$.