Complexity Theory (Winter 2016/17)	Final Exam

This exam has two parts. Please attempt *all* problems from Part I. Part II has three problems and *you can choose any two.*

Part I: Short Questions (XX points)

Attempt all questions in this part.

Problem 1. (2 points) What was the most interesting thing you learnt in this course?

Problem 2. ($5 \times 4 = 20$ points) For each problem, give a brief justification of your answer. In each case, the justification should only be a few lines.

a What is wrong about the following proof that PH = PSPACE? We show QBF is in PH. Given any input φ with k quantifier alternations, we know φ is in Σ_k^p-SAT if it starts with an $\exists$ and φ is in Π_k^p-SAT if it starts with a $\forall$. In either case, it is in PH. Thus, every instance of QBF can be solved by some language in PH and so QBF is in PH.

b What is an example of a NP-hard problem which is (provably) not NP-complete?

c Is it possible that QBF is EXPSPACE-complete?

d A biased coin, which lands heads with probability $\frac{1}{10}$ each time it is flipped, is flipped 200 times consecutively. Using Markov's inequality, give an upper bound on the probability that it lands heads at least 120 times.

e Prove that $\mathrm{PCP}(0, \log n) = \mathrm{P}$.

Problem 1.2. (8 points) Consider the reduction from SAT to INDSET in the book/lectures. Show that the reduction is parsimonious.

OR

Recall that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *linear* if for all $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$, we have $f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$. Show that a function f is linear iff there is some $\mathbf{a} \in \{0,1\}^n$ such that $f(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x}$. (Here, $\mathbf{a} \cdot \mathbf{x} = \sum_{i=1}^n \mathbf{a}_i . \mathbf{x}_i (\mod 2)$.) How many linear functions are there from $\{0,1\}^n$ to $\{0,1\}$?

Part II: Long Questions (XX points)

Attempt *any two* questions in this part.

Problem 1.3. We define a new complexity class involving alternating quantifiers: $\mathbf{S}_2^p$ (the "$\mathbf{S}$" stands for "symmetric alternation"). A language L is in $\mathbf{S}_2^p$ if and only if there is a language $R \in \mathrm{P}$ for which

$$x \in L \Rightarrow \qquad \exists y \forall z (x, y, z) \in R \tag{1}$$

$$x \notin L \Rightarrow \qquad \exists z \forall y (x, y, z) \notin R \tag{2}$$

where as usual $|y| = poly(|x|)$ and $|z| = poly(|x|)$. To make sense of this definition it is useful to think of R as defining for each x a 0/1 matrix M_x whose rows are indexed by y and whose columns are indexed by z. Entry (y, z) of matrix M_x is 1 if $(x, y, z) \in R$ and 0 otherwise. Now, the definition says that $x \in L$ if there is an all-ones row in M_x and $x \notin L$ if there is an all-zeros column in M_x (and it is clear that these configurations are mutually exclusive).

1. Argue that $\mathbf{S}_2^p \subseteq \Sigma_2^p \cap \Pi_2^p$.

2. The language Lex-First-Acceptance consists of those pairs (C_1, C_2) for which C_1, C_2 are Boolean circuits on the same number of inputs, and the lexicographically first string x for which $C_1(x) = 1$ is also accepted by C_2. (If there is no lexicographically first string, i.e., C_1 is unsatisfiable, then (C_1, C_2) is not in the language). A bitstring x lexicographically precedes a bitstring y if the first position i in which they differ has $x_i = 0$ and $y_i = 1$.

   Prove that Lex-First-Acceptance is in $\mathbf{S}_2^p$.

Problem 1.4.

1. Argue that NP = BPP implies that the polynomial hierarchy collapses.

2. Show that $\mathrm{BPP}^{\mathrm{BPP}} = \mathrm{BPP}$. That is, a BPP machine with access to a BPP oracle accepts the same class of languages as BPP. [For this part, use the error amplification property of BPP as given. That is, for any BPP machine, for any input x, by asking some polynomial number of queries to the BPP machine and taking majority, we can assume that the error probability is exponentially small.]

3. Conclude that if $\mathrm{NP} \subseteq \mathrm{BPP}$ then $\mathrm{PH} \subseteq \mathrm{BPP}$.

Problem 1.5.

1. A language L is *sparse* if there is a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that for each $n \geq 0$, $|L \cap \{0,1\}^n| \leq p(n)$. That is, for each length n, the language L has at most $p(n)$ strings of length n. Show every sparse language is in $P/poly$.

2. Show that a language L is in $P/poly$ iff $L \in \mathrm{P}^A$ for some sparse set A.

3. Show that PSPACE $\subseteq P/poly$ implies PSPACE $= \Sigma_2^p$. [Hint: Consider a polynomial circuit family for QBF. Given circuits C_1, …, C_m for QBF instance of size 1, …, m, how can you verify that the circuit C_m is correct using the circuits C_1, …, C_{m-1}?]