Test #2

Due: 10:00am, Thursday December 7

No collaboration or Internet-usage allowed!
You may cite results from lecture, past homework problems, and the textbook.

Solve four problems total, namely #1, #2, and two out of three from $\{\#3, \#4, \#5\}$.

---

1. **(PH collapsing when efficient means expected polynomial time.)**

   Show $\mathsf{NP} \subseteq \mathsf{ZPP} \implies \mathsf{PH} = \mathsf{ZPP}$.

   (Hint: don't be surprised if your proof fits on one line.)

2. **(Optimal Karp–Lipton for NEXP.)**

   (a) (This part is worth 1 point, as the proof is about one sentence long.) Read the proof of the IKW Theorem, Lemma 20.20 in the textbook, which uses the "easy witness method" to show that $\mathsf{NEXP} \subseteq \mathsf{P}/\mathrm{poly} \implies \mathsf{NEXP} = \mathsf{EXP}$. Now show that in fact $\mathsf{NEXP} \subseteq \mathsf{P}/\mathrm{poly} \implies \mathsf{NEXP} = \mathsf{MA}$.

   (b) In lecture we focused on showing that strong hardness assumptions imply deterministic poly-time algorithms for BPP; but, we mentioned that if one works the parameters, one gets that weak hardness assumptions imply deterministic subexponential-time algorithms for BPP. Specifically, one can show that if there is a language $L \in \mathsf{EXP}$ that requires superpolynomial circuit size for almost all input lengths n, then for all $\varepsilon > 0$ there is a pseudorandom generator G with seed length $\ell(n) \leq n^{\varepsilon}$. Under this assumption, conclude that $\mathsf{MA} \subseteq \mathsf{NTIME}(2^n)$.

   (c) The above says that if EXP requires superpolynomial circuit size for almost all input lengths, then MA is nondeterministically simulable in $O(2^n)$ time for almost all n. You may now take it for granted that the "infinitely often" version is also true (the proof is essentially the same); namely, that $\mathsf{EXP} \not\subseteq \mathsf{P}/\mathrm{poly} \implies \mathsf{MA} \subseteq \text{i.o.-}\mathsf{NTIME}(2^n)$. Here i.o.-$\mathcal{C}$ denotes the class of all languages A such that there exists $B \in \mathcal{C}$ with $A \cap \{0,1\}^n = B \cap \{0,1\}^n$ for infinitely many n.

   You may also take for granted (cf. Homework 2, #1(d)) the following Time Hierarchy Theorem result: for all $c \in \mathbb{N}$ it holds that $\mathsf{EXP} \not\subseteq \text{i.o.-}\mathsf{TIME}(2^{n^c})$. (Remark: we do not know the nondeterministic version of this result.)

   Now prove the following: $\mathsf{NEXP} = \mathsf{MA} \implies \mathsf{NEXP} \subseteq \mathsf{P}/\mathrm{poly}$.

3. **(One-way functions and complexity classes.)** A "worst-case one-way function" is a function $f : \{0,1\}^* \to \{0,1\}^*$ with the following properties: (i) f is one-to-one (injective); (ii) f does not stretch or shrink by more than a polynomial amount, i.e., there exists $k > 0$ such that $|x|^{1/k} \leq |f(x)| \leq |x|^k$ for all x; (iii) f is computable in polynomial time; (iv) the inverse function $f^{-1} : \{0,1\}^* \to (\{0,1\}^* \cup \{\perp\})$ is *not* computable in polynomial time, where $f^{-1}(y)$ is defined to be x if $f(x) = y$, or else $\perp$ if $y \notin \mathrm{range}(f)$.

The complexity class UP (not its real name) is defined to be the set of all languages L for which there exists a polynomial-time nondeterministic Turing Machine M with the following properties: (i) if $x \in L$ then $M(x)$ accepts on exactly one "nondeterministic branch"; (ii) if $x \notin L$ then $M(x)$ accepts on exactly zero "nondeterministic branches". As a remark, it is immediate that $\mathsf{UP} \subseteq \mathsf{NP}$, and it's also easy to see that $\mathsf{P} \subseteq \mathsf{UP}$.

(a) Prove that if $\mathsf{UP} \neq \mathsf{P}$ then there is a worst-case one-way function.

(b) Conversely, prove that if $\mathsf{UP} = \mathsf{P}$ then worst-case one-way functions do not exist.

4. **(O1.)** Remember that complexity class "$\mathsf{S_2P}$" from Homework 5, Problem 1? Here we describe a variant of it called "$\mathsf{O_2P}$". The class $\mathsf{O_2P}$ is just like $\mathsf{S_2P}$ except Yolanda and Zeyuan are too lazy to even look at the input x; they only look at its length, n. More precisely, we say that $L \in \mathsf{O_2P}$ if there is a polynomial $p(n)$ and a polynomial-time algorithm V such that for all n, there exist strings $y^*, z^* \in \{0,1\}^{p(n)}$ such that for all $x \in \{0,1\}^n$,

$$x \in L \implies \forall z \in \{0,1\}^{p(n)}\ V(x, y^*, z) = 1,$$
$$x \notin L \implies \forall y \in \{0,1\}^{p(n)}\ V(x, y, z^*) = 0.$$

Prove that $\mathsf{BPP} \subseteq \mathsf{O_2P}$.

5. **(O2: Revenge of Karp–Lipton.)**

(a) Show that $\mathsf{NP} \subseteq \mathsf{P}/\text{poly} \implies \mathsf{PH} = \mathsf{O_2P}$.

(b) Show that $\mathsf{PH} = \mathsf{O_2P} \implies \mathsf{NP} \subseteq \mathsf{P}/\text{poly}$.