Homework 6
Due Fri May 7, 5:00pm

Prof. Amit Chakrabarti
Computer Science Department
Dartmouth College

General Instructions: Same as in Homework 1.

Honor Principle: For this homework, you should work entirely on your own and not discuss with anyone.

13. Let $k > 0$ be an integer. Construct a language in PH that is not in $\mathsf{SIZE}(n^k)$. [2 points]

    This is a hard problem. To get started, recall from the lectures that we proved that there *exist* languages in $\mathsf{SIZE}(n^{2k})$ that are not in $\mathsf{SIZE}(n^k)$. Try writing out this fact formally, using quantifiers: you should have a small, fixed number of quantifier alternations. This suggests that you might be able to place the required language in either Σ_i^p or Π_i^p, for some *fixed* i, independent of k.

    A further hint is given on the next page. I *strongly* recommend that you turn the problem over in your mind for a day at least, before looking at that hint.

14. An *unbounded fan-in circuit* is just like the circuits we defined in class — i.e., DAGs whose vertices (gates) are inputs ($x_1, \ldots, x_n$), negated inputs ($\neg x_1, \ldots, \neg x_n$), and logic gates (AND/OR), with one or more gates of fan-out 0 designated as output(s) — except that the restriction that gates have fan-in 2 is removed. Size and depth are defined as before: number of edges (wires) and maximum path length, respectively.

    Prove that there exists a suitable size function $s : \mathbb{N} \to \mathbb{N}$ such that every Boolean function $f : \{0,1\}^n \to \{0,1\}$ has unbounded fan-in circuits with $O(1)$ depth and $O(s(n))$ size. Find the smallest possible $s(n)$ you can.

    [2 points]

15. Give a full formal proof that $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$. [2 points]

16. For constants $0 < \alpha < \beta < 1$, define the class $\mathsf{BPP}_{\alpha,\beta}$ to be the class of all languages $L \subseteq \Sigma^*$ such that there exists a PTM M that runs in polynomial time and behaves as follows on an input $x \in \Sigma^*$:

$$\begin{aligned} x \notin L &\Rightarrow \Pr_R[M(x,r) = 1] \leq \alpha\,, \\ x \in L &\Rightarrow \Pr_R[M(x,r) = 1] \geq \beta\,. \end{aligned}$$

    Note that our definition of BPP in class coincides with $\mathsf{BPP}_{\frac{1}{3},\frac{2}{3}}$ in this notation.

    Using Chernoff bounds, give a full formal proof that for all α and β as above, $\mathsf{BPP}_{\alpha,\beta} = \mathsf{BPP}$.

    [2 points]

    The Chernoff bound has the following general form. Let $\{X_1, \ldots, X_n\}$ be independent indicator random variables with $\mathbb{E}[X_i] = p_i$. Suppose $X = \sum_{i=1}^n X_i$ and let p be such that $np = p_1 + \cdots + p_n$. Then, for any $\delta > 0$:

$$\Pr[X \geq (1+\delta)np] \;\leq\; \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{np}.$$

    We also have a similar inequality bounding deviations of X *below* its mean. For $0 < \delta < 1$:

$$\Pr[X \leq (1-\delta)np] \;\leq\; \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{np}.$$

Hint for Problem 13: Did you think about the problem for a day, at least? If not, please do!

For a string $w \in \{0,1\}^n$, let B_w denote the Boolean circuit described by w; if w is not a well-formed encoding of a circuit due to syntax errors, define B_w to be a trivial circuit that always outputs 0 (say). Let $C(x)$ denote the output of circuit C on input x. Argue that, for $s \in \mathbb{N}$ and $w, x \in \{0,1\}^*$, the predicates "$\text{size}(B_w) \leq s$" and "$B_w(x) = 1$" are decidable in polynomial time. Therefore, if we use a fixed number of quantifier alternations and then perform an inner computation that involves evaluating these types of predicates (maybe a few times), we'll have either a Σ_i^p or a Π_i^p computation, depending upon whether we start with a "$\exists$" or a "$\forall$" quantifier.

Now consider the following statement $\phi_n(x)$, for an $x \in \{0,1\}^n$, and figure out what it's saying:

$$\phi_n(x) \;=\; \exists\, w \in \{0,1\}^* \left(\text{size}(B_w) \leq n^{2k} \;\wedge\; B_w(x) \;\wedge\; \forall\, v \in \{0,1\}^* \left(\text{size}(B_v) \leq n^k \Rightarrow \exists\, y \in \{0,1\}^n \; (B_v(y) \neq B_w(y))\right)\right).$$

Once you have digested it, you'll find that this is close to what we need to create a suitable language in PH. But unfortunately $\{x \in \{0,1\}^* : \phi_{|x|}(x)\}$ ends up contaning *all* sufficiently long Boolean strings, so this is not the language we seek! Figure out why this happens, and then think of what you can do to fix it. Perhaps you need more quantifiers.