Homework 9

Due: 5:00pm, Thursday April 13

Feature: As before, if your homework is typeset (as opposed to handwritten), you will receive 1 bonus point.

---

1. **(Exact distance.)** (10 points.) Let ST-DIST be the language of all strings $\langle G, s, t, d\rangle$, where G is a directed graph, s and t are vertices in it, $d \in \mathbb{N}$, and the minimum distance from s to t in G is exactly d. Show that ST-DIST $\in \mathsf{NL}$.

2. **(On RP-completeness.)** (You might like to compare this problem with #4 on Homework #6.) Consider the language

$$\text{RS} \coloneqq \{\langle M, x, 1^t\rangle : M \text{ is a randomized TM, } x \text{ is a Boolean string, } t \in \mathbb{N},\\ M(x) \text{ halts within at most } t \text{ steps and has } \mathbf{Pr}[M(x) = \text{accept}] \geq 2/3\}.$$

   (a) (7 points.) Show that RS is RP-hard, under $\leq_m^L$.

   (b) (3 points.) Attempt to show RS $\in \mathsf{RP}$. Describe what seems to go wrong.

   (c) (0 points.) Do you think there is a language that is RP-complete (under $\leq_m^L$)?

3. **(On ZPP.)** Recall that our official definition of the complexity class ZPP is $\mathsf{RP} \cap \mathsf{coRP}$. You will explore equivalent definitions in this problem.

   (a) (5 points.) Consider augmenting our definition of a randomized Turing Machine with one more kind of halting state beyond "Accept" and "Reject", namely "?". Show that ZPP is equal to the class of all languages L such that there is a machine M of this type, with the following three properties:

   - M's running time is poly(n);
   - M never gives a wrong answer, meaning that for all inputs x,
   $$x \in L \implies \mathbf{Pr}[M(x) = \text{Reject}] = 0 \quad \text{and} \quad x \notin L \implies \mathbf{Pr}[M(x) = \text{Accept}] = 0;$$
   - $\mathbf{Pr}[M(x) = ?] \leq 1/3$ for all inputs x.

   (Remark: we sort of talked about the solution of this problem in class, but please spell it out in both directions.)

   (b) (5 points.) Show that ZPP is equal to the class of all languages L such that there is a standard (Accept/Reject only) randomized Turing Machine that never gives a wrong answer and that has *expected* polynomial running time; i.e., for which there exists a constant c with

   $$\mathop{\mathbf{E}}_{\substack{\text{"coin flips"}\\ \text{of } M}}[\#\text{ steps } M(x) \text{ takes}] \leq O(n^c), \quad \text{for all inputs } x \text{ of length } n.$$

   (Recall that usually, including in part (a), the running time function of a randomized Turing Machine is defined by taking the *maximum* over all possible coin flips. In this problem, you may need to design randomized Turing Machines that have no a priori upper bound on how many steps they might take. You may also need Markov's Inequality; you can look up what that is, if you forget it.)

4. **(On PP.)** The randomized complexity classes ZPP, RP, coRP, BPP were defined by John Gill III in the early '70s, basically during his PhD under the direction of Manuel Blum. We didn't define BPP in class yet, but it is the "two-sided error" extension of RP and coRP; namely,

$$\begin{aligned} \mathsf{BPP} = \{L : \text{there is a poly-time randomized Turing Machine } M \text{ such that} \\ x \in L \implies \mathbf{Pr}[M(x) \text{ accepts}] \geq 2/3 \\ x \notin L \implies \mathbf{Pr}[M(x) \text{ accepts}] \leq 1/3\}. \end{aligned}$$

This class is normally considered to capture "efficient randomized computation".

Around the same time, Janos Simon (during *his* PhD) defined another complexity class to model probabilistic polynomial time, called PP. No offense to Simon, but it was not the most successful definition, in the sense that it does not do a great job of capturing "efficient randomized computation". (Actually, he recognized that at the time.) Still, it's kind of an interesting definition, so we explore it here. Formally:

$$\begin{aligned} \mathsf{PP} = \{L : \text{there is a poly-time randomized Turing Machine } M \text{ such that} \\ x \in L \implies \mathbf{Pr}[M(x) \text{ accepts}] > 1/2 \\ x \notin L \implies \mathbf{Pr}[M(x) \text{ accepts}] \leq 1/2\}. \end{aligned}$$

The reason this class is not considered a good model for the problems solvable efficiently with randomness is that there may be no noticeable "gap" in the probabilities of accepting, between $x \in L$ and $x \notin L$. For example, it could be that $x \notin L$ and that $\mathbf{Pr}[M(x) \text{ accepts}]$ is $1/2$, or maybe $1/2 - 2^{-n}$. In that case, even if you re-run $M(x)$ many times, you'll basically see an even mix of "accepts" and "rejects" and you won't gain any real confidence about whether or not $x \in L$.

(a) (1 point.) Show that $\mathsf{BPP} \subseteq \mathsf{PP}$.

(b) (4 points.) Show that $\mathsf{NP} \subseteq \mathsf{PP}$. (This further shows that PP is a "practically unreasonable" class.)

(c) (5 points.) Show that $\mathsf{co\text{-}PP} = \mathsf{PP}$. (This problem is *almost* super-easy, except for one annoying issue...)