Additional complexity problems

Ivan Gavran

February 20, 2017

1. Here is a diagonalization argument that shows $P \neq NP$. Enumerate all poly-time deterministic Turing machines $M_1, M_2, \ldots$. Consider the language $L_{\text{diag}} = \{x : M_x(x) = 0\}$. We claim two things:

   - $L_{\text{diag}}$ is not accepted by any poly-time machine. Indeed, assume there was a machine $M$ accepting it. Let $m$ be the encoding of $M$. If $m \in L_{\text{diag}}$, then $M(m) = 0$ which means $m \notin L_{\text{diag}}$. Otherwise, if $m \notin L_{\text{diag}}$, then $M(m) = 1$ and since $M$ is a decider for $L_{\text{diag}}$ it means $m \in L_{\text{diag}}$.

   - $L_{\text{diag}} \in NP$: on input $x$ a universal Turing machine can simulate execution of $M_x$ with polynomial overhead.

2. Show: if $NP \subset BPP$ then $NP = RP$ (maybe a bit too difficult for an exam question)

3. Show that DAG (Directed Acyclic Graph) reachability is NL-complete.

4. Show that if $\text{NEXP} \neq \text{EXP}$ then $P \neq NP$

5. Class BPL is defined the following way: $L \in \text{BPL}$ if there is a probabilistic logspace Turing Machine $M$ such that

   - $x \in L \Rightarrow \Pr[M(x) = 1] \geq \frac{2}{3}$
   - $x \notin L \Rightarrow \Pr[M(x) = 1] \leq \frac{1}{3}$

   Show that $\text{BPP} \subset P$

**NOTE:** This is not the final version of the problem list, the new problems might be added throughout this week.