**CSE696, Spring 2019** **Problem Set 1** **Due Tue. 3/12**

**Reading:** We are currently in Chapter 7 of the Arora-Barak online text, and at the same time in Chapter 17 for counting complexity, which they consider an "advanced topic" but others do not. After proving the amplification theorem for BPP and noting the #P-completeness of #SAT, we will do the theorems that BPP is contained in the second level of PH and that PH is contained in $\mathsf{P}^{\mathsf{PP}}$. We've also covered some material from chapters 3 and 5 also given out.

(1) Show that the language of Turing machines $M_i$ such that $M_i$ is total and $L(M_i)$ is infinite belongs to $\prod_2$ by writing a $\prod_2^0$ definition for it. You may regard any decidable predicate as "ground" requiring no further quantifiers. (The only ones you need here are the Kleene $T$-predicate $T(i, x, c) \equiv c$ is a valid halting computation of $M_i$ on input $x$, a relation $U(c)$ saying $c$ accepts, and basic numerical comparisons.)

Now how about the language of $i$ such that $M_i$ is total and $L(M_i)$ is finite? Can you prove it is neither in $\prod_2$ nor in $\sum_2$? (One hint: Consider designing a machine that accepts $\langle x, m\rangle$ if and only if $M_i$ accepts $x$ in *exactly* $m$ steps. 12 + 24 = 36 pts.)

(2) Give a $\sum_3^0$ definition for the language $I_{\mathsf{REC}} = \{i : L(M_i) \text{ is decidable}\}$. Note that $M_i$ itself need not be total. Take it as a fact—we won't ask for a proof—that $\sum_3^0$ is optimal.

Now suppose $F$ is a *sound* and *effective* formal system, meaning it has a decidable proof predicate $P(\phi, \pi) \equiv \pi$ is a proof of the sentence $\phi$ in $F$, such that whenever it holds, $\phi$ is actually true. For any TM $M_i$, we can make a sentence $\phi = \tau(i)$ from the *formula* $\tau(i) = (\forall x)(\exists c)T(i, x, c)$. Then when $P(\tau(i), \pi)$ holds, we call $M_i$ a *provably total* machine. Use the fact about $\sum_3^0$ and $I_{\mathsf{REC}}$ to deduce that there are decidable languages that are not accepted by any provably total machine. (12 + 12 = 24 pts.)

(3) Define the "anti-index set" of any subclass $\mathcal{C}$ of the decidable languages by $R_{\mathcal{C}} = \{i : \tau(i) \wedge L(M_i) \notin \mathcal{C}\}$. That is, $R_{\mathcal{C}}$ is the set of total machines that accept languages outside the class. Show that for $\mathcal{C} = \mathsf{P}$, $R_{\mathcal{C}}$ belongs to $\prod_2$. How about for $\mathcal{C} = \mathsf{NP}$? Now try it for $\mathcal{C} = \mathsf{BPP}$ where you don't have a simple recursive enumeration of total machines representing the class, but... (30 pts. total)

(4) Prove that a language $A$ belongs to $\mathsf{NP} \cap \mathsf{co\text{-}NP}$ if and only if $\mathsf{NP}^A = \mathsf{NP}$. (24 pts.)

(5) Define $\mathsf{H}$ to be the intersection of $\mathsf{P}^A$ over all oracles $A$ such that $\mathsf{NP}^A = \mathsf{P}^A$. Show the following facts about $\mathsf{H}$:

(a) $\mathsf{PH} \subseteq \mathsf{H} \subseteq \mathsf{PSPACE}$.

(b) If $\mathsf{PH} = \sum_k^p = \prod_k^p$ for some $k$ (read as saying that the polynomial hierarchy "collapses" to the $k$th level), then $\mathsf{H} = \sum_k^p = \prod_k^p$ too.

(c) If $R_{\mathsf{H}} \notin \prod_2$ then the polynomial hierarchy is infinite and different from PSPACE and H is properly between PH and PSPACE.

(d) If $\mathsf{H} = \mathsf{PH}$, does that prevent the polynomial hierarchy from being infinite? (Actually, I don't know...)

I invented this idea long ago and used "H" to mean a kind of disembodied hierarchy. Except that maybe $\mathsf{H} = \mathsf{PH}$ can be proved without major consequences, the idea hasn't gone anywhere except as an illustration of grasping and playing with both the logical and complexity concepts. (12 + 9 + 9 = 30 pts. total, possibly more if you do something with (d), for 144 pts. on the set)