CSE696, Spring 2021 Problem Set 2 Due Mon. 3/15

Reading:

After working through Arora-Barak chapter 7 (skimming section 7.5), we will go to section 9.3, then chapter 8.

(1) Let $\leq_m^{ql}$ stand for many-one reducibility by functions computable in deterministic *quasi-linear* time, which means time $n(\log n)^{O(1)}$. All functions of the form $cn(\log n)^d$ where c and d are positive rational numbers (and logs are to base 2) are fully time constructible, so you may freely assume the ability of Turing machines to count up to or down from such function values. It follows that the classes $\mathsf{DQL} = \mathsf{DTIME}[n(\log n)^{O(1)}]$ and $\mathsf{NQL} = \mathsf{NTIME}[n(\log n)^{O(1)}]$ are recursively presentable in a natural and relativized way by time-clocked oracle Turing machines. Also, SAT is complete for NQL under $\leq_m^{ql}$. So whether $\mathsf{NQL} \neq \mathsf{DQL}$ is a variant of the P versus NP question. One advantage of $\leq_m^{ql}$ reductions is that they also allow us to probe the fine structure within polynomial time. This problem is about a subtlety that appears more clearly there.

(a) Show that $\mathsf{DTIME}[n^2]$ has a complete set B under $\leq_m^{ql}$. (12 pts.)

(b) Define a recursive presentation of the languages that are complete for $\mathsf{DTIME}[n^2]$ under $\leq_m^{ql}$. Does it yield a recursive presentation of the complete sets for $\mathsf{DTIME}^E[n^2]$ under $\leq_m^{ql}$, for all oracles E? (12 pts.)

(c) Show that there are languages A such that:

- $A \in \mathsf{DTIME}[n^2]$,
- for all $\epsilon > 0$, $A \notin \mathsf{DTIME}[n^{2-\epsilon}]$,
- A is not complete for $\mathsf{DTIME}[n^2]$ under $\leq_m^{ql}$, and
- A is a proper subset of B.

(24 pts., for 48 total) As a "work-in" problem, we will show that the class of languages that belong to $\mathsf{DTIME}[n^{2-\epsilon}]$ for some $\epsilon > 0$ is recursively presentable—indeed, in a relativizing manner (though that is not needed for this problem).

(2) Suppose f is a function in $\#\mathsf{P}$, with $p(n)$ a length-bounded polynomial associated with a polynomial-time predicate $R(x, y)$ such that $f = f_R$. For any n, let us call a circuit C_n with $n + p(n)$ inputs a *witness circuit* for f at n if C_n decides $R(x, y)$ for $x \in \Sigma^n$, so that $f(x) = \#^p y.C_n(x, y)$. Let the size $s(C_n)$ be the count of wires.

(a) Show that not only does the function $2f$ belong to $\#\mathsf{P}$, but also given any n and witness circuit C_n, we can build in $n^{O(1)}$ time a witness circuit C'_n of size $s(C'_n) = s(C_n) + O(1)$. We say that "$\#\mathsf{P}$ is closed under the operation of going from f to $2f$ with additive overhead." (9 pts.)

(b) Now suppose that #P is closed under the operation of going from f to $f(x)/2$ (integer division by 2) with additive overhead. Show that $\mathsf{PP} = \mathsf{NP} \cap \mathsf{co\text{-}NP}$ would follow. (Hint: Utilize the "bare majority" form of PP shown in lectures and iterate divisions by 2, for 12 pts. and 21 total. Can you get $\mathsf{PP} = \mathsf{P}$ to follow? Can you make any of this work if the operation has linear overhead $s(C''_n) \leq Ks(C_n)$ for some constant $K > 1$? There may be extra credit if you can do something nontrivial here...)

Work-in problem: We will show that the operations $f + g$ and $f * g$ have additive overhead in a related sense. This also serves as a work-in for the next problem.

(3) Let $f(w, x)$ be a two-variable function in #P. Show that then the single-variable function

$$g(x) = \sum_{w \in \{0,1\}^{|x|}} f(w, x)$$

belongs to #P. (15 pts.)

(4) Show that BPP is *self-low*, meaning that $\mathsf{BPP}^{\mathsf{BPP}} = \mathsf{BPP}$. Use amplification as desired. Then show that $\mathsf{NP}[\mathsf{BPP}] \subseteq \mathsf{BP}[\mathsf{NP}]$. (9+9 = 18 pts. Does the latter inclusion go the other way? 171,717 extra credit points if you prove it—that should be a hint... For work-in, we will prove that $\mathsf{BP}[\mathsf{BPP}] = \mathsf{BPP}$.

(5) Suppose $\mathcal{F}$ is any formal system of logic that has a decidable proof predicate $P_{\mathcal{F}}(\theta, \pi)$ saying that π is a proof of the sentence θ in $\mathcal{F}$. Now let $R(i)$ be any predicate of Turing machines M_i that entails M_i being total. Define

$$\mathcal{C}_R = \{L(M_i) : (\exists \pi) P_{\mathcal{F}}(R(i), \pi)\},$$

which is the class of languages accepted by programs that $\mathcal{F}$ can prove to have property R.

- Presuming that $\mathcal{C}_R$ is nonempty, show that $\mathcal{C}_R$ is recursively presentable. (6 pts., easier than you may think...)
- Deduce that there is a computable time function $t(n)$ so that $\mathcal{C}_R \subseteq \mathsf{DTIME}[t(n)]$. (6 pts.)
- Show that there is a computable function $\sigma(n)$ such that for every machine M_i in your presentation, the function computed by M_i is $o(\sigma(n))$. In particular, $\sigma(n)$ outgrows all functions that $\mathcal{F}$ can prove to belong to the class $\mathcal{C}_R$. (6 pts., for 18 on the problem and 120 on the set)