1. The Karp-Lipton-Sipser result suggests that co-NP $\not\subseteq$ NP/poly, if we, for example, "believe" that PH does not collapse. The situation is very different for nondeterministic exponential time. Define NE as:

$$\text{NE} = \bigcup_{c \geq 1} \text{NTIME}\left[2^{cn}\right]$$

and as usual define co-NE to be the complements:

$$\text{co-NE} = \{\overline{L} \mid L \in \text{NE}\}.$$

Show that co-NE $\subseteq$ NE/poly. *Hint:* Think census.

2. For a class of languages $\mathcal{C}$, we define $\exists{\cdot}\mathcal{C}$ and BP$\cdot\mathcal{C}$ as follows:

Defn: $L \in \exists{\cdot}\mathcal{C}$ if there exists a language $A \in \mathcal{C}$ and a polynomial $p()$ such that

$$x \in L \iff \exists y,\ |y| = p(|x|) \text{ and } \langle x, y\rangle \in A.$$

Defn: $L \in \text{BP}{\cdot}\mathcal{C}$ if there exists a language $A \in \mathcal{C}$ and a polynomial $p()$ such that

$$x \in L \Longrightarrow \text{Prob}_y[\langle x, y\rangle \in A] \geq 2/3$$

$$x \notin L \Longrightarrow \text{Prob}_y[\langle x, y\rangle \in A] \leq 1/3$$

where y is chosen uniformly at random from strings with length $p(|x|)$.

Observe that if $\mathcal{C} = \text{P}$ then $\exists{\cdot}\text{P} = \text{NP}$ and $\text{BP}{\cdot}\mathcal{C} = \text{BPP}$.

Prove that $\exists{\cdot}\text{BP}{\cdot}\text{P} \subseteq \text{BP}{\cdot}\exists{\cdot}\text{P}$.

Justify any amplification claims you make (but you do not have to reprove the Chernoff bounds). Also, when you claim that you have a BP$\cdot\exists\cdot$P machine M for some language $L \in \exists{\cdot}\text{BP}{\cdot}\text{P}$, make sure you prove both directions of $L \subseteq L(M)$ and $L(M) \subseteq L$.

Does your proof work for $\text{BP}{\cdot}\exists{\cdot}\text{P} \subseteq \exists{\cdot}\text{BP}{\cdot}\text{P}$? Why or why not?

3. Let #SAT(ϕ) be the number of satisfying assignments of a Boolean formula ϕ. We have assumed in class that #SAT is complete for #P. This relies on the fact that there is a version of Cook's reduction from NP computations to SAT that is *parsimonious.* That is, given an NP machine N and an input string x, the parsimonious reduction will construct a Boolean formula ϕ such that $\#\text{acc}_N(x) = \#\text{SAT}(\phi)$. I.e., the number of accepting paths of machine N on input x equals the number of satisfying assignments of ϕ.

Consider the version of Cook's reduction in Theorem 7.37 of the textbook. Is the reduction as presented parsimonious? Justify your answer.