Notation: in these problems, $C$ always denotes a Boolean circuit, and $#C$ denotes the number of input strings that cause $C$ to output 1. Also, if $M$ is a nondeterministic Turing Machine and $x$ is an input, then $#M(x)$ denotes the number of accepting nondeterministic computation paths of $M$ on $x$.

1. (Derandomized restarting via approximate counting.) Show that $\text{BPP}_{\text{path}}$ (the class called “RestartingBPP” in Homework #3, Problem 3b) is a subset of the class $P_{\text{ApproxCount}}$. By the latter, we mean the class of all decision problems solvable in polynomial time given the ability to nonadaptively query an oracle for approximate counting (meaning that, when circuit $C$ is submitted to the oracle, it returns a number $\alpha$ such that $#C \leq \alpha < 2 \cdot #C$).

(Remark: In fact, you might like to try to show that $P_{\text{ApproxCount}} \subseteq \text{BPP}_{\text{path}}$, and thus the two classes are equal.)

2. (An odd problem.) Recall the complexity class $\oplus P$: a language $L$ is in the class if and only if there is a polynomial-time nondeterministic Turing Machine $M$ such that $x \in L$ if $#M(x)$ is odd. As we discussed in class, the Cook–Levin Theorem is “parsimonious”, and therefore the language $\text{ODD-CIRCUIT-SAT} = \{C : $#C$ is odd\}$ is $\oplus P$-complete.

(a) Show that $\oplus P$ is closed under complement and under intersection.

(b) Show that $\oplus P^{\oplus P} = \oplus P$. Here, as usual, $\oplus P^{\oplus P}$ denotes $\bigcup_{A \in \oplus P} \oplus P^A$, and $\oplus P^A$ has the same definition as $\oplus P$ given at the beginning of the problem, except that the machine $M$ has oracle access to language $A$.

3. (Mind the gap.) Recall that $f : \{0,1\}^* \rightarrow \mathbb{N}$ is in the class $\# P$ if there is a polynomial-time nondeterministic Turing Machine $M$ such that $f(x) = #M(x)$ for all $x$. We introduce a new function class called $\text{GapP}$, defined to be all $f : \{0,1\}^* \rightarrow \mathbb{Z}$ such that there is a polynomial-time nondeterministic Turing Machine $M$ with $f(x) = \Delta M(x)$ for all $x$, where $\Delta M(x)$ is defined to be the number of accepting paths minus the number of rejecting paths of $M$ on $x$.

(a) Show that $\text{GapP}$ is the closure of $\# P$ under subtraction. More precisely, show that $\# P \subseteq \text{GapP}$, that every $f \in \text{GapP}$ is the difference of two $\# P$ functions, and that the difference of two $\text{GapP}$ functions is in $\text{GapP}$.

(b) Show that every $f \in \text{GapP}$ can in fact be written as $g - h$, where $g \in \# P$ and $h \in \text{FP}$. (Here $\text{FP}$ is the class of integer-valued functions computable in polynomial time; i.e., those $h$ for which there is a deterministic Turing Machine that on input $x$, prints out $h(x)$ in binary.) Conclude that $\text{GapP} \subseteq \text{FP}^{\# P[1]}$. 

1