Graduate Computational Complexity Theory **CMU 15-855*, Fall 2017**

Homework 6
Due: 10:00am, Tuesday October 24

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 $\mathsf{BPP}_{\mathrm{path}}$ (the class called "$\mathsf{RestartingBPP}$" in Homework #3, Problem 3b) is a subset of the class $\mathsf{P}_{\|}^{\mathrm{ApproxCount}}$. 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 α such that $\#C \le \alpha < 2 \cdot \#C$).

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

2. **(An odd problem.)** Recall the complexity class $\oplus\mathsf{P}$: 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 ODD-CIRCUIT-SAT $= \{C : \#C \text{ is odd}\}$ is $\oplus\mathsf{P}$-complete.

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

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

3. **(Mind the gap.)** Recall that $f : \{0,1\}^* \to \mathbb{N}$ is in the class $\#\mathsf{P}$ 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 GapP, defined to be all $f : \{0,1\}^* \to \mathbb{Z}$ 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 GapP is the closure of $\#\mathsf{P}$ under subtraction. More precisely, show that $\#\mathsf{P} \subseteq \mathsf{GapP}$, that every $f \in \mathsf{GapP}$ is the difference of two $\#\mathsf{P}$ functions, and that the difference of two GapP functions is in GapP.

   (b) Show that every $f \in \mathsf{GapP}$ can in fact be written as $g - h$, where $g \in \#\mathsf{P}$ and $h \in \mathsf{FP}$. (Here 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 $\mathsf{GapP} \subseteq \mathsf{FP}^{\#\mathsf{P}[1]}$.