6.045J/18.400J: Automata, Computability and Complexity Nancy Lynch

Homework 12.2 (Fake)

Due: Never *Elena Grigorescu*

Readings: Sipser, Section 10.2.

Problem 1: Sipser problem 10.11. Let M be a probabilistic polynomial time TM and let C be a language where, for some fixed $0 < \epsilon_1 < \epsilon_2 < 1$,

1. $w \notin C$ implies Pr[M acceps w]$\leq \epsilon_1$
2. $w \in C$ implies Pr[M accepts w]$\geq \epsilon_2$.

Show that $C \in BPP$. (Hint: Use Lemma 10.5)

Problem 2: Define the language class PP as follows: A language $L \in$ PP if and only if there exists a probabilistic polynomial time Turing machine such that:

- If $w \in L$, then $\Pr[M \text{ accepts } w] \geq \frac{1}{2}$.
- If $w \notin L$, then $\Pr[M \text{ accepts } w] < \frac{1}{2}$.

Prove that:

1. BPP $\subseteq$ PP.
2. NP $\subseteq$ PP.
3. PP $\subseteq$ PSPACE.

Hint for (2): Consider a nondeterministic TM for L, and replace rejections with probabilistic decisions.

Problem 3: Use the Fermat test to prove that the following numbers are not prime:

1. 12
2. 15

Problem 4: (Fermat's test) Sipser problem 10.15. Prove Fermat's little theorem. That is, prove that

$$\text{If } p \text{ is prime, and } a \in \mathbb{Z}_p^+, \text{ then } a^{p-1} \equiv 1 \ (\ mod\ p)$$

(Hint: Consider the sequence $a, a^2, \ldots$. What must happen, and how ?)

Problem 5: (Branching program example) Show that the majority function can be computed by a branching program that has $O(n^2)$ nodes.

Problem 6: (Branching program equivalence test)

1. Give a read-once branching program B_1 that computes the function of three Boolean variables, x_1, x_2, and x_3, that has value 1 if and only if exactly one or exactly three of the variables have value 1.
2. Give a different read-once branching program B_2 that computes the same function as in part (a).

3. Compute the polynomials p_1 and p_2 associated with the output 1 box for programs B_1 and B_2, respectively, using the rules given in Sipser's book, p. 378.

4. Choose arbitrary values from Z_7 for the three variables, and evaluate p_1 and p_2 to check that they indeed give the same result.