Problem Set 8 – Due Friday, May 22, 2015

Problem 1. A *probabilistic TM* (PTM) is like an ordinary TM except for having a distinguished state $q_\$$ such that, when it enters this state, it will randomly print a 0 or 1 on its tape, each with probability 0.5. The machine is said to have "flipped a coin." Let $\Pr[M \text{ accepts } x]$ be the probability, over all these coin flips, that M accepts when we start it in configuration (ε, q_0, x). We say that a PTM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_A, q_R, q_\$)$ *decides L with error probability ϵ* if M always halts and $x \in L \Rightarrow \Pr[M \text{ accepts } x] \geq 1-\epsilon$ and $x \notin L \Rightarrow \Pr[M \text{ accepts } x] \leq \epsilon$. We say that PTM M *decides* L if it decides L with error probability 1/4.

1.1 Suppose a PTM M decides L. Describe a (conventional) TM M' that decides L. How much slower is M' than M? (If M takes at most T steps to decide x, how long will M' take to do so?)

1.2 Suppose a PTM M decides L. Describe a PTM M' that decides L with error probability $\epsilon = 10^{-10}$. Make M' run as fast as you can, analyzing its running time with a *Chernoff bound*, say the following one: If $X_1, \ldots, X_n$ are independent $[0,1]$-valued variables and $X = \sum_{i=1}^{n} X_i$ is their sum and $\mu = \mathbf{E}[X]$ is its expected value then $\Pr[X \geq (1+\lambda)\mu] \leq e^{-\lambda^2\mu/(2+\lambda)}$ and for any $\lambda \geq 0$.

1.3* *Extra credit.* Make your algorithm of 1.2 more efficient by choosing a smaller constant—the smallest possible, as established by a computer-supported calculation. That is, identify the smallest constant that will "work" in your algorithm of 1.2 get the requested error probability. Finding his will require some basic combinatorics (stuff you should have learned in ECS 20) plus writing a short computer program. Include your program and a run showing the constant you need.

Problem 2. Guess the classification for each of the following languages as either **recursive**, **r.e.** but not not co-r.e., **co-r.e.** but not r.e., or **neither** r.e. nor co-r.e. No justification is required.

2.1 $\{\langle M, k\rangle : M$ is a TM that accepts at least one string of length $k\}$.

2.2 $\{\langle M, k\rangle : M$is a TM that accepts at most one string of length $k\}$.

2.3 $\{\langle M\rangle : M$ is a TM and M has 50 states$\}$.

2.4 $\{\langle M\rangle : M$ is a TM and $L(M)$ contains a palindrome $\}$.

2.5 $\{\langle M\rangle : M$ is a TM and $L(M) = \emptyset\}$.

2.6 $\{\langle M\rangle : M$ is a TM and $L(M)$ is r.e.$\}$.

2.7 $\{\langle M\rangle : M$ is a TM and $L(M)$ is decidable$\}$.

2.8 $\{\langle G\rangle : G$ is a CFG and $L(G) = \Sigma^*\}$.

2.9 $\{\langle M\rangle : M$ is a C-program that halts on $\langle M\rangle\}$.

2.10 $\{\langle M\rangle :$ M is a TM that sometimes diverges$\}$.

2.11 $\{\langle M, w\rangle : M$ is a TM and M that uses at most 50 tape cells when run on $w\}$.

2.12 $\{\langle G\rangle : G$ is a CFG and G accepts an odd-length string$\}$.

2.13 $\{\langle M\rangle :$ M is a smallest (fewest-state) NFA for $L(M)\}$.

2.14 $\{\langle p\rangle : p$ is a multivariate polynomial and $p(\mathbf{x}) = 0$ for some $\mathbf{x} \in \mathbb{Z}^n\}$.

2.15 $\{\langle p, q\rangle : p$ and q are multivariate polynomials and $p(\mathbf{x}) = q(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{Z}^n\}$.

Problem 3. Say that a language $L = \{x_1, x_2, \ldots\}$ is *enumerable* if there exists a two-tape TM M that outputs $x_1 \sharp x_2 \sharp x_3 \sharp \cdots$ on a designated *output tape*. The other tape is a designated *work tape*. The output tape is write-only, with the head moving only from left-to-right. Say that L is *enumerable in lexicographic order* if L is enumerable as above but where, additionally, $x_1 < x_2 < x_3 < \cdots$, where "$<$" denotes the lexicographic order.

3.1. Prove: L is r.e. iff it is enumerable.

3.2. Prove: L is recursive iff it is enumerable in lexicographic order.