6.045J/18.400J: Automata, Computability and Complexity T.A. Matthew Lepinski

Recitation 13: Randomized Complexity Classes

May 8, 2003

Note: PPT = Probabilistic Polynomial Time. An algorithm is PPT if (a) it is allowed to flip coins and (b) its running time is polynomial for all possible coin tosses.

Randomized Complexity Classes:

- **RP** - $L \in RP$ if there exists a PPT algorithm M such that

$$x \in L \Rightarrow Pr[M(x, y) = ACCEPT] > \frac{2}{3}$$

$$x \notin L \Rightarrow Pr[M(x, y) = ACCEPT] = 0$$

  An RP Algorithm only makes a mistake when x is in the language.

- **co-RP** - $L \in co-RP$ if there exists a PPT algorithm M such that

$$x \in L \Rightarrow Pr[M(x, y) = ACCEPT] = 1$$

$$x \notin L \Rightarrow Pr[M(x, y) = ACCEPT] < \frac{1}{3}$$

  A co-RP Algorithm only makes a mistake when x is not in the language.

- **BPP** - $L \in BPP$ if there exists a PPT algorithm M such that

$$x \in L \Rightarrow Pr[M(x, y) = ACCEPT] > \frac{2}{3}$$

$$x \notin L \Rightarrow Pr[M(x, y) = ACCEPT] < \frac{1}{3}$$

  A BPP Algorithm can make two-sided errors, but the error probability is bounded.

- **PP** - $L \in PP$ if there exists a PPT algorithm M such that

$$x \in L \Rightarrow Pr[M(x, y) = ACCEPT] > \frac{1}{2}$$

$$x \notin L \Rightarrow Pr[M(x, y) = ACCEPT] < \frac{1}{2}$$

  A PP Algorithm can have an error probability arbitrarily close to one-half.

- **ZPP** - $L \in ZPP$ if there exists a randomized algorithm M that correctly decides L and runs in expected polynomial time. A ZPP Algorithm makes no errors, but it need not run in polynomial time for all choices of it's random coins.

- **P/POLY** - $L \in P/Poly$ if there exists an algorithm $M(.,.)$ and a family of advice strings $\{a_1, a_2, \ldots\}$ such that

$$|a_n| < n^c \text{ for some constant c}$$

$$x \in L \Leftrightarrow M(x, a_{|x|}) = ACCEPT$$

  A P/POLY algorithm is always correct but it is allowed a "magic" advice string based on the length of its input.

Facts:

1. The constant $\frac{1}{3}$ in the definitions of RP, co-RP and BPP can be replaced by any constant strictly less than one-half. Additionally, the error probability can be made exponentially close to zero by running the algorithm repeatedly. [From Class]
2. $BBP \subseteq P/POLY$ [From Class]
3. If $P = NP$ then $P = BPP$ [From Class]
4. If $NP = co - NP$ then $BPP \subseteq NP$
5. $RP \subseteq BPP$ and $co - RP \subseteq BPP$
6. $RP \subseteq NP$ and $co - RP \subseteq co - NP$
7. $ZPP \subseteq RP$ and $ZPP \subseteq co - RP$
8. $ZPP = RP \cap co - RP$
9. $NP \subseteq PP$ and $Co - NP \subseteq PP$
10. P/POLY contains undecidable languages

The Following Relationships are Unknown

- Does $P = NP$?
- Does $NP = co - NP$?
- Does P/POLY contain NP?
- Does $NP = RP$?
- Does $P = RP$?
- Does $P = ZPP$?
- Does $P = BPP$?
- Does BPP contain NP?
- Is BPP equal to PP?
- Are BPP and PP both equal to NEXP?