02249 - Computationally Hard Problems, Assignment III

Arianna Taormina, S163671

21 September 2020

Exercise Description

A yes-no–problem is in $RP_{1/10}$ if there is a polynomial p and a randomized p-bounded algorithm A such that for every input X the following holds:

$$i) \text{ True answer for } \boldsymbol{X} \text{ is} \quad \text{YES} \quad \text{then} \quad \boldsymbol{P}_R[A(\boldsymbol{X}, R) = \text{YES}] \geq \frac{1}{10}$$
$$ii) \text{ True answer for } \boldsymbol{X} \text{ is} \quad \text{NO} \quad \text{then} \quad \boldsymbol{P}_R[A(\boldsymbol{X}, R) = \text{NO}] = 1$$

a) Prove that $RP_{1/10}$ = RP

b) Prove that $\mathcal{RP} \subseteq \mathcal{BPP}$.

Exercise Resolution

a) RP problems admit algorithms with one side error (where the true answer is "YES"), meaning that in the case that the True answer is "NO" (ii) then both the RP and $RP_{1/10}$ algorithms always give the right answer.

If the True answer is "YES" (i) and we repeat the p bounded algorithm A from the definition of $RP_{1/10}$ "n" times the probability of having "n" mistakes is:

$$\boldsymbol{P}_R[A(\boldsymbol{X}, R) = \text{NO for each n}] \leq \frac{9^n}{10^n}$$

If we run A 100 times the chance of getting a NO every time is very small, meaning that we are almost sure (99.9973438%) of getting at least one "YES", which is enough because if the true answer is "NO", A would for sure return NO every time. So, by repeating the algorithm "n" time we amplify the probability of success:

$$\boldsymbol{P}_R[A(\boldsymbol{X}, R) = \text{YES at least once}] \geq 1 - \frac{9^n}{10^n}$$

We can conclude that 1/2 used in the definition of RP class is arbitrary and that it can be replaced by any constant p $|0 \leq p \leq 1$ e.g. p = 1/10.

b) We want to prove that the definition of class RP is a restriction of class BPP, or in other words that any RP-algorithm is also a BPP-algorithm. BPP-algorithms can have two-sided error, but they should return a correct answer more than 50% times in both cases (otherwise they wouldn't be more useful than tossing a coin).

A BPP-algorithm A fulfills the following properties:

$$i) \text{ True answer for } \boldsymbol{X} \text{ is} \quad \text{YES} \quad \text{then} \quad \boldsymbol{P}_R[A(\boldsymbol{X}, R) = \text{YES}] \geq \frac{1}{2} + \epsilon$$
$$ii) \text{ True answer for } \boldsymbol{X} \text{ is} \quad \text{NO} \quad \text{then} \quad \boldsymbol{P}_R[A(\boldsymbol{X}, R) = \text{NO}] \geq \frac{1}{2} + \epsilon$$

In the case that the True answer is "NO" (ii) then the RP algorithm always gives the right answer, which satisfies the BPP requirement of :

$$\boldsymbol{P}_R[A(\boldsymbol{X}, R) = \text{NO}] \geq \frac{1}{2} + \epsilon$$

If the True answer is "YES" (i) and we repeat the p bounded algorithm A from the definition of RP "n" times, as already proven above we can boost the probability of success:

$$\boldsymbol{P}_R[A(\boldsymbol{X}, R) = \text{YES at least once}] \geq 1 - \frac{1}{2^n}$$

If we repeat the algorithm a polynomial number of times we can get exponentially close to 1, which fulfills the requirement of:

$$\boldsymbol{P}_R[A(\boldsymbol{X}, R) = \text{YES}] \geq \frac{1}{2} + \epsilon$$