Problem

Define a ZPP-***machine*** to be a probabilistic Turing machine that is permitted three types of output on each of its branches: *accept*, *reject*, and *?*. A ZPP-machine M decides a language A if M outputs the correct answer on every input string w $(\mathit{accept}$ if $w \in A$ and reject if $w \notin A)$ with probability at least $\frac{2}{3}$, and M never outputs the wrong answer. On every input, M may output *?* with probability at most 1/3 . Furthermore, the average running time over all branches of M on w must be bounded by a polynomial in the length of w. Show that RP ∩ coRP = ZPP, where ZPP is the collection of languages that are recognized by ZPP-machines.

Step-by-step solution

Step 1 of 2

- $RP \bigcap co-RP \subseteq ZPP$. Consider $L \in RP \bigcap co-RP$.

Then assume A be an RP algorithm for L and assume B be an $co-RP$ algorithm for L.

• Consider w be the input. One step of our ZPP algorithm will be the following:

1. Run A on w.

2. Then run B on w .

3. If A accepts, accept.

4. If B reject, reject.

Note that this step takes polynomial time since each A and B take polynomial time.

5. If A rejects and B accepts, repeat.

• First, show correctness.

Note that if $w \notin L$ then A always rejects, so if A accepts $w \in L$.

Similarly, if B reject $w \notin L$. Thus, when output an answer, it is we always correct (which is better than probability $\frac{2}{3}$). Note that our algorithm never outputs $?^1$.

• Then, we prove that the running time is polynomial in expectation. Note that since each step has polynomial running time, it suffices to show the number of steps is polynomial in expectation (in fact, we will show it is constant in expectation). Note that if $w \in L$, then A will accept with the probability at least $\frac{1}{2}$ (by definition of RP). Similarly, for $w \notin L$ and B reject. Thus, every step succeeds with the probability of at least $\frac{1}{2}$.

• Assume X is a random variable denoting how many steps to take. Then let $p_k = \Pr[X = k]$ denote the probability by taking exactly k steps $k-1$. Note that p_k is the probability that we fail on the first $k-1$ steps and then succeed, so

$$E|X| = \sum_{k=1}^{\infty} \frac{k}{2^{k-1}} = 4$$

And this is done^2.

Comment

Step 2 of 2

^{1}In fact, ZPP actually stands for zero-error probabilistic polynomial-time. This comes fact that it is equivalent to define ZPP algorithms as never outputting a wrong answer.

^{2}General Formula can be drive to do last sum $(\text{for} |x| < 1)$:

$$\begin{aligned}\sum_{k=1}^{\infty} kx^{k-1} &= \sum_{k=1}^{\infty}\left[\frac{d}{dx}x^k\right] \\ &= \frac{d}{dx}\left[\sum_{k=1}^{\infty} x^k\right] \\ &= \frac{d}{dx}\frac{1}{(1-x)} \\ &= \frac{1}{(1-x)^2}\end{aligned}$$

• $ZPP \subseteq RP$ (It can be proved that $ZPP \subseteq co-RP$ analogously) consider L be in ZPP, and A be an ZPP algorithm for L, and consider $p(n)$ be the expected running time of A on input of length n. Then build the following RP algorithm:

• On input w of length n,

1. Run A on w for time $6p(n)$.

2. If A accepts or rejects, output the answer A.

3. Otherwise (If A does not terminate, or outputs $?$), reject.

• Note that since A has expected polynomial running time $p(n)$ is a polynomial, so $6p(n)$ is also polynomial. It suffices to prove correctness. First, if $w \notin L$, then A will either reject, print $?$ or not terminate. In all of these cases, reject, so it is correct.

• On the other hand, assume $w \in L$. Again, three things can happen: A accepts, A prints $?$, or A does not terminate in time. We are correct in the first two cases, so it suffices to show the probability of the latter two are at most $\frac{1}{2}$. The probability that A outputs $?$ is at most $\frac{1}{3}$.

• The probability that A does not terminate in 6p(n) time is at most $\frac{1}{3}$ (by Markov's inequality). Thus, if $w \in L$, the probability that A is wrong is at most $\frac{1}{2}$, thus this algorithm is an RP algorithm for L.

Comment