CS 1511 Homework 14

Mathew Varughese, Justin Kramer, Zach Smith

Wednesday, March 6

26. a) If a language is in ZPP, then the polynomial-time PTM M can prove that the language is in ZPP.

For this Turing Machine, it will either output the correct answer or "DON'T KNOW".

There is at most a 1/2 chance that it outputs "DON'T KNOW". If the machine is run once, then the chance that it outputs this is at most 1/2, with at least a 1/2 chance of outputting the correct answer.

If the machine is run twice, there is at most a 1/4 chance that it doesn't output the correct answer.

If you take this to a limit of infinity, there is no chance that you do not eventually output the correct answer when you run the machine an infinite amount of times. It will take poly-time to run, so it doesn't matter if we simulate this machine for many many times.

In the other direction, we can use Markov's inequality to show that with our PTM M, L is in ZPP.

26. b)

27 Lemma 7.12 says that A coin with Pr[Heads] = can be simulated by a PTM in expected time O(1) provided the ith bit of is computable in poly(i) time.

We can construct this p such that a random coin that comes up with this probability will make a Turing Machine able to decide a language in undecidable time.

Assume that this is possible to make a PTM that simulates a coin flip with probability p. The ith bit of p can be computed in constant time.

The p is comprised of 1s and 0s. The turing machine in question (say TM T) generates a random "x". This x is a random variable. Then we compare x to that probability p. We do this by comparing bit by bit. If p is 0.0101010 and x is 0.0101011, we know that x has a greater probability than p.

We make each bit of this p represent whether a Turing Machine will halt on a particular language. So, order every Turing Machine and input w like (M,w) and correspond it with a integer. The ith integer in p will be a 1 if M halts on w and 0 if not.

Then, the Turing Machine T could solve the halting problem. To figure out if M halts on w, it would first correspond that M,w pair to an integer. Call this integer i. The ith digit of p would tell us what the answer to this question is. So we flip We flip a coin i times. We count the distribution of heads and tails. We can say with p confidence whether or not M halts w.

28 First we show NP $\in$ PP. Then we show coNP $\in$ PP. This will prove that NP $\cup$ coNP $\in$ PP.

If L is in NP, then there exists a TM T that verifies an answer to L in a polynomial number of steps.

For every x, x is in L, iff $\exists y$ T(x, y) accepts.

So build a T' so it tries w values for y. If this accepts, accept. If not, flip a coin. The coin should biased so it gives heads slighter less than half of the time. This "slighter half" will be of size $w/2^n$. If the coin returns heads, accept. Otherwise reject. This will make it so if an x is in L, it will accept with more than 1/2 probability.

So, NP is in PP.

CoNP is in PP for a similar reason. If L is in CoNP, there exists a TM T that halts on an answer L in a polynomial number of steps.

For every x, x is in L, iff $\forall y$ T(x, y) accepts.

So, x is not in L if there exists a y that T(x, y) rejects. We apply the same logic of the previous problem. We try some inputs for y. If they cause it to fail, than we reject. Then, we flip a biased coin that gives heads slightly higher than 1/2 of the time. Then we reject.