# CS 1511 Homework 18

Mathew Varughese, Justin Kramer, Zach Smith

Sunday, March 25

**33. a)** The verifier could simulate say 1000 trials of the prover and pick the majority answer. Since the verifier is a BPP machine it still will error with low probability if the prover sends an incorrect result, the verifier's work is not changed.

**33. b)** We have polynomial space so verifier writes down every possible thing to ask. This can be done in polynomial space. Both the prover and verifier are polynomial in the run time, which is in PSPACE by the space-time hierarchy.

**33. c)** IP is in PSPACE. So, changing the 2/3 to 1 does not matter. Now we simply have something close to a RP machine, which is actually a subset of the BPP machine. Therefore it is still the same thing, since it runs like a BPP machine that has been run many times.

**33. d)** This means that the verifier always rejects when a string is not in the language. Since the verifier can ask a polynomial number questions, this means that enough trials can be made to know if it is the language. So, IP' = NP since we can verify with a poly-time verifier and with enough trials we can get a definite yes-no answer.

**34** Prove that there exists a perfectly complete $AM[O(1)]$ protocol for proving a lower bound on set size.

In our current situation, our protocol for proving a lower bound on set size takes a polynomial amount of time most of the time, but it can take a constant amount of time sometimes.

We begin with a set $S^l$, with l times the Cartesian product of our set S which we are using the length of to determine K.

In this case, we suggest that there exists several sets of hash functions $h_1, h_2, ...h_i$ such that at a certain size $\cup_i hi(S^l) = 0, 1^{kl}$ allows us to have the prover correctly give a set size lower bound in constant time everytime. This union of hash functions allows us to distinguish if $\mid S \mid \geq K$ or if $\mid S \mid \leq K/c$.