Perfect zero knowledge in $AM \cap co\text{-}AM$

Johan Håstad

930217

Abstract

The purpose of the present note is to provide some intuition on the paper "Statistical zero-knowledge languages can be recognized in two rounds" by Aiello and Håstad [1]. To keep the presentation simple we assume that we are dealing with perfect zero knowledge and that the simulator always terminates in polynomial time (irrespective of its random coins).

1 Introduction

The purpose of this report is to provide some intuition for readers of the paper [1]. One reason for this is that since there has been an error discovered in the proof of [2], (see [3]), the proof in [1] is now the only published proof also for the fact that the complement of statistical zero knowledge languages can also be recognized in a constant number of rounds.

First a small warning. We assume familiarity with the notation of [1] and we make no attempt to be accurate in a formal sense. This is only a couple of pages to help the reader.

2 The discussion

We assume the simulator produces accepting conversations which give a possible r for the coins of the verifier with probability at least $1/2$.

For any conversation s let s_i be the conversation upto the ith move by the verifier and let β_i be the ith move by the prover.

Now for any partial conversation s_i (ending with a verifier move) let $P(\beta_i, s_i, r)$ be the probability that the next move by the prover (as given by the simulator) is β_i given that the initial conversation is s_i and that the verifier coins take the value r. Furthermore let $Q(r)$ be the probability that the verifier coins take the value r. The probability that the simulator produces a given conversation s is now:

$$Q(r)\prod_{i=1}^{k} P(\beta_i, s_i, r)$$

Now consider the following prover: Given partial conversation s_i it generates a β_i according to the same distribution as the simulator. Let us denote this probability by $S(\beta_i, s_i)$. The probability that the verifier together with this prover generates conversation s is

$$2^{-|r|}\prod_{i=1}^{k} S(\beta_i, s_i).$$

If we let $A_s = Q(r)\prod_{i=1}^{k} P(\beta_i, s_i, r)$ and $B_s = 2^{-|r|}\prod_{i=1}^{k} S(\beta, s)$. Then if we restrict summation over only accepting conversations then

$$\sum_s A_s \geq 1/2.$$

If $x \in L$ then $A_s = B_s$ while when $x \notin L$ then

$$\sum_s B_s \leq 2^{-6nk}$$

Now we try to estimate

$$\sum A_s \log A_s$$

and

$$\sum A_s \log B_s$$

Their difference either being 0 (when $x \in L$) or at least $3nk - 1$ (when $x \notin L$). The former follows since when $x \in L$, $A_s = B_s$ while when $x \notin L$, then, by looking at gradients, it is not hard to see that the maximal value of $\sum A_s \log B_s$ given $\sum B_s \leq 2^{-6nk}$ is achieved by $B_s = cA_s$ for some constant c which is independent of s. Now, note that

$$\log A_s = \log Q(r) + \sum \log P(\beta_i, s_i, r)$$

and

$$\log B_s = -r + \sum \log S(\beta_i, s_i).$$

The idea is to sample according to the probability distribution given by the simulator and estimate the two averages from below and above.

If we take a large enough product (i.e. many conversations in parallel) then $\log P(\beta_i, s_i, r)$ does not depend too much on β_i, s and r but only on i, and hence these numbers can be specified in advance. A similar statement is true for $\log S(\beta_i, s_i)$. The protocol gives these numbers and then all that is needed is to generate conversations (by the simulator) and verify that these probabilities are within the prescribed bounds.

If we want to prove that $x \in L$ we prove upper bounds on $\log P(\beta_i, s_i, r)$ (which is an upper bound on the number of simulator coins giving β_i, s_i, r together with a lower bounds of the number of coins giving s_i and r) and lower bounds on $\log S(\beta_i, s_i)$. An important point to note is that we need not do both for the same s, since the answer is independent of s we can use different values in the two protocols. Otherwise we would have problems since if we use the same s in two different protocols the properties of the protocols are not preserved.

If we want to prove that $x \notin L$ we prove lower bounds for $P(\beta_i, s_i, r)$ and upper bounds for $S(\beta_i, s_i)$.

The protocol for $x \notin L$ is more efficient since the prover can point to a specific i and then prove that there is a difference there. To prove that $x \in L$ we need to prove that there is (almost) equality for all i.

References

[1] W. Aiello and J. Håstad "Statistical Zero-Knowledge Languages can be Recognized in Two Rounds", Journal of Computer and System Sciences, Vol 42, 1991, pp 327-345.

[2] L. Fortnow "The complexity of Perfect Zero-Knowledge", Advances in Computing Research (ed. S. Micali), Vol 5, Randomness and Computation pp 327-344.

[3] O. Goldreich, R. Ostrovsky, and E. Petrank "Computational Complexity and Knowledge Complexity", 26th STOC, 1994, pp 534-543.