On the Knowledge Complexity of $\mathcal{NP}$

Erez Petrank* Gábor Tardos†

Abstract

We show that if a language has an interactive proof of logarithmic statistical knowledge-complexity, then it belongs to the class $\mathcal{AM} \cap \text{co}-\mathcal{AM}$. Thus, if the polynomial time hierarchy does not collapse, then $\mathcal{NP}$-complete languages do not have logarithmic knowledge complexity. Prior to this work, there was no indication that would contradict $\mathcal{NP}$ languages being proven with even one bit of knowledge. Our result is a common generalization of two previous results: The first asserts that statistical zero knowledge is contained in $\mathcal{AM} \cap \text{co}-\mathcal{AM}$ [AH-87], while the second asserts that the languages recognizable in logarithmic statistical knowledge complexity are in $\mathcal{BPP}^{\mathcal{NP}}$ [GOP-94].

Next, we consider the relation between the error probability and the knowledge complexity of an interactive proof. Note that reducing the error probability via repetition is not free: it may increase the knowledge complexity. We show that if the error probability $\epsilon(n)$ is less than $2^{-3k(n)}$ (where $k(n)$ is the knowledge complexity) then the language proven has to be in the third level of the polynomial time hierarchy. In the standard setting of negligible error probability, there exist PSPACE-complete languages which have linear knowledge complexity. However, if we insist, for example, that the error probability is less than 2^{-n^2}, then PSPACE-complete languages do not have sub-quadratic knowledge complexity, unless PSPACE= Σ_3^P.

In order to prove our main result, we develop an AM protocol for checking that a samplable distribution D has a given entropy h. For any fractions ϵ, δ, the verifier runs in time polynomial in $1/\delta$ and $\log(1/\epsilon)$ and fails with probability at most ϵ to detect an additive error δ in the entropy. We believe that this protocol is of independent interest.

*Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 3G4. E-mail: erez@cs.toronto.edu

†Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 3G4 and Mathematical Institute of the Hungarian Academy of Sciences, Pf. 127, Budapest, H-1364 Hungary. Partially supported by NSF grant CCR-95-03254 and OTKA-F014919. E-mail: tardos@cs.elte.hu

1 Introduction

The notion of knowledge-complexity was introduced in the seminal paper of Goldwasser Micali and Rackoff [GMR-85, GMR-89]. Knowledge-complexity is intended to measure the *computational advantage* gained by interaction. A formulation of knowledge-complexity, for the case that it is not zero, has appeared in [GP-91]. A very appealing suggestion, actually made by Goldwasser Micali and Rackoff, is to characterize languages according to the knowledge-complexity of their interactive proof systems [GMR-89].

The class of knowledge complexity 0 (better known as zero knowledge) stands at the lowest level of the knowledge complexity hierarchy, and at the top we have the class of languages with polynomial knowledge complexity which includes all IP=PSPACE. Both for zero-knowledge as for the knowledge complexity in general, there are three standard variants of the definitions which result in three hierarchies of languages; that is, *perfect*, *statistical* and *computational*. In this paper we will be only interested in the statistical and perfect hierarchies.

Our main result is a relation between the knowledge complexity and the computational complexity of languages. We show that languages with logarithmic knowledge complexity are in $\mathcal{AM} \cap \text{co}-\mathcal{AM}$. This result has a very interesting implications on languages in $\mathcal{NP}$. Recall that if $\mathcal{NP} \subseteq \text{co}-\mathcal{AM}$ then the polynomial time hierarchy collapses [BHZ-87]. Assuming that the polynomial time hierarchy does not collapse, we get that $\mathcal{NP}$-complete languages do not have logarithmic knowledge complexity. Prior to our result, there was no indication that would contradict all $\mathcal{NP}$ languages having knowledge complexity 1. Note that, if a one-way function exists, then this differs significantly from the *computational* knowledge complexity hierarchy for which NP-complete languages have zero knowledge interactive proofs (and so do PSPACE-complete languages) [GMW-86, IY-87, B+ 88].

1.1 Background on knowledge-complexity

Loosely speaking, an interactive-proof system for a language L is a two-party protocol, by which a powerful *prover* can "convince" a probabilistic polynomial-time *verifier* of membership in L, but will fail (with high probability) when trying to fool the verifier into "accepting" non-members [GMR-89]. An interactive-proof is called *zero-knowledge* if the interaction of any probabilistic polynomial-time machine with the predetermined prover, on common input $x \in L$, can be "simulated" by a probabilistic polynomial-time machine (called the *simulator*), given only x [GMR-89]. We say that a probabilistic machine M *simulates* an interactive proof if the output distribution of M is *statistically close* to the distribution of the real interaction between the prover and the verifier.

The formulation of zero-knowledge presented above is known as *statistical* (almost-perfect) zero-knowledge. Alternative formulations of zero-knowledge are *computational* zero-knowledge and *perfect* zero-knowledge. In this paper we concentrate on statistical zero-knowledge and the knowledge-complexity hierarchy that generalizes it.

Loosely speaking, the knowledge-complexity of a protocol Π is the best possible "quality" of an efficient simulation of Π. Namely, we say that a prover leaks $k(n)$ bits of knowledge to the verifier if there is a probabilistic polynomial-time machine ("simulator") M such that on any input $x \in L$, the machine M on input x, outputs a distribution, of which a subspace of density at least $2^{-k(|x|)}$ is statistically close to the distribution of the conversations in the interaction between the prover and the verifier. For a formal definition and further discussion, see Subsection 2.2.

We say that a language L has knowledge complexity $k(|x|)$ if there is an interactive proof for L with knowledge complexity $k(|x|)$. We consider the knowledge-complexity of a language to be a very natural parameter, and we consider the question of how this parameter relates to the complexity

of deciding the language to be fundamental.

1.2 Previous work

The complexity of recognizing zero-knowledge languages was first considered by Fortnow [F-89]. Building on his work, Aiello and Hastad [AH-87] showed that zero knowledge languages are in $\mathcal{AM} \cap \text{co}-\mathcal{AM}$.

Bellare and Petrank [BP-92] bounded the computational complexity of languages which have *short* interactive-proofs with *low* knowledge-complexity. Goldreich, Ostrovsky, and Petrank [GOP-94] have extended this result showing that any language of logarithmic knowledge-complexity can be recognized in $\mathcal{BPP}^{\mathcal{NP}}$. This was the first relation found between a knowledge complexity of a language (above zero) and its computational complexity. Their result gave the first indication that PSPACE-complete languages do not have low (i.e., logarithmic) knowledge complexity.

Goldreich, Ostrovsky, and Petrank have also showed that the difference between the hierarchy of languages classified according to their *perfect* knowledge complexity and the hierarchy of languages classified according to their *statistical* knowledge complexity is not big. They showed how to transform interactive proofs of *statistical* knowledge-complexity $k(n)$ into interactive proofs of *perfect* knowledge-complexity $k(n) + O(\log n)$. This transformation refers only to knowledge-complexity with respect to the honest verifier.

Aiello, Bellare, and Venkatesan [ABV-95] studied the class of languages which have $k(n)$ knowledge complexity *on the average* (see [GP-91, ABV-95] for a definition of knowledge complexity on the average). They showed that languages with logarithmic *average* knowledge complexity are in $\mathcal{BPP}^{\mathcal{NP}}$. They also showed a closer relation between the perfect and the statistical hierarchies of languages (for the case of average knowledge complexity). They showed that the difference between the hierarchies can be bounded by a negligible fraction. This result is also stronger in the sense that it is not restricted to the honest verifier simulation. We remark that it is not known how to get such a close relation for the worst-case knowledge complexity.

1.3 This work

Our main result is that languages having interactive proofs with logarithmic knowledge-complexity are in $\mathcal{AM} \cap \text{co}-\mathcal{AM}$. The class $\mathcal{AM}$ is the class of languages that have two round Arthur Merlin proofs, or equivalently, have a constant round interactive proof. (There is no restriction on the knowledge complexity of this constant round interactive proof.) See [BM-88, GS-89] for definitions of Arthur Merlin proofs, for some basic properties, and for the equivalence of the definitions.

It was shown in [BHZ-87] that if $\mathcal{NP} \subseteq co-\mathcal{AM}$ then the polynomial time hierarchy collapses. It is believed that the polynomial time hierarchy does not collapse, and under this assumption, our result implies that $\mathcal{NP}$-complete languages do not have logarithmic knowledge complexity. Prior to this result, there was no indication that would contradict all of the languages in $\mathcal{NP}$ having knowledge complexity 1.

Our second result involves the connection of the soundness error probability of the interactive proof and the knowledge complexity of the language. We show that if a language has an interactive proof with error probability $\epsilon(n)$ and knowledge complexity $k(n)$ and if $\epsilon(n) \leq 2^{-3k(n)}$ then the language is in $\mathcal{AM}^{\mathcal{NP}}$ and so it is contained in the third level of the polynomial time hierarchy. Let us say a few words on the implication of this result.

In the regular setting it does not matter in the definition of interactive proofs if we allow the error-probability to be as high as $1/3$ or if we insist that it is as small as 2^{-n^3}. However, reducing the error probability of an interactive proof involves repeated operations of the interactive proof and

thus may increase its knowledge complexity. Therefore, when discussing the knowledge complexity, it seems important to fix the error probability to some predetermined function.

Previous works have chosen the reasonable requirement that the error probability be *negligible* (i.e., an error probability that is asymptotically smaller than any polynomial fraction). Our result implies that if one fails to set the relations between these two measures, then some unnatural assertions hold. For example, one cannot expect to have an interactive proof for a complete language in PSPACE to have knowledge complexity $k(n)$ and error probability less than $2^{-3k(n)}$, unless PSPACE is contained in the third level of the polynomial time hierarchy.

Another aspect of this result concerns the trade-off between reducing the error and increasing the knowledge complexity. Many past works considered the possibility of reducing the error of a probabilistic algorithm while not increasing the number of coin-tosses as much as the naive solution would. It would seem natural to ask the same question about the knowledge complexity. In the naive method, we repeat the protocol t times, so the knowledge complexity increases by a factor of t and the error probability (for simplicity assume one-sided error) decreases from ϵ into ϵ^t. Namely, the logarithm of $1/\epsilon$ and the knowledge complexity increase by the same factor. Assuming PSPACE$\neq \Sigma_3$, and in light of our result, one shouldn't expect to have a general method for doing much better than that. Namely, the logarithm of $1/\epsilon$ cannot increase substantially more rapidly than the knowledge complexity.

1.4 Implications on the hint knowledge complexity:

Another implication of our second result concerns a rather esoteric definition of knowledge complexity called the *Hint* version of knowledge complexity. This definition was presented in [GP-91] and was adequate in different scenarios (see [BCK]). Loosely speaking, an interactive proof has knowledge complexity $k(n)$ in the hint sense, if there is a function $h(x)$ of the input (the hint function) such that the interactive proof on input x can be simulated efficiently given only x and the hint $h(x)$, and $|h(x)| \leq k(n)$. (The difference is that the "help" which the simulator gets does not depend on the random coin-tosses of the verifier (or of the simulation). For an exact definition and detailed explanations see [GP-91].)

It was shown in [GP-91] that this definition does not seem to be adequate, especially, because some protocols in which only a polynomial number of bits are transferred, have exponential knowledge complexity. Here, we claim that we can make a similar assertion for languages. Namely, our result implies that a PSPACE-complete language has super-polynomial knowledge complexity in the hint sense unless PSPACE= Σ_3^P. This counter-intuitive assertion gives yet another indication that the hint measure is not an adequate one.

To see that the above assertion is correct, note that the hint measure does not increase when one uses sequential repetitions of the protocol. Also, note that if a protocol has knowledge complexity $k(n)$ in the hint measure, then it also has at most $k(n)$ knowledge complexity in the standard (fraction) measure considered here. Combining these two properties, we get that if a language has an interactive proof with polynomial hint knowledge complexity $k(n)$ and some constant error probability, then this language also has an interactive proof with $k(n)$ knowledge complexity in the standard measure with error probability $2^{-3k(n)}$ and thus this language is in the third level of the polynomial hierarchy.

1.5 Techniques used

We begin by establishing a separation property which separates x's in the language from x's not in the language. This property is a modification of the separation property used in [AH-87]. Next,

we have to show that this separation can be detected by an AM protocol. For this, we employ the lower and upper bounds on set sizes as presented by [GS-89, F-89], and build on them an AM approximation for the entropy of the output distribution of the simulator. We believe that the protocol for approximating the entropy of a probabilistic machine is of independent interest.

In order to prove the validity of the separation property, we use techniques developed in [GOP-94] which relate the distribution of conversations in the original interactive proof with a hypothetical distribution of conversations that occur when we let the original verifier interact with a simulation based prover, i.e., a prover that acts like the prover in the simulation. (see Subsection 2.3 for a formal definition of this prover).

Our main result is proven for *perfect* knowledge complexity and we employ a result from [GOP-94] asserting that the distance between perfect and statistical knowledge complexity is close enough for our result to hold for statistical knowledge complexity as well.

In our second result which relate the knowledge complexity and the error probability we also employ techniques for deterministic bounds on set sizes developed in [Si-83, St-83, JVV-86, BP-92].

1.6 Organization

In Section 2 we give the definitions and notations we use in the paper. In Section **??** we present the machinery needed for the proof: We present our result on approximating the entropy of a polynomially samplable distribution and also the property we use to recognize a language which can be proven in low knowledge complexity. We then discuss the relation of this work to [AH-87] in Section Section 4. In Section 6 we use the above tools to present our main result: a constant round interactive proof for recognizing languages in $\mathcal{KC}(O(\log n))$. In section 7 we present our result relating error probability to knowledge complexity of interactive proofs.

2 Preliminaries

Let us state some of the definitions and conventions we use in the paper. Throughout this paper we use n to denote the length of the input x. A function $f : \mathsf{N} \to [0,1]$ is called *negligible* if for every polynomial p and all sufficiently large n $f(n) < \frac{1}{p(n)}$. Let the distance between distributions D^1 and D^2 be

$$d(D^1, D^2) = \frac{1}{2} \sum_r |\mathrm{Prob}_{D^1}[r] - \mathrm{Prob}_{D^2}[r]|.$$

We say that an ensemble of distributions D^1_x is statistically close to another ensemble D^2_x over a language L, if the function

$$f(n) = \max_{|x|=n,\ x \in L} \{d(D^1_x, D^2_x)\}$$

is negligible.

2.1 Interactive proofs

We begin by recalling the definitions of interactive proofs presented by [GMR-89, B-85]. For formal definitions and motivating discussions the reader is referred to [GMR-89]. An interactive proof is a protocol in which a (computationally unbounded, probabilistic) *prover* P is interacting with a (probabilistic polynomial-time) *verifier* V. Intuitively, the goal of the prover is to prove to the verifier V that a given input is in a predetermined language. Formally, we say that the pair P and V constitutes an interactive proof for a language L if there exists a negligible function $\epsilon : \mathsf{N} \to [0,1]$ such that

1. **Completeness:** If $x \in L$ then

$$\Pr\left[(P,V)(x) \text{ accepts}\right] \geq 1 - \epsilon(n)$$

2. **Soundness:** If $x \notin L$ then for any prover P^*

$$\Pr\left[(P^*,V)(x) \text{ accepts}\right] \leq \epsilon(n)$$

2.2 Knowledge Complexity

Let us define the statistical (and perfect) knowledge complexity measure of protocols (and specifically of interactive proofs). We use the *fraction* definition of knowledge complexity as presented by [GP-91]. For further intuition and motivation see [GP-91].

Throughout the rest of the paper, we only refer to knowledge-complexity *with respect to the honest verifier*; namely, the ability to simulate the honest verifier's view of its interaction with the prover. (In the stronger definition, one considers the ability to simulate the point of view of *any efficient verifier* while interacting with the prover.) This restriction only strengthen the results presented in the paper.

Let $(P,V)(x)$ be the random variable that is distributed according to the verifier's view of the (probabilistic) interaction between P and V on the input x. The view contains the verifier's random tape as well as the sequence of messages exchanged between P and V.

In order not to have to distinguish the view of the interaction from the conversation itself we insist throughout the paper that the verifier ends the conversation with sending his random coins as the last message. Note that it is important to include the coins of the verifier in the output of the simulation, and calling this the last round of the interaction is just an encoding. For simplicity we also require that the verifier starts the conversation.

We call a conversation valid if all the moves by the verifier are consistent with its coin-flips (as given in the last message). We denote by c_i the i round prefix of a conversation c.

By the *fraction formulation* of knowledge complexity, we say that a protocol has knowledge complexity $k(n)$ if there exists an efficient simulation of the protocol that "partially" succeeds in simulating the protocol. (A "fully successful" simulation implies that the protocol is zero knowledge.) The exact interpretation of "partially successful" is that in order to show that the knowledge complexity is $k(n)$, the simulator must have a subspace of its output distribution which is of density $2^{-k(n)}$, and which simulates the protocol "successfully". The interpretation of a successful simulation would be "exactly equal distributions" for *perfect* knowledge complexity, and "statistically close distributions" for *statistical* knowledge complexity.

We follow with the formal definition. In the definition we prefer to talk about a subspace of the random tapes of the simulator rather than to talk about a subspace of the output distribution of the simulator. Although the meaning is the same, it will be easier to work with this definition when proving properties of knowledge complexity.

Definition 2.1 (knowledge-complexity — fraction version): *Let* ρ: $\mathsf{N} \to (0,1]$. *We say that an interactive proof* (P,V) *for a language* L *has* perfect (resp., statistical) knowledge-complexity $\log_2(1/\rho(n))$ in the fraction sense *if there exists a probabilistic polynomial-time machine* M *with the following good subspace property. For any* $x \in L$ *there is a subset of* M*'s possible random tapes, denoted* S_x*, such that:*

1. *The set* S_x *contains at least a* $\rho(n)$ *fraction of the set of all possible coin tosses of* $M(x)$.

2. *Conditioned on the event that $M(x)$'s coins fall in S_x, the random variable $M(x)$ is* identically distributed (resp., statistically close) *to $(P,V)(x)$. Namely, for the perfect case this means that for every $\bar{c}$*

$$\text{Prob}(M(x,\omega)=\bar{c} \,|\, \omega \in S_x) = \text{Prob}((P,V)(x)=\bar{c})$$

where $M(x,\omega)$ denotes the output of the simulator M on input x and coin tosses sequence ω.

Note that the definition of statistical knowledge complexity zero (i.e., when $k = 0$) exactly matches the definition of statistical zero knowledge as given in [GMR-89]. For further motivation and discussion of zero knowledge, the reader is referred to [GMR-89]. From the above definitions of knowledge complexity combined with the definitions of interactive proofs, the knowledge complexity classes of languages can be formulated:

Definition 2.2 (knowledge-complexity classes):

- $\mathcal{PKC}(k(n))$ = *languages having interactive proofs of perfect knowledge-complexity $k(n)$.*
- $\mathcal{SKC}(k(n))$ = *languages having interactive proofs of statistical knowledge-complexity $k(n)$.*

A connection between the perfect and the statistical hierarchies was given in [GOP-94]:

Theorem 1 *[GOP-94]: For the case of the honest verifier simulation*

$$\mathcal{SKC}(k(n)) \subseteq \mathcal{PKC}(k(n) + \log n)$$

The case of honest verifier simulation suffices for the use we make of it in this paper.

2.3 The simulation based prover

An important ingredient in our proof is the notion of a simulation based prover, introduced by Fortnow [F-89]. Consider a simulator M that outputs conversations of an interaction between a prover P and a verifier V. We define a new prover P_M, called *the simulation based prover*, which selects its messages according to the conditional probabilities induced by the simulation. Namely, on a partial history h of a conversation, P_M outputs a message α with probability

$$\text{Prob}(P_M(h)=\alpha) \stackrel{\text{def}}{=} \text{Prob}(M_{|h|+1}=h\circ\alpha \;\big|\; M_{|h|}=h)$$

where M_t denotes the distribution induced by M on t-long prefixes of conversations. (Here, the length of a prefix means the number of messages in it.)

In perfect zero knowledge if $x \in L$ then P_M equals the original prover P. It is important to note however that the behavior of P_M is *not* necessarily close to the behavior of P if the knowledge-complexity is greater than 0. This is the main reason why the AM protocol presented by [AH-87] for the case of zero knowledge is inappropriate for the case of higher (even 1) knowledge complexity.

2.4 Three distributions used throughout the paper

Let us define three distributions which are going to be used in all that follows. These are distributions on conversations as output by running a protocol or invoking the simulator. Here P and V constitute an interactive proof for some language L, M is a simulator for this interaction, and P_M is the simulation based prover (see Subsection 2.3). We consider the following three distributions:

1. The distribution of conversations output by the simulator. We denote the probability that a conversation c is output by the simulator by $\text{Prob}_M[c]$.

2. The distribution of conversations in the original interactive proof (P,V). We denote the probability that a conversation c is output by this distribution by $\mathrm{Prob}_{(\mathrm{P,V})}[c]$.

3. Last, we consider the interaction between the simulation based prover P_M and the original verifier V. We denote the probability that a conversation c is output by this interaction by $\mathrm{Prob}_{(\mathrm{P_M,V})}[c]$.

For the case of perfect knowledge complexity, an immediate connection between the first and the second distributions follows from the definitions. For any transcript c we have $\mathrm{Prob}_{\mathrm{M}}[c] \geq 2^{-k(n)}\mathrm{Prob}_{(\mathrm{P,V})}[c]$.

Consider now the probability of a conversation c in the third distribution. Let t(n) be the length of the random tape used by V, and let d(n) be the number of rounds. For a valid transcript c we may separate the probabilities of the verifier steps (which have probability $2^{-t(n)}$ since the random tape of the verifier appears in the conversation), and the probabilities of the prover (which are induced by the simulator) and we may write:

$$\mathrm{Prob}_{(\mathrm{P_M,V})}[c] = 2^{-t(n)} \cdot \prod_{i=1}^{\frac{d(n)-1}{2}} \frac{\mathrm{Prob}_{\mathrm{M}}[c_{2i}]}{\mathrm{Prob}_{\mathrm{M}}[c_{2i-1}]} \tag{1}$$

For an invalid conversation c we trivially have $\mathrm{Prob}_{(\mathrm{P_M,V})}[c] = 0$. This simple rewriting of $\mathrm{Prob}_{(\mathrm{P_M,V})}[c]$ was first noted in [AH-87].

3 Approximating the entropy in a constant number of rounds

Our first tool is an AM protocol for verifying the entropy of a polynomial samplable distribution to within an accuracy of $1 + \frac{1}{\mathrm{poly}}$. We consider this protocol to be of independent interest. We begin by explaining the setting, and then we state the theorem and prove it.

Let f be any function taking a (uniformly chosen) n-bit string as input, and we let $\mathrm{Prob}_x(f(x) = y)$ denote the probability that $f(x) = y$ when we uniformly sample $x \in \{0,1\}^n$. The entropy $H(f)$ of f is defined as:

$$H(f) = -\sum_y \mathrm{Prob}_x(f(x) = y) \log \mathrm{Prob}_x(f(x) = y)$$

where the sum extends for all values y in the range of f. Denoting the expectation with respect to z uniformly chosen from $\{0,1\}^n$ by Exp_z we get the following equivalent definition of entropy more suitable to our purposes:

$$H(f) = -\mathrm{Exp}_z[\log \mathrm{Prob}_x(f(x) = f(z))].$$

The idea of the protocol is to measure an empirical average value as an approximation to the expected value of this expression. We generate a large polynomial number m of random strings z_i and approximate the expectation by $-\frac{1}{m}\sum_{i=1}^{m} \log \mathrm{Prob}_x(f(z_i) = f(x))$. As the value of this logarithm is between $-n$ and 0, the Hoefding bound (a variation of the Chernoff bound, see [Hof-63]) states that this average will be close to the actual value $H(f)$ except for an exponentially small error probability.

However, we cannot calculate the probability inside the summation, i.e., given z_i it is hard to calculate $\mathrm{Prob}_x(f(z_i) = f(x))$. Therefore, we use the prover to help the verifier approximate this average. Note that since x is sampled according to the uniform distribution, we have

$$\log \mathrm{Prob}_x(f(z_i) = f(x)) = \log |f^{-1}(f(z_i)| - n,$$

thus the problem can be stated in terms of approximating set sizes. We use the set-size lower and upper bound protocols of [GS-89, F-89] for this. The simplest approximation protocols (i.e., the ones that only guarantee a constant factor approximations) are enough for our purposes because we approximate the product $\prod_{i=1}^{m} |f^{-1}(f(z_i))|$ as a whole rather then each of the sets separately. Let us now state the theorem.

Theorem 2 *Let M be a polynomial time probabilistic machine and let x be an input to M of length n. Consider the output of M on x as a distribution $f(r) = M_r(x)$ where r (the random string of M) is chosen uniformly at random and let H be a value that the verifier would like to verify as matching the entropy $H(f)$ of this distribution. Then for any $\epsilon, \delta > 0$ there is a constant round upper bound interactive proof and a constant round lower bound interactive proof for the entropy $H(f)$ such that:*

1. *The verifier runs in polynomial time in n, $1/\delta$, and $\log(1/\epsilon)$, and is assumed to have a black box access to f.*
2. *If the prover plays optimally then the verifier in the upper bound protocol accepts with probability at most ϵ if $H(f) \geq H + \delta$ and rejects with probability at most ϵ if $H(f) \leq H$.*
3. *Similarly, if the prover plays optimally then the verifier in the lower bound protocol accepts with probability at most ϵ if $H(f) \leq H - \delta$ and rejects with probability at most ϵ if $H(f) \geq H$.*

We later refer to the lower bound protocol mentioned in this theorem as an interactive proof for $H(f) \geq H$ with accuracy δ and error ϵ. Before proving the theorem, let us recall the protocols for lower and upper bounds on set sizes.

3.1 Protocols for set sizes

For the sake of self containment, we include the set-size approximation protocols with their proof. For a more detailed description the reader may refer to [F-89, AH-87].

The main tool in these protocols is *universal family of hash functions* (sometimes denoted by universal$_2$ family of hash functions). This is a collection $H_{D,R}$ of functions mapping a domain D to the a range R such that for every point $X \in D$ and a random element $h \in H_{D,R}$, the value $h(X)$ is uniformly distributed in R, and for two elements $X \neq Y \in D$ the values $h(X)$ and $h(Y)$ are independent.

The concept of universal families of hash functions was first defined by [CW-79] and has been used extensively in complexity theory in recent years. In this work, we shall use the fact that for any n and m there is a polynomial time (in n and m) universal family of hash functions from $\{0,1\}^n$ to $\{0,1\}^m$.

Let us begin with the lower-bound. Suppose we have a subset S of a larger domain D, and we assume that the verifier can check if a given element X is in S. We consider a universal family of hash-functions from D to a range R. Basically, in the following protocol the prover convinces the verifier that the cardinality of the set S is bigger than the cardinality of the range R. We first present the protocol, and then we give a lemma that explains the trade-offs between the acceptance probability and the sizes of R and S. The protocol is as follows:

The verifier picks uniformly a random hash-function h from the family and a random element $Y \in R$ and sends them to the prover. The prover responds with an element $X \in D$. The verifier accepts if $X \in S$ and $h(X) = Y$.

The following lemma implies the soundness and completeness of the above protocol. The size of the range R should be set accordingly.

Lemma 3.1 *If the prover plays optimally then the acceptance probability p in the above protocol satisfies*

$$1 - |R|/|S| \leq p \leq |S|/|R|.$$

Proof: The upper bound on the probability p follows from the observation that for any function h at most $|S|$ of the elements in R have inverse images in S. So for this case, we do not really use the properties of the universal hash functions, but we rely only on the random choice of the element $Y \in R$.

For the lower bound on the probability p fix $Y \in R$ and consider the random variable K which represents the number of its inverse images in S with respect to a random hash-function. Note that the prover has no winning strategy only if $K = 0$, it suffices to bound the probability of this event. The expected value of K is $\mathrm{Exp}(K) = |S|/|R|$. Using the pairwise properties of the hash functions family, one may note that the variance is $|S|(1/|R| - 1/|R|^2) < |S|/|R|$. We employ the Chebychev bound to get that

$$\mathrm{Prob}(K = 0) \leq \frac{|S|/|R|}{(|S|/|R|)^2} = \frac{|R|}{|S|}.$$

So here we really did not use the fact that $Y \in R$ is selected at random, but we used the properties of the hash function family. This completes the proof of Lemma 3.1. ■

Let us now describe the set-size upper bound protocol. Again, we assume that there is a non-empty subset S of a domain D. This time, we do not require that the set will be recognizable in polynomial time, but we have to assume that the verifier has one element X in S which was selected uniformly in S, and is unknown to the prover. Again, we are going to use a universal family of hash functions from the domain D to a range R. The protocol is as follows:

The verifier chooses a random hash-function h from the family and sends h and $h(X)$ to the prover. The prover responds with a value $Z \in D$. The verifier accepts if $X = Z$.

Again, the following lemma implies the completeness and soundness of the protocol.

Lemma 3.2 *If the prover plays optimally, then the acceptance probability p of the above interactive proof protocol satisfies* $1 - |S|/|R| \leq p \leq |R|/|S|$.

Proof: The prover can certainly win if $h(X)$ has a unique inverse image in S (which is X). Fix $X \in S$, by the pair-wise independence of the hash functions family, the probability that another fixed element of S is hashed to the same value as X is $1/|R|$. Thus, the probability of such an element existing between the remaining $|S| - 1$ elements in S is at most $(|S| - 1)/|R|$ hence the lower bound on the probability p.

For the upper bound, we assume, without loss of generality, that the prover chooses its optimal strategy deterministically. Fix any hash-function h. The (deterministic) prover can respond with at most $|R|$ different values for Z according to the value $h(X) \in R$ it receives. It can only win if the randomly chosen element X is one of them, thus he wins with probability at most $|R|/|S|$, and we are done with the proof of the Lemma 3.2. ■

Although the set-size approximation protocols just described are sufficient for the approximation of the entropy we are going to need an improved lower bound protocol later for our protocol. Therefore let us state this simple extension here. The amplification we use is similar to the one used by [JVV-86, BP-92]. In order to approximate better the cardinality of the set S, we simply use the above lower bound protocol for the set S^m.

Lemma 3.3 *For every $\delta > 0$ and $\epsilon > 0$ there is two-round interactive proof protocol for lower bounding the size of a set $S \subset \{0,1\}^n$ in which the verifier is given a claimed lower bound s on $|S|$, and a black box for testing membership in S. The verifier runs in polynomial time in n, $1/\delta$ and $\log(1/\epsilon)$ and furthermore:*

- *If $|S| \geq s$ then the prover can make the verifier accept with probability at least $1 - \epsilon$.*
- *If $|S| \leq s(1-\delta)$ no prover can make the verifier accept with probability above ϵ.*

We call such a protocol a proof for $|S| \geq s$ with accuracy δ and error ϵ.

Proof: We apply the protocol of Lemma 3.1 for S^m with a polynomial time universal family of hash functions from $\{0,1\}^{nm}$ to $\{0,1\}^{\lfloor m \log((1-\delta/2)s) \rfloor}$. By Lemma 3.1 we get that the prover can make the acceptance probability at least $1 - (1-\delta/2)^m$ if $|S| \geq s$ but it cannot make the verifier accept with probability more than $(1-\delta/2)^m$ if $|S| \leq (1-\delta)s$. Thus choosing $m = \lceil \frac{2}{\delta} \cdot \log \frac{1}{\epsilon} \rceil$ proves Lemma 3.3. ■

Note that a similar improvement over the upper bound protocol would require the verifier to be given m random elements in S which are not known to the prover. This is not feasible in our case, and seems a hard demand in general. We will not use an improved upper bound protocol in this paper.

3.2 Approximating the entropy

We are ready now to prove Theorem 2. We begin with describing the protocol for approximating the entropy. Let us choose an approximation parameter $\delta > 0$ and an error parameter $\epsilon > 0$. Take any function f defined on $\{0,1\}^n$ and let the value H be the (lower or upper) bound on $H(f)$ that the prover would like to prove.

Let m be a polynomial in n, $1/\delta$, and $\log(1/\epsilon)$ to be specified later. First, we would like to reduce the error by using many copies of the function f. So consider the function F defined on the m-tuples of n-bit strings $D = \{0,1\}^{mn}$ by $F(x_1, \ldots, x_m) = (f(x_1), \ldots, f(x_m))$. For the upper bound we are going to use a universal family of hash functions $H_{D,u}$ from D to $\{0,1\}^u$, where $u = \lfloor m(n - H - \delta/2) \rfloor$ and for the lower bound protocol we shall use a universal family of hash-functions $H_{D,l}$ from D to $\{0,1\}^l$, where $l = \lfloor m(n - H + \delta/2) \rfloor$.

Let us present the upper bound protocol. We present both lower and upper bound protocols, although our main result we use the lower bound protocol only. We assume that the verifier can compute $f(x)$ and thus also $F(X)$. The protocol follows.

- The verifier uniformly picks a random $X \in D$, a hash-function $h \in H_{D,u}$ and an element $Y \in \{0,1\}^u$. The verifier sends $F(X)$, h, and Y to the prover.
- The prover responds with $Z \in D$.
- The verifier accepts iff $F(Z) = F(X)$ and $h(Z) = Y$.

An intuition: This protocol is similar to the regular *lower* bound protocol. Here the prover shows that there is "enough" weight on F operated on a random point. Namely, if Q is the value of operating F on a uniformly chosen element $X \in D$, then Q has "many more" inverses under F, which means that the distribution generated by F is concentrated on "not many" points, and thus the entropy is "small". Note that the average size of $F^{-1}(F(X))$ shrinks as $H(f)$ grows. That is why we use the set-size lower bound protocol for an upper bound on $H(f)$ and vice versa. The formal details follow later.

Let us also present the lower bound protocol. Here we also assume that the verifier can compute F. The lower bound protocol is as follows:

- The verifier uniformly picks a random $X \in D$ and a hash-function $h \in H_{D,l}$. The verifier sends $F(X)$, h, and $h(X)$ to the prover.
- The prover responds with $Z \in D$.
- The verifier accepts iff $X = Z$.

We call this protocol a proof for $H(f) \geq H$ with accuracy δ and error ϵ.

An intuition: Again, this protocol is similar to the *upper* bound protocol of set sizes. The prover shows that a randomly chosen $X \in D$ yields a point $F(X)$ on which a "small" weight is concentrated. Therefore, the weight is partitioned into many points and the entropy is "large". The formal argument is given in the following lemma.

Lemma 3.4 *The following holds for the above protocols:*

1. *The verifier in both protocols can be run in polynomial time in n, $1/\delta$, and $\log(1/\epsilon)$ if it has black-box access to f.*
2. *If the prover plays optimally then the verifier in the upper bound protocol accepts with probability at most ϵ if $H(f) \geq H + \delta$ and rejects with probability at most ϵ if $H(f) \leq H$.*
3. *Similarly, if the prover plays optimally then the verifier in the lower bound protocol accepts with probability at most ϵ if $H(f) \leq H - \delta$ and rejects with probability at most ϵ if $H(f) \geq H$.*

Proof: Clearly, the statement on the efficiency of the verification process holds, since the verifier only has to sample the domain $\{0,1\}^{mn}$, to sample $h \in H_{D,l}$ or $h \in H_{D,u}$, and to compute h and F on given points. So let us concentrate on the error probabilities of the protocols.

The first source of error in both protocols is that for the uniformly chosen $X = (x_1, \ldots, x_m)$ the average $a = 1/m \sum_{i=1}^{m} \log \mathrm{Prob}_y(f(y) = f(x_i))$ might deviate from its expected value, i.e., from $-H(f)$, by more than $\delta/4$. Call such a choice of X bad, and let us bound the probability of choosing a bad X using the Hoefding Inequality [Hof-63]. This inequality asserts that the probability of the average of m identically distributed independent variables deviating from the expected value by at least Δ is at most $2e^{-2\Delta^2 m/r^2}$ where r is the size of the range of the random variables. We can clearly make this less than $\epsilon/2$ by choosing $m > 8n^2 \log(1/\epsilon)/\delta^2$. So this source of error contributes only $\epsilon/2$ to the error probability. Let us continue and check the error probability that we get from the set-size lower or upper bound protocols.

In both protocols, we use the set size approximation protocols on the set $F^{-1}(F(X))$ for the specific X chosen by the verifier. The cardinality of this set is

$$\begin{aligned} |F^{-1}(F(X))| &= \prod_{i=1}^{m} |f^{-1}(f(x_i))| \\ &= \prod_{i=1}^{m} 2^n \mathrm{Prob}_y(f(y) = f(x_i)) \\ &= 2^{m(n+a)} \end{aligned}$$

where a is the empirical average defined above. Thus, if the choice of X is not bad then we get

$$2^{m(n-H(f)-\delta/4)} < |F^{-1}(F(X))| < 2^{m(n-H(f)+\delta/4)}. \tag{2}$$

Suppose X is not bad, and thus cardinality of $F^{-1}(F(X))$ is within the bounds specified in Equation 2. If the upper bound H, claimed by the prover is valid, i.e., $H(f) \leq H$, then by Lemma 3.1 the verifier rejects with probability at most

$$2^u / 2^{m(n-H(f)-\delta/4)} < 2^{-m\delta/4+1}.$$

If however $H(f) \geq H+\delta$ then the probability of acceptance is at most $2^{m(n-H(f)+\delta/4)}/2^u < 2^{-3m\delta/4}$. Both these error probabilities can be made less than $\epsilon/2$ by making $m > 4(\log(1/\epsilon)+2)/\delta$. This proves the claims on the entropy upper bound protocol.

The proof for the lower bound protocol is similar. Notice that conditioned on any value $Y = F(X)$ sent by the verifier to the prover, the actual value of X is a uniformly distributed random element of the set $F^{-1}(Y)$. Thus the set-size *upper* bound protocol and Lemma 3.2 is applicable, and we are done with the proof of Theorem 2. ■

A remark on public coins: The statement of this theorem can be strengthened into an approximation procedure in AM (i.e., the verifier only having public coin tosses) by applying the standard techniques of transforming an interactive proof to an Arthur-Merlin game [GS-89] The lower bound protocol is already an Arthur-Merlin game as it does not hurt if the prover learns X. Obviously, this can not be said about the upper bound protocol.

A remark on the complexity of the function f: Merlin evaluates f only at the end of the game in the modified protocol. This allows us to use the protocol to approximate the entropy not only of polynomial time computable functions but also for functions in for which $\{(x,y)|f(x)=y\} \in \mathcal{AM}$ and $|f(x)|$ is polynomially bounded in $|x|$. To this end, we only have to modify the protocol so that Merlin helps Arthur evaluate the function.

A remark about perfect completeness: Finally, one can reduce the rejection probability when the bound is correct to zero by standard techniques [GMS-87] making a one-sided error Arthur-Merlin game.

4 An overview on the techniques in [AH-87]

The main result of this paper is that $SKC(O(\log n)) \subset \mathcal{AM} \cap co-\mathcal{AM}$. This generalizes the result of Aiello and Hastad [AH-87] stating $SZK \subset \mathcal{AM} \cap co-\mathcal{AM}$. Let us start by recalling the underlying techniques of this latter paper both because we are going to use some of the same techniques and to see why they don't suffice for our purposes.

4.1 The ideas in [AH-87]

The proof in [AH-87] is as follows. First, they present a property of the simulation that holds if and only if $x \in L$. Their proof then contains two parts: First they prove that indeed this property characterizes the case $x \in L$ versus the case $x \notin L$, and second they show how this property can be proven in AM.

The property involved is actually the size of a relative entropy. They consider two distributions: The distribution of conversations output by the simulator, and the distribution of conversations output by the interaction of the original verifier V with the simulation based prover P_M (see Subsection 2.3). They show that if $x \in L$ then the relative entropy $H(M||(P_M,V))$ is zero for perfect zero knowledge as the two distributions actually coincide (and it is "small" for statistical zero knowledge). However $H(M||(P_M,V))$ is "large" if $x \notin L$ since much of the weight in M is

concentrated on the set of accepting transcripts[1] and that set has negligible weight in (P_M, V). Recall that by the definition of relative entropy:

$$H(M||(P_M,V)) = \sum_c \mathrm{Prob_M}[c] \cdot \log \frac{\mathrm{Prob_M}[c]}{\mathrm{Prob_{(P_M,V)}}[c]}.$$

(We use the notations of Subsection 2.4.)

It is shown in [AH-87] how to prove that this related entropy is big or small in AM. Using approximation of the entropy described in Section 3, we can offer a more compact presentation of that protocol.

Recall Equation 1 from Subsection 2.4. For any valid transcript c it holds that

$$\mathrm{Prob_{(P_M,V)}}[c] = 2^{-t(n)} \cdot \prod_{i=1}^{\frac{d(n)-1}{2}} \frac{\mathrm{Prob_M}[c_{2i}]}{\mathrm{Prob_M}[c_{2i-1}]},$$

where $t(n)$ is the number of random bits used by the verifier V and $d(n)$ is the number of rounds in the protocol.

Assuming that the simulation only gives valid transcripts we may rewrite the relative entropy

$$\begin{aligned} H(M||(P_M,V)) &= \sum_c \mathrm{Prob_M}[c] \log \frac{\mathrm{Prob_M}[c]}{\mathrm{Prob_{(P_M,V)}}[c]} \\ &= \sum_c \mathrm{Prob_M}[c](\log \mathrm{Prob_M}[c] + t(n) - \sum_{i=1}^{d-1} (-1)^i \log \mathrm{Prob_M}[c_i]) \\ &= t(n) + \sum_{i=1}^{d} (-1)^i H(c_i). \end{aligned}$$

Here $H(c_i)$ is the entropy of the first i messages generated by M. So it remains to notice that these $d(n)$ entropies can be approximated in parallel in AM, which follows from Theorem 2.

We remark that the entropy approximations we presented are most accurate and therefore, we do not need to reduce the error probability of the original protocol in order to apply this technique. The error probability had to be reduced in [AH-87] since they used approximations on set sizes which were much less accurate.

4.2 Generalizing these techniques

Let us consider what happens with the relative entropy $H(M||(P_M,V))$ if the knowledge complexity is not zero but logarithmic. It is still big in the case $x \notin L$ for similar reasons. However if $x \in L$, even for the case that $k(n) = 1$, only half of the distribution generated by the simulator has to be identical to the one generated by P and V and the rest is arbitrary. This "bad half" of the distribution M can be concentrated on a single transcript c for which $\mathrm{Prob_M}[c] > 1/2$ but $\mathrm{Prob_{(P_M,V)}}[c] = 2^{-n}$ thus making $H(M||(P_M,V))$ big although $x \in L$. Therefore, this relative entropy is not able to distinguish between $x \in L$ and $x \notin L$.

Note that in our example there is one (or a few) bad conversations that make the relative entropy become large. We can express the relative entropy as an expectation expression:

$$H(M||(P_M,V)) = \mathrm{Exp}_{c \in M}[\log(\mathrm{Prob_M}[c]/\mathrm{Prob_{(P_M,V)}}[c])].$$

[1] The concentration of many accepting conversations in the output of the simulator is not a must, but if this is not the case, then the verifier can verify that $x \notin L$ without interaction with the prover simply by invoking the efficient simulator.

We are going to claim that even though approximating the expected value is not helpful, approximating the tail of the involved distribution will do the work.

In case $x \in L$ the good part of the distribution M (the part that really simulates (P,V)) consists of mostly accepting transcripts c, and for most of them $\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]$ is limited. This is easy to see for the real interaction (P,V), i.e., for the fraction $\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P,V})}[c]$, but it requires an involved calculation for the interaction (P_M,V) (see next section).

If $x \notin L$ however, (P_M,V) is mostly rejecting and thus if M outputs many accepting transcripts then $\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]$ is very big for most of them. See the easy argument in the next section.

These observations lead us, in order to separate between the case of $x \in L$ and the case of $x \notin L$, to consider the probability that a conversation c output by M is accepting and has a small ratio $\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]$. This probability will be substantially bigger in the case $x \in L$ than in the case $x \notin L$. see Section 5 below.

5 The difference between $x \in L$ and $x \notin L$

In this section we introduce a measure and prove that it separates $x \in L$ from $x \notin L$. In the following sections we explain how to use this measure in the case of logarithmic knowledge complexity in order to show that $x \in L$ (or $x \notin L$) in a constant round interactive proof. Thus, we get that $L \in \mathcal{AM} \cap \mathrm{co}\text{-}\mathcal{AM}$. We will also use this measure to show that a language with an interactive proof in which the error probability is smaller than $2^{-2k(n)-6}$ (where $k(n)$ is the knowledge complexity of the interactive proof) is in the third level of the polynomial time hierarchy. Let us state the separation lemma.

Lemma 5.1 *Let (P,V) be an interactive proof for a language L. Let $\epsilon(n) < 1/4$ be the error probability and $k(n)$ be the knowledge complexity of this proof. Let M be the corresponding simulator and P_M the simulation based prover.*

1. *If $x \notin L$ then for any $b > 0$ we have:*

$$\mathrm{Prob}_{\mathrm{M}}\left[c \text{ is valid accepting and } \frac{\mathrm{Prob}_{\mathrm{M}}[c]}{\mathrm{Prob}_{(\mathrm{P_M,V})}[c]} \leq b\right] \leq \epsilon \cdot b$$

2. *Whereas if $x \in L$ then*

$$\mathrm{Prob}_{\mathrm{M}}\left[c \text{ is valid accepting and } \frac{\mathrm{Prob}_{\mathrm{M}}[c]}{\mathrm{Prob}_{(\mathrm{P_M,V})}[c]} \leq 2^{k(n)+2}\right] \geq 2^{-k(n)-2}$$

Proof: Fix x of length n and let $k = k(n)$ and $\epsilon = \epsilon(n)$. We begin by proving Part (1) of the lemma. Let A be the set of valid accepting conversation for which the ratio is small. Namely, for all $c \in A$, we have

$$\mathrm{Prob}_{\mathrm{M}}[c] \leq \mathrm{Prob}_{(\mathrm{P_M,V})}[c] \cdot b.$$

We have to show that $\mathrm{Prob}_{\mathrm{M}}[A]$ is small.

First, by the definition of A, we know that

$$\mathrm{Prob}_{\mathrm{M}}[A] \leq \mathrm{Prob}_{(\mathrm{P_M,V})}[A] \cdot b \tag{3}$$

(simply sum over all conversations in A). We know that since the conversations in A are accepting and since, by the soundness property of the interactive proof, no prover is able to convince the verifier to accept with probability greater than ϵ, then we have

$$\mathrm{Prob}_{(\mathrm{P_M,V})}(A) \leq \epsilon. \tag{4}$$

Combining Equation 3 and Equation 4 we get that $\mathrm{Prob}_{\mathrm{M}}[A] \leq \epsilon \cdot b$ as needed for Part 1 of the lemma.

For Part 2 of the lemma we need a general tool connecting the distribution generated by the original prover P and the verifier V to the distribution generated by P_M and V. Lemma 5.2 establishes this connection. This lemma is implicit in [GOP-94].

Lemma 5.2 [GOP-94]: *Let L, P, V, and M be as above, and fix any $x \in L$. Then, for any set A of conversations it holds that:*

$$\mathrm{Prob}_{(\mathrm{P_M,V})}[A] \geq (\mathrm{Prob}_{(\mathrm{P,V})}[A])^2 \cdot 2^{-k(n)}.$$

For self containment, we provide the proof in Appendix A. Let us now use it to finish the proof of Part 2 of Lemma 5.1. Consider the set A' for which the ratio in the lemma is big. Namely, let A' consist of the transcripts c for which

$$\frac{\mathrm{Prob}_{\mathrm{M}}[c]}{\mathrm{Prob}_{(\mathrm{P_M,V})}[c]} \geq 2^{k+2}$$

By the definition of the set A' (i.e., we sum over all conversations in A'), we get

$$\mathrm{Prob}_{\mathrm{M}}[A'] \geq \mathrm{Prob}_{(\mathrm{P_M,V})}[A'] \cdot 2^{k+2} \tag{5}$$

Using Lemma 5.2 we get that

$$\mathrm{Prob}_{(\mathrm{P_M,V})}[A'] \geq (\mathrm{Prob}_{(\mathrm{P,V})}[A'])^2 \cdot 2^{-k} \tag{6}$$

Combining Equation 5 and Equation 6 we get

$$\mathrm{Prob}_{(\mathrm{P,V})}[A'] \leq \sqrt{\mathrm{Prob}_{\mathrm{M}}[A']}/2 \leq 1/2. \tag{7}$$

Let A be the set of conversations that is mentioned in the lemma. Namely, the set of valid and accepting transcripts for which

$$\frac{\mathrm{Prob}_{\mathrm{M}}[c]}{\mathrm{Prob}_{(\mathrm{P_M,V})}[c]} \leq 2^{k(n)+2}.$$

Note that A is not the complement of A' since in A we require that the conversations will be valid and accepting. In the original interaction (P,V), all conversations are valid and only ϵ are not accepting. Therefore,

$$\mathrm{Prob}_{(\mathrm{P,V})}[A] \geq 1 - \mathrm{Prob}_{(\mathrm{P,V})}[A'] - \epsilon \geq \frac{1}{2} - \epsilon \geq \frac{1}{4}.$$

We conclude by recalling that there is a subspace of density at least 2^{-k} in the simulation is identical to the interaction between P and V and thus

$$\mathrm{Prob}_{\mathrm{M}}[A] \geq 2^{-k} \cdot \mathrm{Prob}_{(\mathrm{P,V})}[A] \geq 2^{-k-2}$$

and we are done with the proof of Part 2 of Lemma 5.1. ■

6 The main theorem

We now use the above machinery to introduce a constant round interactive proof for the language L or its complement. Using [GS-89, BM-88], we get that L is in $\mathcal{AM} \cap \text{co}-\mathcal{AM}$. Formally, we prove the following theorem.

Theorem 3

$$\mathcal{SKC}(O(\log n)) \subseteq \mathcal{AM} \cap \text{co}-\mathcal{AM}$$

We will only show that

$$\mathcal{PKC}(O(\log n)) \subseteq \mathcal{AM} \cap \text{co}-\mathcal{AM}$$

since it was shown in [GOP-94] (see Theorem 1) that

$$\mathcal{SKC}(O(\log n)) = \mathcal{PKC}(O(\log n)).$$

We remark that the theorem in [GOP-94] only applies for the honest verifier simulation, but it suffices for us since we are only using the simulation of the honest verifier.

So let us begin by recalling the setting. We have a language L which is in $\mathcal{PKC}(k(n))$ for some $k(n) = O(\log n)$. Namely, there is an interactive proof (P, V) for L, and there is a simulator M which runs efficiently and outputs a distribution on conversations between P and V. We also consider the distribution of conversations generated by an interaction between the simulation based prover P_M and the original verifier V (see Section 2).

Let A be the set of conversations c which are valid and accepting. We use Lemma 5.1 with $b = 2^{k(n)+3}$ to obtain the following separation:

If $x \notin L$ then

$$\text{Prob}_{\text{M}}\left[c \in A \quad \wedge \quad \log \frac{\text{Prob}_{\text{M}}[c]}{\text{Prob}_{(\text{P}_{\text{M}},\text{V})}[c]} \leq k(n) + 3 \right] \leq \epsilon \cdot 2^{k(n)+3} \tag{8}$$

Whereas if $x \in L$ then

$$\text{Prob}_{\text{M}}\left[c \in A \quad \wedge \quad \log \frac{\text{Prob}_{\text{M}}[c]}{\text{Prob}_{(\text{P}_{\text{M}},\text{V})}[c]} \leq k(n) + 2 \right] \geq 2^{-k(n)-2} \tag{9}$$

Notice that the probability is a polynomial fraction in one case and it is negligible in the other.

In our protocol the new verifier V', with the help of the new prover P', approximates the probability that a conversation c, output by the simulator, satisfies the three following properties: The conversation c is valid, it is accepting, and $\log \frac{\text{Prob}_{\text{M}}[c]}{\text{Prob}_{(\text{P}_{\text{M}},\text{V})}[c]} \leq k(n) + 2$. The verifier V' does that by running the simulator M a large (yet polynomial) number of times, and checking what is the fraction of the conversations that satisfy these conditions. The probability that the simulator outputs such a conversation is then very well approximated by the fraction of the actual output conversations that satisfy these properties.

It is easy to check if a conversation is valid and accepting but in order to approximate $\text{Prob}_{\text{M}}[c]$ and $\text{Prob}_{(\text{P}_{\text{M}},\text{V})}[c]$, the verifier needs the prover's help. The approximations of these probabilities will translate into approximations of set sizes. Actually, approximating $\text{Prob}_{\text{M}}[c]$ will require one set approximation, and approximating $\text{Prob}_{(\text{P}_{\text{M}},\text{V})}[c]$ will require approximations of $d(n)$ sets (where $d(n)$ is the number of rounds in the interaction). Since we only know how to approximate set sizes (and not how to compute them exactly) in a constant round interaction of P' and V', we really need the difference in the thresholds in the separation property of Inequalities 8 and 9.

We are going to approximate the sizes of the sets involved in the following way. The prover will state the size of the set, and then he will prove corresponding lower and upper bounds. As explained in Section 3, lower bounds (and even accurate ones) are easy to get. The verifier only has to be able to recognize elements in the sets involved and this will turn out easy. However, there is a problem with the upper bounds. in order to get upper bounds, we must let the verifier have a "hidden" random element in each of the sets that have to be bounded. As it turns out, the random seed of M producing the conversation c is such an element for any of the $d(n)$ sets that are involved in this computation. Unfortunately, this hidden random element can only be used once. After that, the seed is not hidden anymore, and cannot be used for all the other sets.

To solve this, we begin by "believing" the prover instead of checking the upper bounds. Namely, we check all lower bounds on the stated sizes and we do not check any upper bound. We use the given values in the protocol as if they were verified. After that, we check that "most" of them were "almost" correct in the following manner. We use all the given set sizes to compute a related entropy. This is a second use of these values, but now we don't have to trust the outcome. We can actually check it since we know how to approximate the entropy using Theorem 2. Since the cheating prover cannot cheat in the lower bounds, then all his cheatings have to be biased into stating smaller set sizes than the sizes actually are. This bias would lead to a wrong entropy calculation and later to rejection.

In order to present the protocol, let us first explain how the probabilities $\mathrm{Prob}_{\mathrm{M}}[c]$ and $\mathrm{Prob}_{(\mathrm{P_M,V})}[c]$ are stated in terms of set sizes. As in the proof of Lemma 5.2 we define Ω to be the set of the possible random tapes of the simulator M and for a prefix of a conversation h let Ω_h be all the random tapes with which M outputs a conversation starting with h. Clearly, $\mathrm{Prob}_{\mathrm{M}}[c] = |\Omega_c|/|\Omega|$. Using Equation 1 from Subsection 2.4 one gets that for valid transcripts c

$$\mathrm{Prob}_{(\mathrm{P_M,V})}[c] = 2^{-t(n)} \cdot \prod_{i=1}^{\frac{d(n)-1}{2}} \frac{|\Omega_{c_{2i}}|}{|\Omega_{c_{2i-1}}|},$$

where $t(n)$ is the length of the random tape of V, $d(n)$ is the number of messages exchanged by P and V and c_i denotes the i-message prefix of c. Denoting the length of the random tape of M by $t'(n)$ we have $|\Omega| = 2^{t'(n)}$ and for valid transcripts c

$$\log \frac{\mathrm{Prob}_{\mathrm{M}}[c]}{\mathrm{Prob}_{(\mathrm{P_M,V})}[c]} = t(n) - t'(n) - \sum_{i=1}^{d(n)} (-1)^i \log |\Omega_{c_i}|. \tag{10}$$

It remains to approximate the sizes of the sets Ω_{c_i} for all $i = 1, 2, \ldots, d(n)$.

Let us set the following parameters. The probability of error is set to $\epsilon_0 = 2^{-n}$, the quality of approximations is set to $\delta = 2^{-k(n)-9}/(d(n))^2$, and the number of simulator conversations that we check is $\ell = \lceil n(t'(n))^2/\delta^2 \rceil$. Notice that as a function of n ϵ_0 is negligible, δ is a polynomial fraction and ℓ is a polynomial.

The protocol for recognizing L on input x

The verifier V' picks ℓ random conversations $c^1, \ldots, c^\ell$ from the distribution generated by M and sends them to P'.

The prover P' states the numbers ω_i^j (claimed to be the sizes of $\Omega_{c_i^j}$) for all $i = 1, 2, \ldots, d(n)$ and $j = 1, \ldots, \ell$. Then, he proves to the verifier V' that ω_i^j is a lower bound on the size of the set $\Omega_{c_i^j}$ for all $i = 1, 2, \ldots, d(n)$ and $j = 1, \ldots, \ell$. All the lower bounds are done in parallel, according to

the protocol of Lemma 3.3[2], and with accuracy δ and error probability ϵ_0. If the prover fails to prove any of the bounds, then the verifier rejects and halts.

The verifier V' computes, for each of the conversations c^j $(j = 1, 2, \ldots, l)$ an approximation of $\log(\mathrm{Prob}_{\mathrm{M}}[c^j]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c^j])$ by computing $v_j = t(n) - t'(n) - \sum_{i=1}^{d(n)} (-1)^i \log \omega_i^j$. Then the verifier counts the number of conversations which are valid and accepting and for which $v_j \leq k(n) + 2.5$. It rejects if this number is below $\ell \cdot 2^{-k(n)-3}$. Next, the verifier uses the values ω_i^j stated by the prover to compute for each round i $(1 \leq i \leq d(n))$ the empirical entropy $h_i = 1/\ell(n) \sum_{j=1}^{l} (t'(n) - \log \omega_i^j)$.

Finally, the prover P' proves that $h_i - \delta$ is a lower bound on the entropy $H(c_i)$ for each round $i = 1, 2, \ldots, d(n)$. The random variable c_i represents the output of the simulator M truncated to the first i rounds. He proves these lower bounds using the protocol of Theorem 2 $d(n)$ times in parallel to prove that $h_i - \delta \leq H(c_i)$ with accuracy δ and error ϵ_0. If any of these protocols ends in rejection then the verifier rejects. Otherwise, it accepts.

The protocol for $\overline{L}$:

The protocol for $\overline{L}$ is actually the same protocol except that we reverse the rule for rejection in the third part of the protocol for L. The modified verifier rejects if the number of indices j for which c_j is valid and accepting and $v_j \leq k(n) + 2.5$ is greater than $\ell \cdot 2^{-k(n)-3}$.

In order to prove Theorem 3 it is enough to prove the following lemma.

Lemma 6.1 *The above protocol is a constant round interactive proof for L while the modified protocol is a constant round interactive proof for the complement of L.*

Proof: Clearly the protocol has a constant number of rounds since the protocols for bounds on set sizes and the entropy value can be performed in a constant number of rounds. Let us go on and prove the soundness and completeness properties of this interactive proof. Since $k(n)$ is $O(\log n)$ and $\epsilon(n)$ is negligible, we may assume in what follows that $\epsilon(n) < 2^{-2k(n)-7}$. This is true for large enough n.

A common source of error for both protocols and for both soundness and completeness, comes from the possibility that the number of "good" conversations output by the simulator is far from its expected value. Namely, for $x \in L$, the frequency of the conversations c that are valid, accepting, and have $\log(\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]) \leq k(n) + 3$ amongst the ℓ random conversations output by M, is substantially different from the actual probability of such a conversation being output by M. The Chernoff bound limits the probability of the difference being at least δ to $2e^{-2\delta^2 \ell}$. Notice that this error probability is negligible. The same argument applies for $x \notin L$ and the difference between the actual and empirical probability of valid accepting conversations with $\log(\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]) \leq k(n) + 2$.

A similar source of error is the possibility that for some $i = 1, \ldots, d(n)$ the empirical entropy $H_i = 1/\ell \sum_{j=1}^{\ell} (t'(n) - \log |\Omega_{c_i^j}|)$ is far from the real entropy $H(c_i)$. As in Section 3 we use Hoefding inequality to bound the probability of this difference exceeding δ by $2e^{-2\delta^2 \ell/(t'(n))^2}$. This error probability is also negligible.

We call a choice of the random conversations c^j $(j = 1, 2, \ldots, \ell(n))$ bad if any of the above discrepancies occur. The probability of the verifier getting a bad set of conversations when invoking the simulator in the first step is negligible.

[2]Good lower bound protocols in AM are well known. For self containment, we include one in Section 3.

We begin with the completeness property of the protocol for L. So suppose we apply this protocol on an input $x \in L$. If the choice of the conversations is not bad and the prover gives the correct values $\omega_i^j = |\Omega_{c_i^j}|$ then by Inequality 9 rejection can come only from errors in the set size lower bound protocols or the entropy lower bound protocols. Since we run only $(\ell+1)d(n)$ such protocols and since the probability to make an error in any of them is at most $\epsilon_o = 2^{-n}$, then the probability of such error is at most $(\ell+1)d(n)\epsilon_0$ which is negligible. The proof of completeness of the protocol for $\overline{L}$ is similar using Inequality 8.

We now turn to proving the soundness. Consider again the protocol for L but this time on an input $x \notin L$. Suppose that the set of ℓ conversations output by the simulator is not bad. The prover has three possible strategies when stating the values ω_i^j.

Possibility 1: The values ω_i^j stated by the prover contain one value which is a little higher than it should be: The first cheating strategy is when the prover states the values ω_i^j $(1 \le i \le d(n),\ 1 \le j \le \ell(n))$ such that one of them satisfies $\omega_i^j > (1+\delta)|\Omega_{c_i^j}|$ In this case, he passes the lower bound protocol with the verifier not rejecting with probability at most ϵ_0. So assume that for all the values ω_i^j stated by the prover it holds that $\omega_i^j < (1+\delta)|\Omega_{c_i^j}|$, i.e., the stated values are never too high.

Possibility 2: The values ω_i^j stated by the prover contain a fraction $15\delta d(n)$ being somewhat lower than they should be. A second possibility is that the prover states the numbers ω_i^j such that out of the $\ell(n)\cdot d(n)$ numbers ω_i^j, there are $15\delta d(n)^2\ell(n)$ which are smaller by a factor of $2^{-1/(3d(n))}$ than the size of Ω_i^j. In this case, there must be a round i $(1 \le i \le d(n))$ for which $\omega_i^j < 2^{-1/(3d(n))}|\Omega_{c_i^j}|$ for at least $15\delta d(n)\ell(n)$ numbers out of the $\ell(n)$ possible indices j. Since the first possibility does not hold, we also know that $\omega_i^j < (1+\delta)|\Omega_{c_i^j}|$ for all the values ω_i^j. In this case, the verifier approximation h_i is far from the real empirical entropy H_i:

$$\begin{aligned} h_i &= \frac{1}{\ell(n)}\sum_{j=1}^{\ell(n)}(t'(n) - \log \omega_i^j) \\ &> \frac{1}{\ell(n)}\sum_{j=1}^{\ell(n)}(t'(n) - \log |\Omega_{c_i^j}|) - log(1+\delta) + \frac{15\delta d(n)}{3d(n)} \\ &= H_i - \log(1+\delta) + 15\delta/3. \end{aligned}$$

However, since we have ruled out bad sampling of the simulator, the empirical entropy H_i is close to the real entropy $H(c_i)$, i.e., $H_i \ge H(c_i) - \delta$. Thus:

$$\begin{aligned} h_i &\ge H_i + 3\delta \\ &\ge H(c_i) + 2\delta. \end{aligned}$$

So when the prover tries to show that $h_i - \delta \le H(c_i)$ (using the entropy lower bound protocol) he will succeed with probability at most ϵ_0.

Possibility 3: Neither of the above happen. In this case we are going to show that the number of conversations for which the verifier computes $v_j \le k(n) + 2.5$ is less then $\ell(n)\cdot 2^{-k(n)-3}$ and thus the verifier rejects. If neither of the above two possibilities happen then for all indices except for at most $15\delta(d(n))^2\ell(n)$ pairs (i,j) we have

$$2^{-1/(3(d(n))} \cdot |\Omega_{c_i^j}| \ \le\ \omega_i^j \ \le\ (1+\delta)|\Omega_{c_i^j}|. \tag{11}$$

Furthermore, the number of conversations c^j for which Inequality 11 holds for all rounds i is at least $\ell(n) - 15\delta(d(n))^2\ell(n)$. For such a conversation c^j, the verifier's approximation of $\log(\mathrm{Prob}_{\mathrm{M}}[c_j]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c_j])$ is correct to within $1/3$. Namely,

$$\begin{aligned} v_j &= t(n) - t'(n) - \sum_{i=1}^{d(n)} (-1)^i \log \omega_i^j \\ &\geq t(n) - t'(n) - \sum_{i=1}^{d(n)} (-1)^i \log |\Omega_i^j| \ - \ 1/3 \\ &= \log(\frac{\mathrm{Prob}_{\mathrm{M}}[c_j]}{\mathrm{Prob}_{(\mathrm{P_M,V})}[c_j]}) \ - \ 1/3 \end{aligned}$$

We call these conversations "well approximated". Therefore, if a conversation is well approximated, and $v_j \leq k(n) + 2.5$, then we also get that for this conversation $\log(\mathrm{Prob}_{\mathrm{M}}[c_j]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c_j]) \leq 3$. By Inequality 8, we know that the probability that a conversation output by the simulator is valid accepting and having $\log(\mathrm{Prob}_{\mathrm{M}}[c_j]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c_j]) < k(n) + 3$ is at most $\epsilon \cdot 2^{k(n)+3}$. Also, since the set of conversations is not bad, then the actual fraction of conversations for which $\log(\mathrm{Prob}_{\mathrm{M}}[c_j]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c_j]) < k(n) + 3$ is at most $\epsilon \cdot 2^{k(n)+3} + \delta$.

Thus the number of "good conversations" counted by the verifier is limited to $(2^{k(n)+3}\epsilon + \delta)\ell(n) + 15\delta(d(n))^2\ell(n)$. By the setting of δ and since ϵ is negligible, we get that this is at most $2^{-k(n)-3}\ell$ and the verifier rejects.

Thus the overall acceptance probability is negligible, and we proved the soundness of the protocol.

For the soundness of the protocol for the complement of L we take $x \in L$ and suppose the verifier does not choose a bed set of conversations. We consider the same three possibilities for the values ω_i^j as above. In the first two cases the acceptance probability is at most ϵ_0 for the same reasons. In the third case we use Inequality 9 to show that the verifier sees more than $2^{-k(n)-3}\ell$ valid accepting conversations with $v_j \leq k(n) + 2.5$ and thus the verifier rejects. ■

A remark about the precision of calculations: During the protocol, the verifier is required to compute $v_j = t(n) - t'(n) - \sum_{i=1}^{d(n)} (-1)^i \log \omega_i^j$ and $h_i = 1/\ell(n) \sum_{j=1}^{l} (t'(n) - \log \omega_i^j)$, which involves calculations of real numbers. One solution is to let him compute 2^{v_j} and $2^{\ell h_i}$ which only involves multiplications of integer fractions. Another solution is to use rounding such that the result is accurate to within $\delta/2$ and make the protocol itself be accurate to within a $\delta/2$ approximation error. Thus the overall approximation error is below δ.

7 The connection between knowledge and error

In this section we state that if a language L has an interactive proof whose error probability is small compared to its knowledge complexity then L has limited computational complexity. Our result is as follows:

Theorem 4 *If there is a interactive proof for a language L with statistical knowledge complexity $k(n)$ and error probability $\epsilon(n) \leq 2^{-3k(n)}$ and if $k(n)$ is computable in polynomial time, then $L \in \mathcal{AM}^{\mathcal{NP}}$.*

Remarks: The term $\mathcal{AM}^{\mathcal{NP}}$ refers to an AM protocol in which the verifier has access to an $\mathcal{NP}$-complete oracle (the computational unbounded prover doesn't need one). Using standard

techniques, it can be shown that $\mathcal{AM}^{\mathcal{NP}} \subseteq \Pi_3^P$, and therefore all languages having this type of interactive proof must be in the third level of the polynomial time hierarchy. (The $\mathcal{AM} \subseteq \Pi_2^P$ result is stated in [B-85] and the proof generalizes to any oracle.) Note also that $k(n)$ has to be computable in polynomial time in n and not in the length of the string $k(n)$, which is usually much smaller. So the restriction is quite liberal.

Proof: We use a result of [GOP-94] (see Theorem 1) again to shift attention from statistical to perfect knowledge complexity. We prove that if a language L has an interactive proof with *perfect* knowledge complexity $k(n)$ and error below $2^{-2k(n)-6}$ then $L \in \mathcal{AM}^{\mathcal{NP}}$. This implies our theorem. (Note that by definition of interactive proofs $\epsilon(n)$ is a negligible fraction. This property of $\epsilon(n)$ is needed only for the transformation from perfect to statistical knowledge complexity. The statement for perfect knowledge compliexity holds also for non-negligible $\epsilon(n)$.)

The proof is based on the observation that Inequalities 8 and 9 of Section 6 still separate the elements of L from the non-elements. Let us call a conversation c *good* if it is valid, accepting, and $\log(\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]) < k(n) + 2.5$. When $x \in L$ the probability of M outputting a good conversation is much bigger than when $x \notin L$. But if $k(n)$ is super-logarithmic then both of these probabilities may be negligible. Thus, the procedure of sampling the simulator for a polynomial number of times and checking if these three conditions hold is not useful any more. Instead, we let the prover prove that there are "many" random seeds making the simulator M output good conversations. This has the flavor of a set-size lower bound protocol.

In the set size lower bound described in Subsection 3.1 it is required that the verifier is able to recognize elements in the set. In our case, checking if c is valid and accepting is simple, but we do not know how to approximate $\log(\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c])$ in polynomial time. By Equation 10 in Section 6, this approximation comes down to approximating set-sizes. Note that all these sets which need to be approximated are recognizable in polynomial time. It is shown in [Si-83, St-83, BP-92] how to approximate the cardinality of a set S, which is recognizable in polynomial time, using efficient computation with access to an $\mathcal{NP}$ oracle. The approximation there fails with negligible probability to give an approximation which is within $1 + \frac{1}{\mathrm{poly}}$ from the exact cardinality.

We apply the protocol of Lemma 3.3 to prove $|S| > 2^{-k(n)-2}$ with accuracy $1/2$ and negligible error, where S is the set of random tapes that cause M to produce good conversations. Instead of the black-box access to membership in S we have a randomized process of approximating $\log(\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c])$ with Equation 10. We can set the accuracy of each set-size approximation to within $1/(3d(n))$ and the error of these approximations negligible again. This does not give exact membership test in S but except for negligible error it accepts if the random tape produces a valid and accepting conversation c with $\log(\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]) < k(n) + 2$ while it rejects except for a negligible probability if the output is not valid or not accepting or if $\log(\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]) > k(n) + 3$. Using Inequalities 8 and 9 this is enough for our purposes. ■

Clearly, our result implies that $\overline{L} \in \Sigma_3^P$. However, we do not know how to do better in the sense of producing a protocol in $\mathcal{AM}^{\mathcal{NP}}$ for the language $\overline{L}$. This asymmetry follows from the more restricting demand in the upper bound protocol. There, the verifier is required to be able to produce a random element in the set involved hidden from the prover.

8 Open questions

Many questions regarding the relation between knowledge complexity and computational complexity are still open. Can one show a better lower bound on the knowledge complexity of $\mathcal{NP}$-complete

languages or even of PSPACE-complete languages? Any such bound implies PSPACE$\neq \mathcal{BPP}$ so one would only expect such results with complexity assumptions like the polynomial time hierarchy not collapsing. But no such lower bound on the knowledge complexity which is higher than the present super-logarithmic bound is known.

Also, there are open questions with regards to the hierarchy of languages classified by their knowledge complexity. Is there a constant factor collapse? Namely, is $KC(2k(n)) = KC(k(n))$? Actually, in view of the results presented in this paper, there is no difference between the limitations known today for zero knowledge languages and languages with logarithmic knowledge complexity. Could one show that these classes collide? Namely, $KC(O(\log n)) = KC(0)$? Can one even show that $KC(1) = KC(0)$? Can one give indications that this is not the case?

9 Acknowledgment

We would like to thank Rafi Ostrovsky for helpful discussions.

References

[ABV-95] W. Aiello, M. Bellare and R. Venkatesan. Knowledge on the Average – Perfect, Statistical and Logarithmic. *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing,* ACM (1995).

[AH-87] W. Aiello and J. Håstad. Perfect Zero-Knowledge can be Recognized in Two Rounds. *Proceedings of the 28th Annual IEEE Symposium on the Foundations of Computer Science,* IEEE (1987).

[B-85] L. Babai. Trading Group Theory for Randomness. *Proceedings of the 17th Annual ACM Symposium on the Theory of Computing,* ACM (1985).

[BM-88] L. Babai and S. Moran. Arthur-Merlin Games: A Randomized Proof System and a Hierarchy of Complexity Classes. *JCSS*, Vol. 36, pages 254–276, 1988.

[BCK] Bar-Yehuda, R., B. Chor, and E. Kushilevitz, "Privacy, Additional Information, and Communication", *5th IEEE Structure in Complexity Theory*, July 1990, pp. 55-65.

[BMO-90] M. Bellare, S. Micali and R. Ostrovsky. The (True) Complexity of Statistical Zero-Knowledge. *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing,* ACM (1990).

[BP-92] M. Bellare and E. Petrank. Making Zero-Knowledge Provers Efficient. *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing,* ACM (1992)

[B+ 88] M. Ben-Or, S. Goldwasser, O. Goldreich, J. Håstad, J. Kilian, S. Micali and P. Rogaway. Everything Provable is Provable in Zero-Knowledge. *Advances in Cryptology — Proceedings of CRYPTO 88,* Lecture Notes in Computer Science 403, Springer-Verlag (1989). S. Goldwasser, ed.

[BHZ-87] R. Boppana, J. Håstad and S. Zachos. Does *co-NP* Have Short Interactive Proofs". *Information Processing Letters*, Vol 25 (1987), No. 2, pp 127–132.

[CW-79] L. Carter and M. Wegman. Universal Classes of Hash Functions. *J. Computer and System Sciences* **18**, 143–154 (1979).

[F-89] L. Fortnow. The Complexity of Perfect Zero-Knowledge. *Advances in Computing Research (ed. S. Micali)* Vol. 18 (1989).

[GMS-87] O. Goldreich, Y. Mansour and M. Sipser. Interactive Proof Systems: Provers that never Fail and Random Selection. *Proceedings of the 28th Annual IEEE Symposium on the Foundations of Computer Science,* IEEE (1987).

[GMW-86] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design", *Proc. 27th FOCS 86,* See also *Jour. of ACM.* Vol 38, No 1, July 1991, pp. 691–729.

[GMW-87] O. Goldreich, S. Micali, and A. Wigderson, "How to Play any Mental Game or a Completeness Theorems for Protocols of Honest Majority", STOC87.

[GP-91] O. Goldreich and E. Petrank. Quantifying Knowledge Complexity. *Proceedings of the 32nd Annual IEEE Symposium on the Foundations of Computer Science,* IEEE (1991). Submitted for publication, 1995.

[GMR-85] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proofs. *Proceedings of the 17th Annual ACM Symposium on the Theory of Computing,* ACM (1985).

[GMR-89] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proofs. *SIAM J. Comput.* **18** (1), 186-208 (February 1989).

[GOP-94] O. Goldreich,, R. Ostrovsky, and E. Petrank. Computational Complexity and Knowl edge Complexity. *26th ACM Symp. on Theory of Computation*, May 1994. pp. 534-543.

[GS-89] S. Goldwasser, and M. Sipser, Private Coins vs. Public Coins in Interactive Proof Systems, *Advances in Computing Research (ed. S. Micali),* 1989, Vol. 5, pp. 73-90.

[H-94] J. Håstad. Perfect Zero-Knowledge in $\mathcal{AM} \cap$co-$\mathcal{AM}$. Unpublished 2-page manuscript explaining the underlying ideas behind [AH-87]. 1994.

[Hof-63] W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables, *Amer. Stat. Assoc. Jour.*, March 1963, pp 13–30.

[ILu-90] R. Impagliazzo and M. Luby, One-Way Functions are Essential for Complexity Based Cryptography, *30th FOCS*, pp. 230–235, 1990.

[ILe-90] R. Impagliazzo and L.A. Levin, No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random, *31st FOCS*, pp. 812-821, 1990.

[IY-87] R. Impagliazzo and M. Yung. Direct Minimum-Knowledge computations. *Advances in Cryptology — Proceedings of CRYPTO 87,* Lecture Notes in Computer Science 293, Springer-Verlag (1987).

[JVV-86] M. Jerrum, L. Valiant and V. Vazirani. Random Generation of Combinatorial Structures from a Uniform Distribution. *Theoretical Computer Science* **43**, 169-188 (1986).

[LFKN-90] C. Lund, L. Fortnow, H. Karloff and N. Nisan. Algebraic Methods for Interactive Proof Systems. *Proceedings of the 31st Annual IEEE Symposium on the Foundations of Computer Science,* IEEE (1990).

[Ost-91] R. Ostrovsky. One-Way Functions, Hard on Average Problems, and Statistical Zero-Knowledge Proofs. *Proceedings of Structures In Complexity Theory 6th Annual Conference* IEEE (1991).

[OW-93] R. Ostrovsky and A. Wigderson. One-Way Functions are Essential For Non-Trivial Zero-Knowledge, *Proc. 2nd Israeli Symp. on Theory of Computing and Systems*, 1993.

[OVY-91] R. Ostrovsky, R. Venkatesan and M. Yung. Fair Games Against an All-Powerful Adversary. *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. Vol 13. (Jin-Yi Cai ed.) pp. 155-169.

[Sh-90] A. Shamir. IP=PSPACE. *Proc. 22nd ACM Symp. on Theory of Computing*, pages 11–15, 1990.

[Si-83] M. Sipser. A Complexity Theoretic Approach to Randomness. *Proceedings of the 15th Annual ACM Symposium on the Theory of Computing,* ACM (1983).

[St-83] L. Stockmeyer. The Complexity of Approximate Counting. *Proceedings of the 15th Annual ACM Symposium on the Theory of Computing,* ACM (1983).

A Proof of Lemma 5.2

The following Lemma is implicit in [GOP-94]. It was proven there as part of the proof of Lemma 4.2 where it was shown for a specific set A of accepting conversations. One should note that the proof holds for any set A and for the sake of self containment we provide their proof here.

Lemma 5.2: Let L, P, V, and M be as in Section 5, and fix any $x \in L$. Then, for any set A of conversations it holds that:

$$\mathrm{Prob}_{(\mathrm{P_M,V})}[A] \geq (\mathrm{Prob}_{(\mathrm{P,V})}[A])^2 \cdot 2^{-k(n)}.$$

Proof: The intuition of the proof is as follows. The set A has probability $\mathrm{Prob}_{(\mathrm{P,V})}[A]$ when P interacts with V and it has probability at least $2^{-k(n)} \cdot \mathrm{Prob}_{(\mathrm{P,V})}[A]$ in the output of the simulation, which can be thought of as P_M interacting with V_M (for a simulation based verifier V_M defined similarly to the simulation based prover). Now, when we look at a kind of "intermediate" interaction between P_M and V, we intuitively expect the probability of A in this case $\mathrm{Prob}_{(\mathrm{P_M,V})}[A]$ to be in-between the two probabilities or above the minimum of the two. The facts turn out to be almost like that and we actually have to loose an additional factor as $\mathrm{Prob}_{(\mathrm{P,V})}[A]$ is squared. The formal details follow.

Recall that we have perfect simulation and therefore there is a subset of the random tapes of the simulator, denoted S which has density at least 2^{-k} (for $k = k(n)$) and such that if we pick a random tape in S and run the simulation then we get exactly the distribution of conversations that are output during the original interaction of P and V (on x).

We begin by defining subsets of the possible random tapes of the simulator. Let Ω be all the possible random tapes of the simulator, let S be the "good" subspace of this set (i.e., if we run the

simulator on a uniformly chosen random tape in S we get a distribution which exactly equals the distribution of conversations between P and V, see Definition 2.1). Let Ψ be the set of random tapes of the simulator on which the simulator outputs conversations in the set A.

For any prefix h of a conversation, we define three corresponding subsets: Ω_h is the set of random tapes that make the simulation output a conversation of which h is a prefix. S_h contains the random tapes in S with the same property, i.e., $S_h = \Omega_h \cap S$. And last, we define $\Psi_h = S_h \cap \Psi$. This is the set of random tapes in the "good" subset on which the simulator outputs conversations in the set A having prefix h.

So let's check a few properties of these sets. First, $S = S_\lambda$ and $\Omega = \Omega_h$ (where λ is the empty string). Second, $|S_\lambda|/|\Omega_\lambda| \geq 2^{-k}$ since the density of S in the random tapes of the simulator is at least 2^{-k}. Also, since the simulator on a uniformly chosen random tape in S outputs the distribution of the original interaction between P and V, it also holds that $\mathrm{Prob}_{(\mathrm{P,V})}[A] = |\Psi_\lambda|/|S_\lambda|$. Another useful expression is that given a partial history h, the probability that the simulation based prover outputs the message α on a given history h is exactly $|\Omega_{h\circ\alpha}|/|\Omega_h|$. Also, since the simulator perfectly simulates the original interaction between P and V if we run it on the subset S, then we may write the probability that the original verifier answers β on a given history so far h as $|S_{h\circ\beta}|/|S_h|$.

We would like to show that

$$\mathrm{Prob}_{(\mathrm{P_M,V})}[A] \geq (\mathrm{Prob}_{(\mathrm{P,V})}[A])^2 \cdot 2^{-k}. \tag{12}$$

Actually, it is enough to show that

$$\mathrm{Exp}_c\left[\frac{|\Psi_c|^2}{|S_c| \cdot |\Omega_c|}\right] \geq \mathrm{Exp}_c\left[\frac{|\Psi_\lambda|^2}{|S_\lambda| \cdot |\Omega_\lambda|}\right] \tag{13}$$

Where the expectation over the languages c are taken by the distribution of conversations output by the interaction of P_M with V. Note that the right term is a constant which is actually equal to:

$$\left(\frac{|\Psi_\lambda|}{|S_\lambda|}\right)^2 \cdot \frac{|S_\lambda|}{|\Omega_\lambda|} \;=\; (\mathrm{Prob}_{(\mathrm{P,V})}[A])^2 \cdot 2^{-k}$$

Whereas the expression inside the expectation of the left term is 0 if $c \notin A$ and at most 1 if $c \in A$. Thus, the left term is smaller than $\mathrm{Prob}_{(\mathrm{P_M,V})}[A]$.

Inequality 13 involves a relation between sets describing full conversations (on the left side) and sets describing empty conversations (on the right side). We shall prove that the same inequality holds for any increase of one round in the conversations involved in the set description and thus by transitivity we shall get that Inequality 13 holds. For any round i, let c_i denote the first i rounds of a given conversation c. We will show that for all $0 \leq i \leq d-1$ (where d is the number of rounds) it holds that

$$\mathrm{Exp}_c\left[\frac{|\Psi_{c_{i+1}}|^2}{|S_{c_{i+1}}| \cdot |\Omega_{c_{i+1}}|}\right] \geq \mathrm{Exp}_c\left[\frac{|\Psi_{c_i}|^2}{|S_{c_i}| \cdot |\Omega_{c_i}|}\right] \tag{14}$$

Actually, we will show something stronger. We will show that *for any* prefix h of a conversation, it holds that

$$\sum_\beta \mathrm{Prob}_{(\mathrm{P_M,V})}(h \circ \beta | h) \cdot \frac{|\Psi_{h\circ\beta}|^2}{|S_{h\circ\beta}| \cdot |\Omega_{h\circ\beta}|} \geq \frac{|\Psi_h|^2}{|S_h| \cdot |\Omega_h|} \tag{15}$$

Where the summation is over all possible messages β that might follow the history h. Having proven Inequality 15, we get that this also holds when we take the expectancy over all possible h's of length i and Inequality 14 holds as well. So it remains to prove Equation 15 and we shall do

that separately for β being played in a prover round (i.e., by the simulation based prover) and for β being played in a verifier round (by the original verifier).

Prover's step: The left term of Equation 15 in this case is

$$\sum_{\beta} \text{Prob}(P_M(h) = \beta) \cdot \frac{|\Psi_{h \circ \beta}|^2}{|S_{h \circ \beta}| \cdot |\Omega_{h \circ \beta}|} = \sum_{\beta} \frac{|\Omega_{h \circ \beta}|}{|\Omega_h|} \cdot \frac{|\Psi_{h \circ \beta}|^2}{|S_{h \circ \beta}| \cdot |\Omega_{h \circ \beta}|}$$

The last equality is true since (by definition) P_M behave exactly like the simulator acts in prover steps. The above is equal to

$$\frac{1}{|\Omega_h|} \cdot \sum_{\beta} \frac{|\Psi_{h \circ \beta}|^2}{|S_{h \circ \beta}|} \geq$$

By the Cauchy-Schwartz Inequality, this is greater or equal to

$$\frac{1}{|\Omega_h|} \cdot \frac{(\sum_{\beta} |\Psi_{h \circ \beta}|)^2}{\sum_{\beta} |S_{h \circ \beta}|}$$

Since $\Psi_{h \circ \beta}$ is a partition of the set Ψ_h over all β, then it holds that $\sum_{\beta} |\Psi_{h \circ \beta}| = |\Psi_h|$. The same is true also for $S_{h \circ \beta}$ and S_h. Thus the expression above equals

$$\frac{|\Psi_h|^2}{|S_h| \cdot |\Omega_h|}$$

as needed.

Verifier's step: The left term of Equation 15 in this case is

$$\sum_{\beta} \text{Prob}(V(h) = \beta) \cdot \frac{|\Psi_{h \circ \beta}|^2}{|S_{h \circ \beta}| \cdot |\Omega_{h \circ \beta}|} = \sum_{\beta} \frac{|S_{h \circ \beta}|}{|S_h|} \cdot \frac{|\Psi_{h \circ \beta}|^2}{|S_{h \circ \beta}| \cdot |\Omega_{h \circ \beta}|}$$

The last equality is true since V behave exactly like the simulator acts on the random tapes in S. The above is equal to

$$\frac{1}{|S_h|} \cdot \sum_{\beta} \frac{|\Psi_{h \circ \beta}|^2}{|\Omega_{h \circ \beta}|} \geq$$

and again by the Cauchy-Schwartz Inequality, this is greater or equal to

$$\frac{|\Psi_h|^2}{|S_h| \cdot |\Omega_h|}$$

and we are done with the proof of Lemma 5.2. ■