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 $\mathcal{AM} \cap \text{co}-\mathcal{AM}$ [F-89, AH-91], 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 negligible error probability $\delta(n)$ is less than $2^{-3k(n)}$ (where $k(n)$ is the knowledge complexity) then the language proven is in the third level of the polynomial time hierarchy (specifically, it is in $\mathcal{AM}^{\mathcal{NP}}$. In the standard setting of negligible error probability, there exist PSPACE-complete languages which have sub-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 δ, ϵ, the verifier runs in time polynomial in $1/\epsilon$ and $\log(1/\delta)$ and fails with probability at most δ to detect an additive error ϵ in the entropy. We believe that this protocol is of independent interest. Subsequent to our work Goldreich and Vadhan [GV-98] established that the problem of comparing the entropies of two samplable distributions if they are noticeably different is a natural complete promise problem for the class of statistical zero knowledge ($\mathcal{SZK}$).

---

*This research was performed while the authors were visiting the Computer Science Department at the University of Toronto, preliminary version of this paper appeared in [PT-96]

†Dept. of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel. Email: erez@cs.technion.ac.il.

‡Rényi Institute, Hungarian Academy, POB. 127, Budapest, H-1364 Hungary. Partially supported by the grants OTKA T-020914, T-030059, and FKFP 0607/1999. E-mail: tardos@renyi.hu

1 Introduction

The ability of a party M to compute a function depends on the information it accesses and its computational power. This ability may increase when M interacts with another (possibly more powerful or more informed) party. The *knowledge complexity* measure, introduced by Goldwasser Micali and Rackoff [GMR-85, GMR-89], is meant to measure how much party M has gained through the interaction in this respect, alternatively phrased as the *amount of knowledge* gained by party M. The special case in which the interaction does not increase the computational ability of M at all is well known as zero knowledge 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 only be 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 implication 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 $\mathcal{NP}$-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. If we require the distributions of the simulator to be *equal* to the distribution of the real interaction we get the *perfect* zero-knowledge. (Yet another alternative is

computational zero-knowledge but we do not consider it here.)

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 Section 2.2.

We say that a language L has knowledge complexity $k(n)$ if there is an interactive proof for L with knowledge complexity $k(n)$. 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-91] (see also [H-94] for an intuition) 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. Namely, it is only guaranteed that the interaction between the prover and the *honest verifier*, i.e., the verifier that follows the protocol, can be simulated efficiently.

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 these knowledge complexities is negligible for any language. 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 $\mathcal{NP} \subseteq \text{co}-\mathcal{AM}$ 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 and the knowledge complexity of an interactive proof. We show that if a language has an interactive proof with negligible error probability $\delta(n)$ and statistical knowledge complexity $k(n)$ and if $\delta(n) \leq 2^{-3k(n)}$ then the language is in $\mathcal{AM}^{\mathcal{NP}}$ and so it is contained in the third level of the polynomial time hierarchy. We note that one may use the techniques in [GOP-94] to get a result for the case of logarithmic knowledge complexity. Specifically, if there exists an interactive proof for L with error $\delta(n)$ and logarithmic knowledge complexity $k(n)$ and if there exists a polynomial $p(n)$ such that $(1-\delta(n))^2 \cdot 2^{-k(n)} > \delta(n) + \frac{1}{p(n)}$, then the language in in $\mathcal{BPP}^{\mathcal{NP}}$. Our result applies only to negligible $\delta(n)$ but allows any knowledge complexity function that satisfies $k(n) \leq \frac{1}{3}\log_2(1/\delta(n))$. Let us say a few words on the implications of this result.

In the regular setting of zero-knowledge (or interactive proofs) it does not matter in the definition 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, the standard approach to reducing the error probability involves repeated applications 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. Following previous works we choose the reasonable requirement that the error probability be *negligible* (i.e., asymptotically smaller than any polynomial fraction).

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 δ into δ^t. Namely, the logarithm of $1/\delta$ and the knowledge complexity increase by the same factor. Assuming PSPACE$\neq \Sigma_3^P$, 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 $1/\delta$ cannot increase substantially more rapidly than the knowledge complexity for languages outside $\mathcal{AM}^{\mathcal{NP}}$.

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-90]). 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 $|h(x)| \leq k(|x|)$. (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, 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 negligible 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 in the language from x not in the language. This property is a modification of the separation property used in [AH-91]. 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 samplable distribution is of independent interest. We note that it is sublimed from a protocol in [AH-91] which is used there for a specific distribution.

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 the distribution of conversations in a mental experiment in which the original verifier interacts with the simulation-based prover, i.e., a prover that acts like the prover in the simulation (see Section 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 relates 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 3 we present our AM protocol for proving the entropy of a samplable distribution. In Section 4 we provide an overview of the construction in [AH-91] and explain why it doesn't work in the case that the knowledge complexity is greater than 0. In Section 5 we present the property of the simulator that tells apart inputs in the language from inputs not in the language. The tools presented in the above sections are used in Section 6 to present our main result: a constant round interactive proof for recognizing the languages in logarithmic statistical knowledge complexity. In Section 7 we present our result relating error probability to knowledge complexity of interactive proofs. In Section 8 we raise a few open questions.

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, V) constitutes an interactive proof for a language L if there exist negligible functions $\delta_c : \mathsf{N} \to [0,1]$ the *completeness error* and $\delta_s : \mathsf{N} \to [0,1]$ the *soundness error* such that

1. **Completeness:** If $x \in L$ then
$$\mathrm{Prob}\left[(P,V)(x) \text{ accepts }\right] \geq 1 - \delta_c(n)$$
2. **Soundness:** If $x \notin L$ then for any prover P^*
$$\mathrm{Prob}\left[(P^*,V)(x) \text{ accepts }\right] \leq \delta_s(n)$$

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 notation. For simplicity we also require that the verifier starts the conversation, and that the number of messages making up the conversation depends on the input length only.

The prover and the verifier speak in alternate rounds, the verifier taking the odd numbered rounds and the prover speaking in the even numbered rounds. 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 at least $2^{-k(n)}$, 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 $\bar{c}$

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

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

Note that the definition of perfect (statistical, corr.) knowledge complexity zero (i.e., when $k=0$) exactly matches the definition of perfect (and statistical, corr.) 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]

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

Note that this result is only proved for the honest verifier simulation—the definition of knowledge complexity presented here.

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

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

where M_t denotes the t message long prefix of the random conversation output by the simulator M. Notice that P_M is not defined for prefixes h output by M with zero probability.

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-91] 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 Section 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 $\text{Prob}_{\text{M}}[c]$.
2. The distribution of conversations in the original interactive proof (P, V). We denote the probability that a conversation c is output by this interactive proof by $\text{Prob}_{(\text{P,V})}[c]$.
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 $\text{Prob}_{(\text{P}_\text{M},\text{V})}[c]$.

All these distribution depend on the input x. In our notation we suppress x, the input should be clear from the context. 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 $\text{Prob}_{\text{M}}[c] \geq 2^{-k(n)} \cdot \text{Prob}_{(\text{P,V})}[c]$, where $k(n)$ is the perfect knowledge complexity.

Consider now the probability of a conversation c in the third distribution. We would like to express $\text{Prob}_{(\text{P}_\text{M},\text{V})}[c]$ using only probabilities of the form $\text{Prob}_{\text{M}}[c_i]$, where c_i is the i-round prefix of the conversation c. Let us partition the computation of $\text{Prob}_{(\text{P}_\text{M},\text{V})}[c]$ to a round-by-round computation:

$$\text{Prob}_{(\text{P}_\text{M},\text{V})}[c] = \prod_{i=1}^{d(n)} \text{Prob}_{(\text{P}_\text{M},\text{V})}[c_i | c_{i-1}],$$

where $d(n)$ is the number of messages sent by P and V. Recall that $d(n)$ is odd and the terms with i being odd are determined by the verifier V. Thus, if c is valid, i.e., the verifier moves are consistent with his coin-tosses and the history so far, then the product of all the odd terms equals the probability of V indeed picking the random tape specified in the end of the conversation c. Thus,

$$\prod_{i=0}^{\frac{d(n)-1}{2}} \text{Prob}_{(\text{P}_\text{M},\text{V})}[c_{2i+1} | c_{2i}] = 2^{-t(n)}$$

where $t(n)$ is the length of the random tape used by V.

The terms that have an even i are determined by P_M thus $\text{Prob}_{(\text{P}_\text{M},\text{V})}[c_{2i} | c_{2i-1}] = \text{Prob}_{\text{M}}[c_{2i} | c_{2i-1}]$. Here P_M is well defined if c is output by M with positive probability. For a valid transcript c with $\text{Prob}_{\text{M}}[c] > 0$ we thus have:

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

For an invalid conversation c we trivially have $\text{Prob}_{(\text{P}_\text{M},\text{V})}[c] = 0$. This simple rewriting of $\text{Prob}_{(\text{P}_\text{M},\text{V})}[c]$ was first noted in [AH-91].

3 Approximating the entropy in a constant number of rounds

Our first tool is an AM protocol for verifying the entropy of a polynomially samplable distribution to within an accuracy of $\frac{1}{\text{poly}}$. We consider this protocol to be of independent interest but emphasize that it is based on set size lower and upper bound protocols of [GS-89, F-89] and it is sublimed from a protocol in [AH-91], which is used there for a specific distribution. It is worth comparing this protocol to the one given in [GV-98] for the same purpose (or more exactly to approximate the difference of two entropies). Their public coin interactive proof has the advantage of having strong zero-knowledge properties, ours has the advantage of having a constant number of rounds. The [GV-98] protocol uses the set size lower bound protocol but instead of the upper bound protocol they use an elaborate "pushing game" from the paper [Oka-96], requiring more than constant number of rounds. We begin by explaining the setting.

Let D be a discrete distribution and we let $\text{Prob}_D[y]$ denote the weight of the element y in this distribution. The entropy $H(D)$ of D is defined as:

$$H(D) = -\sum_y \text{Prob}_D[y] \log \text{Prob}_D[y], \tag{2}$$

where the sum extends for all values y in the range of D.

We call an ensemble D_x of distributions *polynomially samplable* if there exist a polynomial time randomized machine whose output on input x is distributed according to D_x.

We state our result on approximating the entropy here but before the proof we recall the set-size approximation protocols needed for it.

Theorem 2 *Let D_x be a polynomially samplable ensemble of distributions. There exists a constant round upper bound interactive proof and a constant round lower bound interactive proof for the entropy $H(D_x)$ that on input x and $\delta, \epsilon, \alpha > 0$ satisfies:*

1. *The verifier runs in polynomial time in $|x|$, $1/\epsilon$, and $\log(1/\delta)$.*
2. *If the prover plays optimally then the verifier in the upper bound protocol accepts with probability at most δ if $H(f) \geq \alpha + \epsilon$ and rejects with probability at most δ if $H(f) \leq \alpha$.*
3. *Similarly, if the prover plays optimally then the verifier in the lower bound protocol accepts with probability at most δ if $H(f) \leq \alpha - \epsilon$ and rejects with probability at most δ if $H(f) \geq \alpha$.*

We later refer to the lower bound protocol mentioned in this theorem as an interactive proof for $H(f) \geq \alpha$ with accuracy ϵ and error δ.

As we shall see in Section 3.2 computing the entropy of a samplable distribution is equivalent to computing the average size of the set $f^{-1}(f(x))$ in logarithmic scale. Here x is taken uniformly from the domain $\{0,1\}^t$ of the efficiently computable function f.

3.1 Protocols for set sizes

For the sake of self containment, we include the set-size approximation protocols. For a more detailed description the reader may refer to [F-89, AH-91].

The main tool in these protocols is *universal family of hash functions* (sometimes denoted by universal$_2$ family of hash functions) [CW-79]. This is a collection H of functions mapping a domain D to the a range R such that for every point $X \in D$ and a random element $h \in H$, 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 existence of polynomial time universal families $H_{n,m}$ for $D = \{0,1\}^n$ and $R = \{0,1\}^m$ is well known (take for example the collection of affine linear maps over the two element field).

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. 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. For the simple proof see [AH-91].

Lemma 3.1 [GS-89] *If the prover plays optimally then the acceptance probability p in the above protocol satisfies*

$$1 - \frac{|R|}{|S|} \leq p \leq \frac{|S|}{|R|}.$$

Another way to state the lemma is that if $|S| < \epsilon|R|$ then the the verifier accepts with probability at most ϵ, but if $|R| < \epsilon|S|$ then the prover can make the verifier reject with probability less than ϵ.

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 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 $X = Z$.

The following lemma implies the completeness and soundness of the protocol.

Lemma 3.2 [F-89] *If the prover plays optimally, then the acceptance probability p of the above interactive proof protocol satisfies*

$$1 - \frac{|S| - 1}{|R|} \leq p \leq \frac{|R|}{|S|}.$$

Another way to state the lemma is that if $|R| < \epsilon|S|$ then the the verifier accepts with probability at most ϵ, but if $|S| - 1 < \epsilon|R|$ then the prover can make the verifier reject with probability less than ϵ.

The protocol is from [F-89] and its correctness is proved there. Nevertheless, we include the short proof here because the upper bound proved there on the acceptance probability is somewhat weaker and the proof (and the bound) is slightly more complicated.

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 pairwise 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 on the probability p, we assume, without loss of generality, that the prover chooses its optimal response for every message he receives deterministically. Let us fix the hash function h. For any possible value $\alpha \in R$ that the verifier may send, the prover has one (optimal) response $Z = Z(\alpha)$. So the prover has at most $|R|$ different possible answers and it can only win if the random element X that the verifier chooses in S is one of these values (recall that the verifier only accepts if $X = Z$). The probability that an X randomly chosen in S will fall into this set of at most $|R|$ element is at most $|R|/|S|$, and we are done with the proof of the Lemma. ■

Although the set-size approximation protocols just described are sufficient for the approximation of the entropy we need an improved lower bound protocol later for our main protocol (specifically, for the second step in the protocol for recognizing L or $\overline{L}$ in Section 6). 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, where m is an integer which depends on the desired accuracy.

Lemma 3.3 *For every $\epsilon > 0$ and $\delta > 0$ there is two-round 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, $1/\epsilon$ and $\log(1/\delta)$ and furthermore:*

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

We call such a protocol a proof for $|S| \geq s$ with relative accuracy $(1+\epsilon)$ and error δ. For self containment, we include the standard proof.

Proof: Setting m appropriately (see below), 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 $\{0,1\}^{\lfloor m \log((1-\epsilon/2)s) \rfloor}$. By Lemma 3.1 we get that the prover can make the acceptance probability at least $1-(1-\epsilon/2)^m$ if $|S| \geq s$ but it cannot make the verifier accept with probability more than $(1-\epsilon/2)^m$ if $s \geq |S|(1+\epsilon)$. Thus choosing $m = \lceil \frac{2}{\epsilon} \cdot \log \frac{1}{\delta} \rceil$ proves Lemma 3.3. ■

Note that a similar improvement over the upper bound protocol would require the verifier being 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.

3.2 Approximating the entropy

As a first step toward the proof of Theorem 2 we get an expression for the entropy that is more suitable for our purposes. As the distribution D_x is the distribution of a polynomial time randomized machine on input x we may consider this output as a function $f(r)$ on the random tape r. We fix the length of the random tape r to a suitable value t thus we have $r \in \{0,1\}^t$. Here f is computable in polynomial time (given x). We rewrite the definition of the entropy given by Equation 2 to get:

$$\begin{aligned} H(D_x) = H(f) &= -\sum_y \sum_{r:f(r)=y} \mathrm{Prob}(r) \log \mathrm{Prob}_s[f(s) = y] \\ &= -\sum_r \mathrm{Prob}(r) \log \mathrm{Prob}_s(f(s) = f(r)) \\ &= -\mathrm{Exp}_r[\log \mathrm{Prob}_s[f(s) = f(r)]] \\ &= t - \mathrm{Exp}_r[\log |f^{-1}(f(r))|]. \end{aligned}$$

Here $\mathrm{Prob}(r) = 2^{-t}$ is the probability of choosing r when uniformly sampling $\{0,1\}^t$, Exp_r denotes the expectation over a random r such selected, and Prob_s denotes probability with respect to a uniformly selected $s \in \{0,1\}^t$.

The idea of the protocol is to measure an empirical average value as an approximation to the expected value of $\log |f^{-1}(f(r))|$. We generate a large polynomial number m of independent random samples r_i and approximate the expectation by $-\frac{1}{m}\sum_{i=1}^m \log |f^{-1}(f(r_i))|$. We bound the probability of this approximation being far from the expectation by a variant of the Chernoff bound.

However, we cannot calculate the value inside the summation, i.e., given r_i it is hard to calculate $\log |f^{-1}(f(r_i))|$. Therefore, we use the set size lower and upper bound protocols of [GS-89, F-89] for this. Note that for lower bounding the entropy we need to upper bound $|f^{-1}(f(r_i)|$ and vice versa. For the entropy lower bound protocol one needs a uniform random element in the set. Fortunately as r_i is chosen uniformly in $\{0,1\}^t$ it is also uniform in $f^{-1}(f(r_i))$. 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.

We present both lower and upper bound protocols, although for proving our main result we use the lower bound protocol only.

We are given a function f defined on $\{0,1\}^t$, an approximation parameter $\epsilon > 0$ and an error parameter $\delta > 0$ and let the value α be the (lower or upper) bound on $H(f)$ that the prover would like to prove. Let m be a polynomial in t, $1/\epsilon$, and $\log(1/\delta)$ to be specified later. First, we reduce the error by using many copies of the function f. So consider the function F defined on the m-tuples of t-bit strings $D = \{0,1\}^{mt}$ by $F(r_1, \ldots, r_m) = (f(r_1), \ldots, f(r_m))$. For the upper bound we use a universal family of hash functions $H_{mt,u}$ from D to $\{0,1\}^u$, where $u = \lfloor m(t - \alpha - \epsilon/2) \rfloor$ and for the lower bound protocol we use a universal family of hash-functions $H_{mt,l}$ from D to $\{0,1\}^l$, where $l = \lfloor m(t - \alpha + \epsilon/2) \rfloor$.

We assume in both protocols that the verifier can compute $f(r)$ and thus also $F(X)$.

Let us start with the upper bound protocol.

- The verifier uniformly picks a random $X \in D$, a hash-function $h \in H_{mt,u}$ 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 $h(Z) = Y$.

Let us also present the lower bound protocol.

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

The following lemma states that the above protocols satisfy the conditions of Theorem 2.

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

1. *The verifier in both protocols runs in polynomial time in t, $1/\epsilon$, and $\log(1/\delta)$ 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 δ if $H(f) \geq \alpha + \epsilon$ and rejects with probability at most δ if $H(f) \leq \alpha$.*
3. *Similarly, if the prover plays optimally then the verifier in the lower bound protocol accepts with probability at most δ if $H(f) \leq \alpha - \epsilon$ and rejects with probability at most δ if $H(f) \geq \alpha$.*

Proof: Clearly, the statement on the efficiency of the verification process holds, since the verifier only has to sample the domain $\{0,1\}^{mt}$, to sample $h \in H_{mt,l}$ or $h \in H_{mt,u}$, 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 $\epsilon/4$. Call such a choice of X bad, and let us bound the probability of choosing a bad X using the Hoeffding Equation [Hoe-63] (a variant of the Chernoff bound). This inequality asserts that the probability of the average of m identically distributed independent variables deviating from the expected value by at least E is at most $2e^{-2E^2m/R^2}$ where R is the size of the range of the random variables. We can clearly make this less than $\delta/2$ by choosing $m > 8t^2 \log(1/\delta)/\epsilon^2$. So this source of error contributes only $\delta/2$ to the error probability. Let us continue and check the error probability that we get from the set-size lower and 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

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

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

$$2^{m(t-H(f)-\epsilon/4)} < |F^{-1}(F(X))| < 2^{m(n-H(f)+\epsilon/4)}. \quad (3)$$

Suppose X is not bad, and thus cardinality of $F^{-1}(F(X))$ is within the bounds specified in Equation 3. If the upper bound α, claimed by the prover is valid, i.e., $H(f) \leq \alpha$, then by Lemma 3.1 the verifier rejects with probability at most

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

If however $H(f) \geq \alpha + \epsilon$ then the probability of acceptance is at most $2^{m(t-H(f)+\epsilon/4)}/2^u < 2^{-3m\epsilon/4}$. Both these error probabilities can be made less than $\delta/2$ by making $m > 4(\log(1/\delta)+2)/\epsilon$. 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. ■

3.3 Remarks

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 upper 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 lower bound protocol.
A remark on the complexity of the function f: In the protocol derived from the previous remark Arthur evaluates f at the end of the game. This allows us to use the protocol to approximate the entropy not only of polynomial time computable functions but also for functions 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 of the techniques in [AH-91]

The main result of this paper is that $SKC(O(\log n)) \subset \mathcal{AM} \cap co-\mathcal{AM}$. This generalizes the result of Fortnow [F-89] and Aiello and Hastad [AH-91] stating $SZK \subset \mathcal{AM} \cap co-\mathcal{AM}$. Let us start by recalling the underlying techniques of the [AH-91] paper. This is done both because we are going to use some of the same techniques and to see why they don't suffice by themselves for our purposes.

4.1 The ideas in [AH-91]

The proof in [AH-91] 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 and its negation can be proven in AM.

We alter their argument a little bit. A similar simplified argument can also be found in [GV-98]. For simplicity we only consider perfect zero-knowledge here. We may assume without loss of generality that the simulator outputs mostly valid, accepting transcripts. 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.

The distinguishing property is actually the magnitude 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 as in Section 2.4. If $x \in L$ then the relative entropy $H(M(x)||(P_M, V)(x))$ is zero as the two distributions actually coincide, as we consider perfect zero knowledge. However if $x \notin L$ then much of the weight in $M(x)$ is concentrated on the set of accepting transcripts and that set has negligible weight in $(P_M, V)(x)$. It is well known (and easy to see) that the relative entropy $H(M(x)||(P_M, V)(x))$ is large in such a case. Recall that by the definition of relative entropy:

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

(We use the notations of Section 2.4.)

It is shown in [AH-91] how to prove that this relative entropy is big or small in AM. Using the approximation of the entropy described in Section 3, we can offer a more compact presentation of that protocol.

Recall Equation 1 from Section 2.4. For any valid transcript c with $\mathrm{Prob}_{\mathrm{M}}[c] > 0$ it holds that

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

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

Using Equation 1 we may rewrite the relative entropy

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

Here $H_M(c_i)$ 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.

4.2 Generalizing these techniques

Let us consider what happens with the relative entropy $H(M(x)||(P_M, V)(x))$ if the knowledge complexity is not zero. 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 $\mathrm{Prob}_{\mathrm{M}}[c] > 1/2$ but $\mathrm{Prob}_{(\mathrm{P_M,V})}[c] = 2^{-n}$ thus making $H(M(x)||(P_M, V)(x))$ 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:

$$H(M(x)||(P_M, V)(x)) = \mathrm{Exp}_{c \in M}[\log \frac{\mathrm{Prob}_{\mathrm{M}}[c]}{\mathrm{Prob}_{(\mathrm{P_M,V})}[c]}].$$

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 $x \notin L$.

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

In this section we formalize and prove the separation property motivated at the end of the preceding section. In the next section we use this property in the case of logarithmic perfect knowledge complexity to show that $x \in L$ (or $x \notin L$) in a constant round interactive proof. Thus, we get that $L \in \mathcal{AM} \cap \mathrm{co}{-}\mathcal{AM}$. In Section 7 we use the same property to lower bound the error probability of an interactive proof of a language outside $\mathcal{AM}^{\mathcal{NP}}$ in terms of its knowledge complexity.

We call a valid transcript c that leads to acceptance an *accepting* transcript. We denote this condition by $Ac(c)$.

Lemma 5.1 *Let (P,V) be an interactive proof for a language L. Let $\delta_s(n)$ be the soundness error probability and $k(n)$ be the perfect knowledge complexity of this proof and suppose that the completeness error probability $\delta_c(n)$ is at most $1/4$. Let M be the corresponding simulator and P_M the simulation-based prover. Let x be a string of length n and let $k = k(n)$ and $\delta = \delta_s(n)$.*

1. *If $x \notin L$ then the simulator outputs accepting conversations that have a small ratio with very small probability. Formally:*

$$\mathrm{Prob}_{\mathrm{M}}\left[Ac(c) \wedge \frac{\mathrm{Prob}_{\mathrm{M}}[c]}{\mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}[c]} \leq 2^{k+3}\right] \leq 2^{k+3}\delta.$$

2. *Whereas if $x \in L$ then the simulator has a substantial probability of outputting accepting transcripts with a small ratio. Formally,*

$$\mathrm{Prob}_{\mathrm{M}}\left[Ac(c) \wedge \frac{\mathrm{Prob}_{\mathrm{M}}[c]}{\mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}[c]} \leq 2^{k+2}\right] \geq 2^{-(k+2)}$$

Discussion: In our applications $\delta \cdot 2^{k+3}$ is much smaller than $2^{-(k+2)}$ thus the lemma separates the $x \in L$ and $x \notin L$ cases. Note that the bound on the ratio slightly differ in the two cases. This difference is important since we will not be able to compute this ratio precisely for a given transcript when applying the lemma. However, we will have means to approximate this ratio, and thus, we need the gap.

Proof: We begin by proving part 1 of the lemma. Let $b = 2^{k+3}$. Let A be the set of accepting conversation for which the ratio is small. Namely, for all $c \in A$, we have

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

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

First, by the definition of A, we know that

$$\mathrm{Prob}_{\mathrm{M}}[A] \leq \mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}[A] \cdot b \tag{4}$$

(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 δ, we have

$$\mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}(A) \leq \delta. \tag{5}$$

Combining Equations 4 and 5 we get that $\mathrm{Prob}_{\mathrm{M}}[A] \leq \delta \cdot b$ 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 k be the perfect knowledge complexity of the interaction between the probabilistic parties P and V, and let M be the corresponding simulator, P_M the simulation-based prover. Then, for any set A of conversations it holds that:*

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

For self containment, we provide the proof in the Appendix. 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

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

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

$$\text{Prob}_{\text{M}}[A'] \geq \text{Prob}_{(\text{P}_{\text{M}},\text{V})}[A'] \cdot 2^{k+2} \tag{6}$$

Using Lemma 5.2 we get that

$$\text{Prob}_{(\text{P}_{\text{M}},\text{V})}[A'] \geq (\text{Prob}_{(\text{P},\text{V})}[A'])^2 \cdot 2^{-k} \tag{7}$$

Combining Equations 6 and 7 we get

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

Let A be the set of accepting transcripts for which

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

Note that A contains all accepting conversations not in A'. In the original interaction (P, V), all conversations are valid and only an $\delta_c(n) \leq 1/4$ fraction is not accepting. Therefore,

$$\text{Prob}_{(\text{P},\text{V})}[A] \geq 1 - \text{Prob}_{(\text{P},\text{V})}[A'] - \delta_c(n) \geq \frac{1}{4}.$$

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

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

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 and 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. Recall that a language is in $\mathcal{SKC}(O(\log n))$ if it has an interactive proof with statistical knowledge complexity $O(\log n)$ and *negligible error*.

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).

Notice that in the separation result Lemma 5.1 the probability is a polynomial fraction in one case and it is negligible in the other.

In our protocol on input x of length n the new verifier V', with the help of the new prover P', approximates the probability that a conversation c, output by the simulator $M(x)$, is accepting and satisfies $\log \frac{\text{Prob}_{\text{M}}[c]}{\text{Prob}_{(\text{P}_{\text{M}},\text{V})}[c]} \leq k + 2.5$, where $k = k(n)$. 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 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 $d-1$ sets (where $d = d(n)$ 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 Lemma 5.1.

We approximate the sizes of the sets involved in the following way. The prover states the size of the set and then he proves 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 sets to be approximated. Unfortunately, this hidden random element can only be used once. After that, the seed is not hidden any more, 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 $\mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}[c]$ 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 Section 2.4 one gets that for valid transcripts c with $\mathrm{Prob}_{\mathrm{M}}[c] > 0$

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

Here $t = t(n)$ is the length of the random tape of V, $d = d(n)$ is the (odd) number of messages exchanged by P and V and c_i denotes the i-message prefix of c. Recall our convention that V speaks in odd rounds and P speaks in even rounds. Denoting the length of the random tape of M by $t' = t'(n)$ we have $|\Omega| = 2^{t'}$ and for valid transcripts c with $\mathrm{Prob}_{\mathrm{M}}[c] > 0$

$$\log \frac{\mathrm{Prob}_{\mathrm{M}}[c]}{\mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}[c]} = t - t' - \sum_{i=1}^{d(n)} (-1)^i \log |\Omega_{c_i}|. \tag{8}$$

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 $\delta_0 = 2^{-n}$, the quality of approximations is set to $\epsilon = 2^{-k(n)-9}/(d(n))^2$, and the number of simulator conversations that we check is $\ell = \lceil n(t'(n))^2/\epsilon^2 \rceil$. Notice that as a function of n δ_0 is negligible, ϵ 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 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 using the protocol of Lemma 3.3 and with relative accuracy $1 + \epsilon$ and error probability δ_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 ($j = 1, 2, \ldots, \ell$) an approximation of $\log(\mathrm{Prob}_{\mathrm{M}}[c^j]/\mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}[c^j])$ 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 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 $h_i = 1/\ell \sum_{j=1}^{\ell} (t'(n) - \log \omega_i^j)$.

Finally, the prover P' proves that $h_i - \epsilon$ 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 d lower bounds with accuracy ϵ and error δ_0 in parallel using the protocol of Theorem 2. 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 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 $k(n) = O(\log n)$ and the soundness and completeness error probabilities $\delta_s(n)$ and $\delta_c(n)$ are negligible, we may assume in what follows that $\delta_s(n) < 2^{-2k(n)-7}$ and $\delta_c(n) < 1/4$. 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 accepting, and have $\log(\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]) \leq k + 3$ 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 ϵ to $2e^{-2\epsilon^2\ell}$. 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 accepting conversations with $\log(\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]) \leq k + 2$.

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

We call a choice of the random conversations c^j $(j = 1, 2, \ldots, \ell)$ 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.

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 Lemma 5.1 (2) rejection can come only from errors in the set size lower bound protocols or the entropy lower bound protocols. Since we run only $(\ell + 1)d$ such protocols and since the probability to make an error in any one of them is at

most $\delta_0 = 2^{-n}$, then the probability of such error is at most $(\ell+1)d\delta_0$ which is negligible. The proof of completeness of the protocol for $\overline{L}$ is similar using Lemma 5.1 (1).

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 $(1 \le i \le d,\ 1 \le j \le \ell)$ such that one of them satisfies $\omega_i^j > (1+\epsilon)|\Omega_{c_i^j}|$ In this case he passes the lower bound protocol with probability at most δ_0. So assume that for all the values ω_i^j stated by the prover it holds that $\omega_i^j \le (1+\epsilon)|\Omega_{c_i^j}|$, i.e., the stated values are never too high.

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

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

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., $H_i \ge H(c_i) - \epsilon$. Thus:

$$h_i \ge H_i + 3\epsilon \ge H(c_i) + 2\epsilon.$$

So when the prover tries to show that $h_i - \epsilon \le H(c_i)$ (using the entropy lower bound protocol) he will succeed with probability at most δ_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 $v_j \le k + 2.5$ is less then $\ell \cdot 2^{-k-3}$ and thus the verifier rejects. If neither of the above two possibilities happen then for all indices except for at most $15\epsilon d^2\ell$ pairs (i,j) we have

$$2^{-1/(3d)} \cdot |\Omega_{c_i^j}| \ \le\ \omega_i^j \ \le\ (1+\epsilon)|\Omega_{c_i^j}|. \tag{9}$$

Furthermore, the number of conversations c^j for which Equation 9 holds for all rounds i is at least $\ell - 15\epsilon d^2\ell$. For such a conversation c^j, the verifier's approximation of $\log(\mathrm{Prob}_{\mathrm{M}}[c^j]/\mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}[c^j])$ is correct to within 1/3. Namely,

$$v_j \ = \ t - t' - \sum_{i=1}^{d}(-1)^i \log \omega_i^j$$

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

We call these conversations "well approximated". Therefore, if a conversation is well approximated, and $v_j \leq k+2.5$, then we also get that for this conversation $\log(\mathrm{Prob}_{\mathrm{M}}[c^j]/\mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}[c^j]) \leq k+3$. By Lemma 5.1 (1), we know that the probability that a conversation output by the simulator is accepting and having $\log(\mathrm{Prob}_{\mathrm{M}}[c^j]/\mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}[c^j]) < k+3$ is at most $\delta_s(n)\cdot 2^{k+3}$. Also, since the set of conversations is not bad, then the actual fraction of conversations for which $\log(\mathrm{Prob}_{\mathrm{M}}[c_j]/\mathrm{Prob}_{(\mathrm{P_M},\mathrm{V})}[c_j]) < k+3$ is at most $\delta_s(n)\cdot 2^{k+3}+\epsilon$.

Thus the number of "good conversations" counted by the verifier is limited to $(2^{k+3}\delta_s(n)+\epsilon)\ell + 15\epsilon d^2\ell$. By the setting of ϵ and the assumption $\delta_s(n) < 2^{-2k-7}$ we get that this is at most $2^{-k-3}\ell$ 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 δ_0 for the same reasons. In the third case recall that $\delta_c(n) < 1/4$ and use Lemma 5.1 (2) to show that the verifier sees more than $2^{-k-3}\ell$ accepting conversations with $v_j \leq k+2.5$ and thus the verifier rejects. ■

A remark about the precision of calculations: During the protocol, the verifier is required to compute $v_j = t - t' - \sum_{i=1}^{d}(-1)^i \log\omega_i^j$ and $h_i = 1/\ell \sum_{j=1}^{l}(t' - \log\omega_i^j)$, which involves calculations with 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 $\epsilon/2$ and make the protocol itself be accurate to within a $\epsilon/2$ approximation error. Thus the overall approximation error is below ϵ.

7 The connection between knowledge and error

In this section we state that if a language L has an interactive proof whose soundness 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 perfect knowledge complexity $k(n)$, soundness error probability $\delta(n) \leq 2^{-(2k(n)+6)}$, completeness error probability below $1/4$ 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 computationally 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 relativizes.) Note also that $k(n)$ has to be computable in polynomial time in n and not in $\log n$, so the restriction is quite liberal.

We also remark that in this theorem our explicit bounds on the error probability replaces the requirement for them in the definition of the interactive proof: they need not be negligible.

As mentioned in the abstract, standard definitions of interactive proofs allow any negligible error probability. In this case, one has PSPACE-complete languages which have sub-linear knowledge complexity. This can be deduced from the result [LFKN-90, Sh-90] that PSPACE-complete languages have interactive proofs using standard padding techniques. Applying enough polynomial padding to a PSPACE-complete language it remains PSPACE-complete but the interactive proof for it becomes sub-linear in length and thus in the knowledge it reveals. 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$= \Sigma_3^P$.

Proof: The proof is based on the observation that Lemma 5.1 still separates the elements of L from the non-elements. Let us call a conversation c *good* if it is 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 counting good conversations 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 is a set-size lower bound protocol.

In the set size lower bound described in Section 3.1 it is required that the verifier is able to recognize elements in the set. In our case, checking if c is 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 8 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 probabilistic computation with access to an $\mathcal{NP}$ oracle. The approximation there fails with negligible probability to give an approximation with relative accuracy $1 + \frac{1}{\mathrm{poly}}$.

We apply the protocol of Lemma 3.3 to prove $|S| > 2^{-(k(n)+2)}\Omega$ with relative accuracy $1/2$ and negligible error, where S is the set of random tapes that cause M to produce good conversations and Ω is the set of all random tapes of M. 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 8. We can set the relative 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 an 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 accepting or if $\log(\mathrm{Prob}_{\mathrm{M}}[c]/\mathrm{Prob}_{(\mathrm{P_M,V})}[c]) > k(n) + 3$. Lemma 5.1 shows that this is enough for our purposes. ■

We can use [GOP-94] again to extend the above result to statistical knowledge complexity. Here however it is not enough to cite Theorem 1, we actually need some of the details of the transformation of the interactive proof in that theorem. This more detailed statement is implicit in [GOP-94].

Lemma 7.1 [GOP-94] *Let (P,V) be an interactive proof for a language L with statistical knowledge complexity $k(n)$ soundness error $\delta_s(n)$ and completeness error $\delta_c(n)$. Then there exists a prover P' such that (P',V) is an interactive proof for L with perfect knowledge complexity $k(n)+O(\log n)$, with completeness error $\delta_c(n)+\delta_0(n)$ for some negligible fraction $\delta_0(n)$, and with the same soundness error $\delta_s(n)$.*

Note that we added specific statement on how the transformation [GOP-94] preserves the errors in the transformation: the soundness error does not change at all (since the verifier is not modified) and the completeness error only increases by a negligible fraction.

Corollary 7.2 *If there is a interactive proof for a language L with statistical knowledge complexity $k(n)$, negligible soundness error probability $\delta_s(n) < 2^{-3k(n)}$ and completeness error probability below $1/5$ and if $k(n)$ is computable in polynomial time, then $L \in \mathcal{AM}^{\mathcal{NP}}$.*

Proof: By Lemma 7.1 the same language L has an interactive proof with perfect knowledge complexity $k'(n) = k(n)+O(\log n)$, the same soundness error probability and with completeness error probability below $1/4$ for large enough n. We have that $\delta_s(n) < 2^{-(2k'(n)+6)}$ also holds for large enough n since $\delta_s(n)$ is both negligible and bounded by $2^{-3k(n)}$. Thus Theorem 4 is applicable and proves Corollary 7.2. ■

8 Open questions

Many questions regarding the relation between knowledge complexity and computational complexity are still open. Can one show a higher (conditional) 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 (conditional) lower bound, which is higher than the super-logarithmic lower bound we give here, is known on the knowledge complexity of any language. Does the unlikely assumption $PSPACE = PKC(\log^2 n)$ imply that the polynomial hierarchy collapses (or another similar consequence)?

Let us now consider the low end of the knowledge complexity hierarchy. 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, does $SKC(O(\log n)) = SKC(0)$? Is it even true that $SKC(1) = SKC(0)$? Or can one give indications that this is not the case?

It is also open how rich the knowledge complexity hierarchy of languages is. For example, Is there a constant factor collapse? Namely, is $SKC(2k(n)) = SKC(k(n))$?

The statement of Theorem 3 is symmetric, it claims the same about the languages having low knowledge complexity and about their complements. The same cannot be said about Theorem 4 and Corollary 7.2. This asymmetry comes from the more demanding requirements of set size upper bound protocols. Theorem 4 implies that languages having certain interactive proofs are in Π_3^P. Can one prove that the same languages are in Σ_3^P? A bolder goal would be to prove that certain knowledge complexity classes, say $SKC(O(\log n))$ are

closed under complementation. This would extend the work in [Oka-96, GV-98] establishing this for SZK.

Our main result (Theorem 3) bounds the computational complexity of languages having *negligible error* interactive proofs leaking only logarithmic knowledge. It is not clear what can be said if the soundness error probability is allowed to be high. Our techniques break down as soon as Lemma 5.1 does not provide a separation.

9 Acknowledgment

We would like to thank Rafail Ostrovsky for helpful discussions and the anonymous referees for their many enlightening remarks.

Appendix: 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. For the sake of self containment we provide their proof here.

Lemma 5.2 (restated): *Let k be the perfect knowledge complexity of the interaction between the probabilistic parties P and V, and let M be the corresponding simulator, P_M the simulation-based prover. Then, for any set A of conversations it holds that:*

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

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 $2^{-k} \cdot \mathrm{Prob}_{(\mathrm{P,V})}[A]$ in the output of the simulation, which can be thought of as P_M interacting with V_M. (The simulation-based verifier V_M is defined similarly to the simulation-based prover P_M.) When we look at a kind of "intermediate" interaction between P_M and V, we intuitively expect the probability $\mathrm{Prob}_{(\mathrm{P_M,V})}[A]$ to be in-between the two probabilities or above the minimum of the two. This is not necessarily true, i.e., the probability of events in the intermediate interaction is not always in between the two interactions, but this intuition does lead to the above Lemma, which looses an additional factor as $\mathrm{Prob}_{(\mathrm{P,V})}[A]$ is squared. The formal details follow.

Proof: Recall that the perfect simulation means that there is a subset of the random tapes of the simulator, denoted S, which has density at least 2^{-k} 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.

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 mentioned above. Let Ψ be the set of good 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 $\Omega = \Omega_\lambda$ (where λ is the empty string). Second, $|S_\lambda|/|\Omega_\lambda| \geq 2^{-k}$, this is the density of S in the random tapes of the simulator. 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 $\text{Prob}_{(\text{P,V})}[A] = |\Psi_\lambda|/|S_\lambda|$. 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 $|\Omega_{h\circ\alpha}|/|\Omega_h|$. We may write the probability that the original verifier answers β on a given history h as $|S_{h\circ\beta}|/|S_h|$.

We would like to show that

$$\text{Prob}_{(\text{P}_\text{M},\text{V})}[A] \geq (\text{Prob}_{(\text{P,V})}[A])^2 \cdot 2^{-k}. \tag{10}$$

Using the fact that $\text{Prob}_{(\text{P,V})}[A] = \frac{|\Psi_\lambda|}{|S_\lambda|}$, we have

$$\left(\frac{|\Psi_\lambda|}{|S_\lambda|}\right)^2 \cdot \frac{|S_\lambda|}{|\Omega_\lambda|} \geq \left(\text{Prob}_{(\text{P,V})}[A]\right)^2 \cdot 2^{-k} \tag{11}$$

and since $|\Psi_c| \leq |S_c| \leq |\Omega_c|$ for every c, and Ψ_c is empty for a complete transcript $c \notin A$, we have

$$\text{Prob}_{(\text{P}_\text{M},\text{V})}[A] \geq \text{Exp}_c\left[\frac{|\Psi_c|^2}{|S_c| \cdot |\Omega_c|}\right]. \tag{12}$$

Here and in the rest of this appendix Exp_c denotes the expectation over the random conversation c output by P_M and V. Note that a problem rises here for conversations c that have positive probability in the interaction (P_M, V) but cannot occur in the original interaction (P, V). In this case, we have $|S_c| = |\Psi_c| = 0$ and in the above expectation we get a division of zero by zero. Thus, we modify the expectation to sum only over conversations that have positive probability in the original interaction (P, V). In other words, in this expectation, we define $|\Psi_c|^2/(|S_c| \cdot |\Omega_c|)$ to be zero for conversations c with $S_c = \emptyset$. Using Equation 11 and 12 we get that in order to prove that Equation 10 holds, it is enough to show that

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

Equation 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 Equation 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

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

Actually, we will show something stronger. We will show that *for any* prefix h of a conversation that has positive probability in the original interaction (P, V) (i.e., with $S_h \neq \emptyset$), it holds that

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

Where the summation is over all possible messages β that might follow the history h in the original interaction (P, V). Having proven Equation 15, we get that this also holds when we take the expectation over all possible h of length i and Equation 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

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

The last equality is true since (by definition) P_M behave exactly like the simulator acts in prover steps. By the Cauchy-Schwartz inequality we can write

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

The sets $\Psi_{h\circ\beta}$ over all β satisfying $S_{h\circ\beta} \neq \emptyset$ are a partition of the set Ψ_h since $\Psi_{h\circ\beta} \subseteq S_{h\circ\beta}$. Thus, 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 on the right 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

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

The last equality is true since V behave exactly like the simulator acts on the random tapes in S. Using Cauchy-Schwartz again, the above is equal to

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

Note again that the sets $\Psi_{h\circ\beta}$ over all β satisfying $S_{h\circ\beta} \neq \emptyset$ are a partition of the set Ψ_h since $\Psi_{h\circ\beta} \subseteq S_{h\circ\beta}$. The sets $\Omega_{h\circ\beta}$ over all β such that $S_{h\circ\beta} \neq \emptyset$ are not necessarily a partition of Ω_h as nonempty parts corresponding to β with $S_{h\circ\beta} = \emptyset$ may be missing. Thus, we can only claim that $\sum_{\beta} |\Omega_{h\circ\beta}| \leq |\Omega_h|$, but this is good enough for us, and we are done with the proof of Lemma 5.2. ∎

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-91] W. Aiello and J. Håstad. Perfect Zero-Knowledge can be Recognized in Two Rounds. *JCSS*, Vol. 42, pages 327–345, 1991.

[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-90] R. Bar-Yehuda, B. Chor, E. Kushilevitz, and A. Orlitsky, Privacy, Additional Information, and Communication, *IEEE Transactions on Information Theory*, Vol. 39, No. 6, November 1993, pp. 1930-1943.

[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] M. Furer, O. Goldreich, Y. Mansour, M. Sipser and S. Zachos. On Completeness and Soundness in Interactive Proof Systems. *Advances in Computing Research: A Research Annual, Vol. 5, Randomness and Computation* (ed. S. Micali), 1989, pp. 429-442.

[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.

[GV-98] O. Goldreich and S. Vadhan, Comparing Entropies in Statistical Zero-Knowledge with Applications to the Structure of this Class. In Proceedings of the 14th Annual IEEE Conference on Computational Complexity, 1999.

[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 Knowledge Complexity. To appear in *SIAM Journal on Computing.* A preliminary version appeared in *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-91], 1994.

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

[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. *Journal of the ACM*, 39 (04), page 859-868 (1992).

[Oka-96] T. Okamoto, On relationships between statistical zero-knowledge proofs. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing,*, pp. 649–658, 1997.

[PT-96] E. Petrank and G. Tardos. On the knowledge complexity of NP. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science,*, pp. 494–503, 1996.

[Sh-90] A. Shamir. IP=PSPACE. Journal of the ACM, 39(4), pp. 869-877, 1992 A preliminary version appeared in *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).