Electronic Colloquium on Computational Complexity - Reports Series 1996 - available via:
ECCC
TR96-018
FTP: ftp.eccc.uni-trier.de:/pub/eccc/
WWW: http://www.eccc.uni-trier.de/eccc/
Email: ftpmail@ftp.eccc.uni-trier.de with subject 'help eccc'

On the Message Complexity of Interactive Proof Systems

Oded Goldreich*
Department of Computer Science
and Applied Mathematics
Weizmann Institute of Science
Rehovot, ISRAEL.
oded@wisdom.weizmann.ac.il

Johan Håstad
Department of Computer Science
Royal Institute of Technology
10044 Stockholm, SWEDEN.
johanh@nada.kth.se

February 23, 1996

Abstract

We investigate the computational complexity of languages which have interactive proof systems of bounded message complexity. In particular, we show that

- If L has an interactive proof in which the total communication is bounded by $c(n)$ bits then L can be recognized a probabilitic machine in time exponential in $O(c(n) + \log(n))$.
- If L has an AM-proof in which the prover sends $c(n)$ bits then L can be recognized a probabilitic machine in time exponential in $O(c(n)\log(c(n)) + \log(n))$.
- If L has an interactive proof in which the prover sends $c(n)$ bits then L can be recognized a probabilitic machine with an NP-oracle in time exponential in $O(c(n)\log(c(n)) + \log(n))$.

*Work done while being on a sabatical leave at LCS, MIT.

1 Introduction

In 1992, Kilian demostrated that *computationally-sound* proof systems (aka 'argument systems' [3]) may be able to recognize any language in $\mathcal{NP}$ while using only polylogarithmic message complexity [6]. Specifically, assuming the existence of hashing functions for which collisions cannot be found by subexponential circuits, Kilian showed that any $L \in \mathcal{NP}$ has a *computationally-sound* proof systems in which both the bi-directional message complexity and the randomness complexity are polylogarithmic. Furthermore, this proof system is in the public-coins model (i.e., is an Arthur-Merlin game) of Babai [1].

Kilian's result should be contrasted with the following observation, which indicates that Kilian's result is unlikely for interactive proof systems (rather than argument systems).

Theorem 1 (interactive proofs with bounded message and randomness complexities): *Let $c(\cdot)$ be an integer function and $L \subseteq \{0,1\}^*$. Suppose that L has an interactive proof system in which both the randomness and communication complexities are bounded by $c(\cdot)$. Then $L \in \mathrm{Dtime}(2^{O(c(\cdot))} \cdot \mathrm{poly}(\cdot))$.*

Proof: Consider the *tree of all possible executions* of the proof system on input $x \in \{0,1\}^*$. (This tree is defined in the next section and considering it is standard practice.) This tree has at most $2^{2c(|x|)}$ leaves and its value can be easily computed within the stated time bound. ■

Theorem 1 is the starting point of our investigation. Its proof is easy since the hypothesis contains a bound on the randomness complexity of the verifier. However, what we consider fundamental in Kilian's result is the low message complexity. Thus, we wish to waive the extra hypothesis. In fact, waiving the bound on the randomness complexity, we obtain a very similar bound

Theorem 2 (interactive proofs with bounded message complexity): *Let $c(\cdot)$ be an integer function and $L \subseteq \{0,1\}^*$. Suppose that L has an interactive proof system in which the communication complexity is bounded by $c(\cdot)$. Then $L \in \mathrm{BPtime}(2^{O(c(\cdot))} \cdot \mathrm{poly}(\cdot))$.*

Theorem 2 refers to interactive proof system in which the bi-directional communication complexity is bounded. However, it seems that the more fundamental parameter is the uni-directional communication complexity in the prover-to-verifier direction. In fact, waiving also the bound on the verifier's message length, we obtain a similar bound for the special case of Arthur-Merlin interactive proof systems. Namely,

Theorem 3 (Arthur-Merlin proofs with bounded prover-messages): *Let $c(\cdot)$ be an integer function and $L \subseteq \{0,1\}^*$. Suppose that L has an Arthur-Merlin proof system in which the total number of bits sent by the prover is bounded by $c(\cdot)$. Then $L \in \mathrm{BPtime}(2^{O(c(\cdot)\log c(\cdot))} \cdot \mathrm{poly}(\cdot))$.*

Theorem 3 may not hold for general interactive proofs, and if it does this may be hard to establish. The reason being that Graph Non-Isomorphism has an interactive proof system in which the prover sends a single bit [4]. Thus, we are currently content with a weaker result.

Theorem 4 (interactive proofs with bounded prover-messages): *Let $c(\cdot)$ be an integer function and $L \subseteq \{0,1\}^*$. Suppose that L has an interactive proof system in which the total number of bits sent by the prover is bounded by $c(\cdot)$. Then $L \in \mathrm{BPtime}(2^{O(c(\cdot)\log c(\cdot))} \cdot \mathrm{poly}(\cdot))^{NP}$.*

2 Formal Treatment

We assume that the reader is familiar with the basic definitions of interactive proofs as introduced by Goldwasser, Micali and Rackoff [5] and Babai [1]. Here we merely recall them, while focusing on some parameters.

Definition 1 (interactive proof systems):

- *An* interactive proof system for a language *L is a pair (P,V) of interactive machines, so that V is probabilistic polynomial-time, satisfying*
  - *Completeness: For every $x \in L$ the verifier V, the verifier V accepts with probability* at least $\frac{2}{3}$, *after interacting with P on common input x.*
  - *Soundness: For every $x \notin L$ and every potential prover P^*, the verifier V accepts with probability* at most $\frac{1}{3}$, *after interacting with P^* on common input x.*

  An interactive proof system is said to be an Arthur-Merlin game *if the verifier's message in each round consists of all coins it has tossed in this round.*
- *Let m and r be integer functions. The complexity class $\mathcal{IP}(m(\cdot),r(\cdot))$* (resp., *$\mathcal{AM}(m(\cdot),r(\cdot))$ consists of languages having an interactive proof system* (resp., an Arthur-Merlin proof system) *in which, on common input x, the interaction consists of at most $r(|x|)$ communication rounds during which the* total *number of bits sent* from the prover to the verifier *is bounded by $m(|x|)$.*

For an integer function t, we let $\mathrm{BPtime}(t(\cdot))$ (resp., $\mathrm{BPtime}(t(\cdot))^{NP}$) denote the class of languages recognizable by probabilistic $t(\cdot)$-time machines (resp., orcale machines with access to an oracle set in $\mathcal{NP}$) with error at most $1/3$. Our main result is

Proposition 5 (interactive proofs with bounded message and round complexity):

$$\mathcal{AM}(m(\cdot),r(\cdot)) \subseteq \mathrm{BPtime}(2^{O(m(\cdot)+r(\cdot)\log r(\cdot))} \cdot \mathrm{poly}(\cdot)) \quad (1)$$

$$\mathcal{IP}(m(\cdot),r(\cdot)) \subseteq \mathrm{BPtime}(2^{O(m(\cdot)+r(\cdot)\log r(\cdot))} \cdot \mathrm{poly}(\cdot))^{NP} \quad (2)$$

Theorem 3 follows from Part (1) of Proposition 5, whereas Theorem 4 follows from Part (2).

2.1 The Tree of all Possible Executions

The main ingridient of our proof is a probabilistic procedure for evaluating the value of the *tree of all possible executions of a proof system*. Let (P,V) be an interactive proof system in which, on common input x, the interaction consists of at most $r(|x|)$ communication rounds during which the total number of bits sent from the prover to the verifier is bounded by $m(|x|)$. In defining the tree of possible executions of (P,V) we assume, for simplicity, that the interaction between the P and V on input x consists of exactly $r \stackrel{\mathrm{def}}{=} r(|x|)$ rounds so that the i^{th} round starts with the verifier sending a $\mathrm{poly}(|x|)$-bit long message which is replied by a prover message of length m_i so that $\sum_{i=1}^{r} m_i = m(|x|)$. In case (P,V) is not an Arthur-Merlin system, we assume that in the last round the verifier sends to the prover the outcome of all coins flipped by it during the execution.

Definition 2 (the tree of all possible executions of a proof system): *The nodes in the tree correspond to possible prefixes of the execution of V with some prover P^*. Nodes which correspond to prefixes ending with a V-message are called* prover nodes *and the other are* verifier nodes. *The root is called the 0-level of the tree. Nodes at the $(2i-1)^{\text{st}}$ level are prover's nodes and each has 2^{m_i} children. Nodes at the even levels are verifier's nodes and each has upto $2^{\text{poly}(|x|)}$ children. The* leaves are verifier nodes *and their* value *is either 0 or 1 depending on whether V accepts in this execution or not. The* value of an internal prover node *is defined as the maximum of the values of its children. The* value of an internal verifier node *is defined as the weighted average of the values of its children, where the weights correspond to the probabilities of the various verifier messages. The* value of the tree *is defined as the value of its root.*

Clearly, to decide if x is in the language accepted by (P,V), it suffices to approximate the value of the corresponding tree of possible executions. This can be done by the following procedure

2.2 Evaluating the tree of all possible executions

To evaluate a tree such as in Definition 2, we select for each (verifier) node at the $2i^{\text{th}}$ level a sample of $O(m^4)$ children. The sample is selected according to the weights (i.e., the probabilities of the various verifier's messages). Note that the sample may contain several occurences of the same node. This defines an *approximation tree* in which each node at the $(2i-1)^{\text{st}}$ level has 2^{m_i} children, whereas each node at the $2i^{\text{th}}$ level has $\text{poly}(m)$ children. The value of the approximation tree is defined recursively as above, where the leaves have the same value as in the tree of all possible executions, the value of nodes at odd levels is the maximum of the value of their children, and the value of nodes at even level is the (unweighted) average of the values of their children.

Lemma 6 (evaluating the tree of all possible executions): *With probability at least* 3/4, *the value of the approximation tree is within* 0.1 *away from the value of the corresponding tree of all possible executions.*

The total size of the approximation tree is

$$\prod_{i=1}^{r} (2^{m_i} \cdot \text{poly}(m)) = \text{poly}(2^m \cdot m^r) = \text{poly}(2^m \cdot r^r)$$

(For the last equality use $2^m \cdot m^r \leq (2^m \cdot r^r)^2$, which can be shown by considering two cases w.r.t the relative sizes of 2^m and m^r.) Thus, once Lemma 6 is proven, Part (1) of Proposition 5 follows immediately. To prove Part (2) of Proposition 5, we merely use the uniform generation procedure of Bellare and Petrank [2] in order to sample children at the verifier nodes. Since the latter procedure can be implemented in probabilistic polynomial-time with access to an oracle in $\mathcal{NP}$, Part (2) follows.

Proof of Lemma 6: Let $s = O(m^4)$ be the size of the sample used for each verifier node. Using Chernoff bound, we observe that with probability at least $1-2^{-m^2}$ the approximation at any specific verifier node is within $\frac{1}{10r}$ of the expected value. Since the number of verifier nodes is bounded by $2^m \cdot s^r = O(2^m \cdot m^{4r}) < \frac{1}{4} \cdot 2^{m^2}$ we conclude that with probability at least $\frac{3}{4}$ the approximation at every verifier node is within $\frac{1}{10r}$ of the expected value. In such a case the value of the root of the approximation tree is within 0.1 of the value of the tree of all possible executions, and the lemma follows. ■

2.3 Proof of Theorem 2

Finally, we prove Theorem 2. Let $c(\cdot)$ and $L \subseteq \{0,1\}^*$ be as in the theorem, and consider a fixed $x \in \{0,1\}^*$. Let $c \stackrel{\text{def}}{=} c(|x|)$. Considering the tree of all possible executions (on input x), we associate with each internal node the probability that this node is reached assuming that the prover's moves do match the path to this node. Note that the probability associated with each node is the product of the probabilities of the corresponding verifiers' moves along this path. Nodes which are reached with probability smaller than, say, $0.01 \cdot 2^{-c}$ can be ignored, as the tree of all possible computations has at most 2^c leaves.

Suppose one uniformly selects a set of $O(2^{2c} \cdot c)$ possible random-tapes for the verifier. Then, with probability at least $9/10$, the fraction of random-tapes within this sample which corresponds to each node in the tree, approximates the probability associated with this node up to a $0.01 \cdot 2^{-c}$ additive term. It follows that by uniformly selecting such a sample, we can approximate the value of the tree of all possible executions (on input x), in time polynomial in the size of the tree and the size of the sample. The theorem follows. ■

References

[1] L. Babai. Trading Group Theory for Randomness. In *17th STOC*, pages 421–420, 1985.

[2] M. Bellare and E. Petrank. Making Zero-Knowledge Provers Efficient. In *24th STOC*, pages 421–420, 1992.

[3] G. Brassard, D. Chaum and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *JCSS*, pages 156–189, 1988. Preliminary version by Brassard and Crépeau in *27th FOCS*, 1986.

[4] O. Goldreich, S. Micali and A. Wigderson. Proofs that Yield Nothing but their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *JACM*, Vol. 38, No. 1, pages 691–729, 1991. Preliminary version in *27th FOCS*, 1986.

[5] S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, Vol. 18, pages 186–208, 1989. Preliminary version in *17th STOC*, 1985.

[6] J. Kilian. A Note on Efficient Zero-Knowledge Proofs and Arguments. In *24th STOC*, pages 723–732, 1992.