Introducing IP, AM, MA

Interactive Proof Systems

Florian Zuleger

March 30, 2006

Abstract

We intoduce the notion of interactive proof systems and the complexity classes IP, AM, MA, emphazing the role of randomness and interaction in these models. The concept is demonstrated by giving an interactive proof system for the graph non-isomorphism problem. We discuss issues regarding the relations between the complexity classes with respect to the number of rounds allowed. Furthermore we give an zero knowledge proof for the 3-coloring problem.

Contents

1 Introduction

A proof is a way of convincing a party of a certain claim. When talking about proofs, we consider two parties: the *proover* and the *verifier*. Given an assertion, the prover's goal is to convince the verifier of it's validity, whereas the verifier's objective is to accept only a correct assertion. In mathematics, for instance, the prover provides a fixed sequence of claims and the verifier checks that they are truthful and that they imply the theorem. In real life, however, the notion of a proof has a much wider interpretation. A proof is a process rather than a fixed object, by which the validity of the assertion is established. For instance, a job interview is a process in which the candidate tries to convince the employer that she should hire him. In order to make the right decision, the employer can adapt her questions according to the answers of the candidate, and therefore extract more information, and lead to a better decision. This example exhibits the power of a proof process rather than a fixed proof. In particular it shows the benefits of interaction between the parties.

In many contexts, finding a proof requires creativity and originality, and therefore attracts most of the attention. However, in our discussion of proof systems, we will focus on the task of the verifier - the verification process. Typically the verification procedure is considered to be relatively easy while finding the proof is a harder task. The asymmetry between the complexity of verification and finding proofs is captured by the complexity class NP.

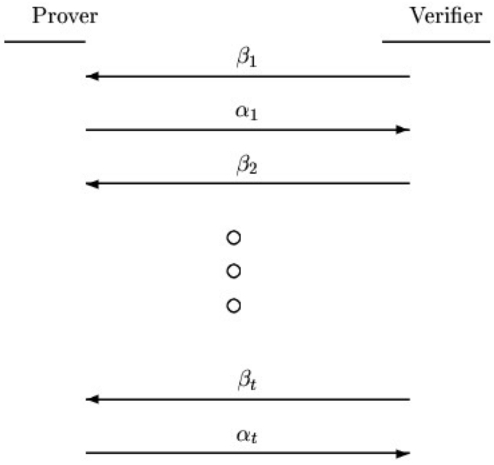
We can view NP as a proof system, where the only restriction is on the complexity of the verification procedure (the verification procedure must take at most polynomial-time). For each language $L \in$ NP there exists a polynomial-time recognizable relation R_L such that:

$$L = \{x | \exists y : s.t. (x, y) \in R_L\}$$

and $(x, y) \in R_L$ only if $|y| \leq poly(|x|)$. In a proof system for an NP language L, a proof for the claim "$x \in L$" consists of the prover sending a witness y, and the verifier checking in polynomial-time whether $(x, y) \in R_L$. Such a witness only exists if the claim is true, hence, only true assertions can be proved by this system. Note that there is no restriction on the time complexity of finding the proof (witness). A good proof system must have the following properties:

1. The verifier strategy is efficient (polynomial-time in the NP case)
2. Correctness requirements:
   - **Completeness**: For a true assertion, there is a convincing proof strategy (in the case of NP, if $x \in L$ the a witness y exists).
   - **Soundness**: For a false assertion, no convincing proof strategy exists (in the case of NP, if $x \notin L$ then no witness y exists).

In the following discussion we introduce the notion of *interactive proofs*. To do so, we generalize the requirements from a proof system, adding interaction and randomness. Roughly speaking, an interactive proof is sequence of questions and answers between the parties. The verifier asks the prover a question β_i and the prover answers with message α_i. At the end of the interaction, the verifier decides based the knowledge he acquired in the process whether the claim is true or false.



2 The Definition of IP

Following the above dicussion we define

Definition 2.1 (interactive proof systems:) *An* interactive proof system *for a language L is a two-party game between a verifier and a prover that interact on a common input in a way satisfying the following properties:*

1. *The verifier strategy is a probabilistic polynomial-time procedure (where time is measured in terms of the length of the common input)*
2. *Correctness requirements:*
   - ***Completeness**: There exists a prover strategy P, such that for every $x \in L$, when interacting on the common input x, the prover P convinces the verifier with probability at least $\frac{2}{3}$.*
   - ***Soundness**: For a false assertion, no convincing proof strategy exists (in the case of NP, if $x \notin L$ then no witness y exists).*

Definition 2.2 (The IP Hierachy:) *The complexity class IP consists of all languages having an interactive proof system.*

We call the number of message exchanges (a question and an answer) between the two parties, the number of rounds *in the system. After a certain number of rounds the verifier decides whether to accept or reject.*

For every integer function r(.), the complexity class IP(r(.)) consists of all the languages that have an interactive proof system in which, on common input x, at most $r(|x|)$ rounds are used.

For a set of integer functions R, we denote

$$IP(R) = \bigcup_{r \in R} IP(r(.))$$

2.1 Comments

- Clearly, $NP \subseteq IP$ (actually, $NP \subseteq IP(1)$). Also, $BPP = IP(0)$.
- The number of rounds in IP cannot be more than a polynomial in the length of the common input, since the verifier strategy must run in polynomial-time. Therefore, if we denote by **poly** the set of all integer polynomial functions, then $IP = IP(poly)$.

- The length of the messages exchanged cannot be more than a polynomial in the length of the common input, since the verifier cannot read or write such messages in polynomial-time
- Much like in the definition of the complexity class BPP, the probabilities $\frac{2}{3}$ and $\frac{1}{3}$ in the completeness and soundness requirements can be replaced with probabilities as extreme as $1-2^{-p(.)}$ and $2^{-p(.)}$, for any polynomial p(.). In other words the following claim holds:

  Claim 2.3 *Any language that has an interactive proof system, has one that achieves error probability of at most $2^{-p(.)}$ for any polynomial p(.).*

  Proof: We repeat the proof system sequentially for k times, and take a majority vote. Denote by z the number of accepting votes. If the assertion holds, then z is the sum of k independent Bernoulli trials with probability of success at least $\frac{2}{3}$. An error in the new protocol happens if $z < \frac{1}{2}k$.
  Using Chernoff's Bound:

  $$Pr[z < (1-\delta)E(z)] < e^{-\frac{\delta^2 E(z)}{2}}$$

  We choose $k = O(p(.))$ and $\delta = \frac{1}{4}$ and note that $E(z) = \frac{2}{3}k$ (so that $\frac{3}{4} \cdot \frac{2}{3} = \frac{1}{2}$) to get:

  $$Pr[z < (1 - \frac{1}{2}k] < 2^{-p(.)}$$

  The same argument holds for the soundness error (as due to the sequentiell nature of the interaction we can assert that in each of the k iterations, for any history of prior interactions, the success probability of any cheating strategy is bounded by 1/3). □

- Introducing both interaction and randomness in the IP class is essential:
  - By adding interaction only, the interactive proof systems collapse to NP-proof systems. Given an interactive proof system for a prover and a deterministic verifier, we construct an NP-proof system. The prover can predict the verifier's part of the interaction an send the full transcript as an NP witness. The verifier checks that the witness is a valid and accepting transcript of the original proof system.
  - By adding randomness only, we get a proof system in which the prover sends a witness and the verifier can run a BPP algorithm for checking its validity. We obtain a class (also denoted Merlin-Arthur game - MA) which seems to be a randomized (and perhaps stronger) version of NP.

2.2 Graph Non-Isomorphism(GNI)

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called *iosmorphic* (denoted $G_1 \cong G_2$) if there exists a 1-1 and onto mapping $\pi : V_1 \to V_2$ such that $(u, v) \in E_1 \Leftrightarrow (\pi(u), \pi(v)) \in E_2$. The mapping π, if existing, is called an *isomporhism* between the graphs. If no such mapping exists then the graphs are *non-isomophic* (denoted $G_1 \ncong G_2$).

GNI is the language containing all pairs of non-isomorphic graphs. Formally:

$$GNI = \{(G_1, G_2) : G_1 \ncong G_2\}$$

An interactive proof system for GNI:

- G_1 and G_2 are given as input to the verifier and the prover. Assume without loss of generality that $V_1 = V_2 = \{1, 2, \ldots, n\}$.

- The verifier chooses $i \in_R \{1,2\}$ and $\pi \in_R S_n$ (S_n is the group of all permutations on $\{1,2,\ldots,n\}$). He applies the mapping π on the graph G_i to obtain a graph H

  $$H = (\{1,2,\ldots,n\}, E_H) \text{ where } E_H = \{(\pi(u),\pi(v)) : (u,v) \in E_i\}$$

  and sends the graph H to the prover.
- The prover sends $j \in \{1,2\}$ to the verifier.
- The verifier accepts iff $j = i$.

Motivation: if the input graphs are non-isomorphic, as the prover claims, then the prover should be able to distinguish (not necessarily by an efficient algorithm) isomporhic copies of one graph from isomorphic copies of the other graph. However, if the input graphs are isomorphic, then a random isomorphic copy of one graph is distributed identically to a random isomorphic copy of the other graph and therefore, the best choice the prover could make is a random one. This fact enables the verifier to dinstinguish between the two cases. Formally:

Claim 2.4 *The above protocol is an interactive proof system for GNI.*

Proof: We have to show that the above protocol is an interactive proof system satisfies the two properties in the definition of interactive proof systems:

- The verifier's strategy can be easily implemented in probabilistic polynomial time. (The prover's complexity is unbounded and indeed, he has to check isomorphism between two graphs, a problem not known to solved in probabilistic polynomial time.)
- - Completeness: In case $G_1 \ncong G_2$, every graph can be isomorphic to at most one of G_1 or G_2. It follows that the prover can always send the correct j (i.e. a j such $j = i$).
  - Soundness: In case $G_1 \cong G_2$ we show that the prover convinces the verifier with probability $\frac{1}{2}$ (the probability ranges over all the possible coin tosses of the verifier, i.e. the choice of i and π). Denote by H the graph sent by the verifier. $G_1 \cong G_2$ implies that H is isomorphic to both G_1 and G_2. For $k = 1, 2$ let

    $$S_{G_k} = \{\sigma \in S_n | \sigma G_k = H\}$$

    This means that when choosing $i = k$, the verifier can obtain H only by choosing $\pi \in S_{G_k}$.
    Assume $\tau \in S_n$ is an isomorphism between G_1 and G_2, i.e. $G_1 = \tau G_2$. For every $\sigma \in S_{G_1}$ it follows that $\sigma\tau \in S_{G_2}$ (because $\sigma\tau G_2 = \sigma G_1 = H$). Therefore, τ is a 1-1 mapping from S_{G_1} to S_{G_2} (since S_n is a group). Similary, τ^{-1} is a 1-1 mapping from S_{G_2} to S_{G_1}. Combining the two arguments we get $|S_{G_1}| = |S_{G_2}|$. Therefore, given that H was sent, the probability that the verifier chose $i = 1$ is equal to the probability of the choise $i = 2$. It follows that for every decision the prover makes he has success probabiliy $\frac{1}{2}$.

If we repeated the above protocol twice we get the required probabilities (like in the amplification argument). □

Corollary 2.5 *$GNI \in IP(2)$.*

Remark: ISOMORPHISM is not known to be in P, but of course it is in NP (guessing the right permutation and then checking the isomorphism in polynomial time), whereas GNI is not known to be in NP.

Remark: We state that the secrecy of the outcome of the coin tosses is essential to this protocol.

3 Public-Coins Systems and the Number of Rounds

Definition 3.1 (public-coin interactive proofs - AM:) *Public coin proof systems (known also as Arthur-Merlin games) are a special case of interactive proof systems, in which, at each round,* ***the verifier can only toss coins, and send their outcome to the prover****. After a certain number of rounds the verifier decides* ***deterministically*** *whether to accept or reject.*

For every integer function r(.), the complexity class AM(r(.)) consists of all the languages that have an Arthur-Merlin proof system in which, on common input x, at most $r(|x|)$ *rounds are used.*

Denote $AM = AM(1)$*.*

We note that the definition of AM as Arthur-Merlin games with one round is inconsistent with the notation IP = IP(poly).

The difference between the Arthur-Merlin games and the general interactive proof systems can be viewed as the difference between asking tricky questions, versus asking random questions. Surprisingly it was shown that these versions are essentially equivalent:

Theorem 3.2 (Relating IP(.) to AM(.))*:*

$$\forall r(.) : IP(r(.)) \subseteq AM(r(.) + 1)$$

The following theorem shows that power of AM(r(.)) is invariant under a linear change in the number of rounds:

Theorem 3.3 (Linear Speed-up Theorem)*:*

$$\forall r(.) \geq 2 : AM(2r(.)) = AM(r(.))$$

The above theorems are quoted without proof. Combing them we get:

Corollary 3.4

$$\forall r(.) \geq 2 : IP(2r(.)) = IP(r(.))$$

Corollary 3.5 (Collapse of constant-round IP to one-round AM)*:*

$$IP(O(1)) = AM(1)$$

Corollary 3.6 (Relating MA to AM)

$$MA \subseteq AM$$

Theorem 3.7 (Relating MA to PP)*:*

$$MA \subseteq PP$$

Proof: Let $L \in MA$. Thus there are a polynomial p and a polynomial-time Turing machine Q such that:

$$x \in L \Rightarrow \exists s \in \{0,1\}^{p(|x|)} : Pr[Q(x,r,x)] > \frac{2}{3}$$

$$x \notin L \Rightarrow \forall s \in \{0,1\}^{p(|x|)} : Pr[Q(x,r,x)] < \frac{1}{3}$$

where probability is taken over uniform distribution in $\{0,1\}^{p(|x|)}$.

Using standard amplification we can construct a new polynomial p_1 and a new polynomial-time machine Q_1 such that

$$x \in L \Rightarrow \exists s \in \{0,1\}^{p(|x|)} : Pr[Q_1(x,r,s)] > 1 - 4^{-p(|x|)}$$

$$x \notin L \Rightarrow \forall s \in \{0,1\}^{p(|x|)} : Pr[Q_1(x,r,s)] < 4^{-p(|x|)}$$

where probability is taken over uniform distribution in $\{0,1\}^{p_1(|x|)}$.

Consider now the uniform distribution on pairs $< r, s > \in \{0,1\}^{p(|x|)+p_1(|x|)}$. We have

$$x \in L \Rightarrow \exists Pr[Q_1(x,r,s)] > 2^{-p(|x|)}(1 - 4^{-p(|x|)}) > 4^{-p(|x|)}$$

$$x \notin L \Rightarrow Pr[Q_1(x,r,s)] < 4^{-p(|x|)}$$

This is equivalent to $L \in PP$. □

4 Perfect Completeness and Soundness

In the definition of interactive proof systems we require the existence of a prover strategy that for $x \in L$ convinces the verifier with probability at least $\frac{2}{3}$ (analogous to the definition of the complexity class BPP). One can consider a definition requiring *perfect completeness*; i.e., convincing the verifier with probability 1. We will now show that the definitions are equivalent.

Theorem 4.1 *If a language has an interactive proof system then it has one with perfect completeness.*

We will show that given a public coin proof system we can construct a perfect completeness public coin proof system.

We use the fact that public coin proof systems and interactive proof system are equivalent, so if L has an interactive proof system it also has a public coin system. We define:

$$AM^0(r(.)) = \{L | \ L \text{ has a perfect completeness } r(.) \text{ round public coin proof system}\}$$

So given an interactive proof system we create a public coin proof system and using the following lemma convert it to one with perfect completeness. Thus, the above theorem which refers to arbitrary interactive proofs follows from the following lemma which refers only to public-coin interactive proofs.

Lemma 4.2 *If L has a public coin proof system then it has one with perfect completeness*

$$AM(r(.)) \subseteq AM^0(r(.)+1)$$

Proof: Given an Arthur-Merlin proof system, we construct an Arthur-Merlin proof system with perfect completeness and one more round.

Assume, without loss of generality, that the Arthur-Merlin proof system consists of t rounds, an that Arthur sends the same number of coins m in each round (otherwise, ignore the redundant coins). Also assume that the completeness and soundness error probabilities of the proof system are at most $\frac{1}{3tm}$. This is obtained using standard amplification.

We denote the messages sent by Arthur (the verifier) $r_1, \ldots, r_t$ and the messages sent by Merlin (the prover) $\alpha_1, \ldots, \alpha_t$. Denote by $< P, V >_x (r_1, \ldots, r_t)$ the outcome of the game on common input x between the optimal prover P and the verifier V in which the verifier uses coins $r_1, \ldots, r_t$: $< P, V >_x (r_1, \ldots, r_t) = 0$ if the verifier rejects and $< P, V >_x (r_1, \ldots, r_t) = 1$ otherwise.

We construct a new proof system with perfect completeness, in which Arthur and Merlin play tm games simultaneously. Each game is like the original game except that the random coins are shifted by a fixed amount. The tm shifts (one for each game) are sent by Merlin in an additional at the beginning. Arthur accepts if at least one of the games is accepting. Formally, we add an additional round at the beginning in which Merlin sends the shifts $S^1, \ldots, S^{tm}$ where $S^i = (S^i_1, \ldots, S^i_t), S^i_j \in \{0,1\}^m$ for every i between 1 and tm. For game i and round j, Merlin considers the random coins to be $r_j \oplus S^i_j$ and sends as a message α^i_j where α^i_j is computed according to $(r_1 \oplus S^i_1, \ldots, r_t \oplus S^i_t)$. The entire message in round j is $\alpha^1_j, \ldots, \alpha^{tm}_j$. At the end of the protocol Arthur accepts if at least one out of the tm games is accepting.

In order to show perfect completeness we will show that for every $x \in L$ there exists $S^1, \ldots, S^{tm}$ such that for all $r_1, \ldots, r_t$ at least one of the games is accepting. We use a probabilistic argument to show that the complementary event occurs with probability strictly smaller than 1.

$$Pr_{S^1,\ldots,S^{tm}}[\exists r_1, \ldots, r_t \bigwedge_{i=1}^{tm} (< P, V >_x (r_1 \oplus S^i_1, \ldots, r_t \oplus S^i_t) = 0)]$$

$$\leq_{(1)} \sum_{r_1,\ldots,r_t \in \{0,1\}^m} Pr_{S^1,\ldots,S^{tm}}[\bigwedge_{i=1}^{tm} (< P, V >_x (r_1 \oplus S^i_1, \ldots, r_t \oplus S^i_t) = 0)]$$

$$\leq_{(2)} 2^{tm} \cdot (\frac{1}{3tm})^{tm} < 1$$

Inequality (1) is obtained using the union bound. Inequality (2) is due to the fact that the $r_j \oplus S^i_j$ are independent random variables so the results of the games are independent, and that the optimal prover fails to convince the verifier on a true assertion with probability at most $\frac{1}{3tm}$.

We still have to show that the proof system suggested satisfies the soundness requirement. We show that for every $x \notin L$ and for any prover strategy $P^\star$ and choices of shifts $S^1, \ldots, S^{tm}$ the probability that one or more of the tm games is accepting is at most $\frac{1}{3}$.

$$Pr_{r_1,\ldots,r_t}[\bigvee_{i=1}^{tm} (< P, V >_x (r_1 \oplus S^i_1, \ldots, r_t \oplus S^i_t) = 1)]$$

$$\leq_{(1)} \sum_{i=1}^{tm} Pr_{r_1,\ldots,r_t}[< P^\star, V >_x (r_1 \oplus S^i_1, \ldots, r_t \oplus S^i_t) = 1)]$$

$$\leq_{(2)} \sum_{i=1}^{tm} \frac{1}{3tm} = \frac{1}{3}$$

Inequality (1) is obtained using the union bound. Inequality (2) is due to the fact that any prover has probability of at most $\frac{1}{3tm}$ of success for a single game. □

Unlike the last theorem, requiring *perfect soundness* (i.e. for every $x \notin L$ and every prover strategy $P^\star$, the verifier always rejects after interacting with $P^\star$ on common input x) reduces the model to an NP-proof system, as seen in the following proposition:

Proposition 4.3 *If a lanuage L has an interactive proof system with perfect soundness then $L \in NP$.*

Proof: Given an interactive proof system with perfect soundness we construct an NP-proof system. In case $x \in L$, by the completeness requirement, there exists an accepting transcript. The prover finds an outcome of the verifiers's coin tosses that gives such a transcript and sends the full transcript along with the coin tosses. The verifier checks in polynomial time that the transcript is valid and accepting and if so - accepts. This serves as an NP-witness to the fact that $x \in L$. If $x \notin L$ then due to the perfect soundness requirement, no outcome of verifier's coin tosses yields an accepting transcript and therefore there are no NP-witnesses. □

Remark: This is an alternative argument for interactive proof systems collapsing to NP without randomness. This is due to the fact that perfect soundness is equivalent to a deterministic verifier.

5 A zero knowledge proof for the 3-COLORING problem

We shall conclude this paper with a very interesting protocol that uses interactive proofs and cryptography.

Suppose that Alice is a girl with superintellectual abilities capable to solve NP-problems and that Bob - an ordinary guy, but a good friend - is only able to compute problems in P. So Alice can color the nodes of a large graph $G = (V, E)$ with three colors, such that no two adjacent nodes have the same color. Since 3-COLORING is an NP-complete problem, Alice is very proud an excited, and wants to convince Bob that she has a coloring of G. There is nothing difficult here: Since 3-COLORING is in NP, she can simply send her 3-coloring to Bob. But Alice is worried that, if Bob finds out from her ho to color G, he can announce it the same way to his friends without appropriate credit to Alice's ingenuity. What is required here is a *zero knowledge proof*, that is, an interactive protocol at the end of which Bob is convinced that with very high probability Alice has a legal 3-coloring of G, but has no clue about the actual 3-coloring.

Here is a protocol that can achieve this seemingly impossible task. Suppose that Alice's coloring is $\chi : V \mapsto \{00, 11, 01\}$, that is, the three colors are these three strings of length two. The protocol proceeds in rounds. At each round, Alice carries out the following steps: First she generates a random permutation π of the three colors. Then she generates $|V|$ RSA public-private key pairs, (p_i, q_i, d_i, e_i), one for each node $i \in V$. For each node i she computes the probabilistic encoding (y_i, y_i'), according the $j$the RSA system, of the color $\pi(\chi(i))$ - the color of i permuted unter π. Supppose that $b_i b_i'$ are the two bits of $\pi(\chi(i))$; then $y_i = (2x_i + b_i)^{e_i} mod p_i q_i$ and $y_i' = (2x_i' + b_i')^{e_i} mod p_i q_i$, where x_i and x_i' are random integers no greater than $\frac{pq}{2}$. All these computations are private to Alice. Alice reveals to Bob the integers $(e_i, p_i q_i, y_i, y_i')$ for each node $i \in V$. That is, the public part of the RSA systems, and the encrypted colors.

It is now Bob's turn to move. Bob picks at random an edge $[i, j] \in E$, and inquires whether its endpoints have a different color, as they should. Alice then reveals to Bob the secret keys d_i and d_j of the endpoints, allowing Bob to compute $b_i = y_i^{e_i} mod 2$, and similarly for b_i',b_j and b_j', and check that, indeed $b_i b_i' \neq b_j b_j'$. This concludes the description of a round. Alice and Bob repeat $k|E|$ times, where k is a parameter representing the desired reliability of the protocol.

Obviously, if Alice has a legal coloring of G, all inquiries of Bob will be satisfied. But what if she does not? If she has no legal coloring, then necessarily at each round there is an edge $[i, j] \in E$ such that $\chi(i) = \chi(j)$. At each round Bob has a probability of at least $\frac{1}{|E|}$ if discovering that edge. After $k|E|$ rounds, the probability of Bob finding out that Alice has no legal coloring is at least $1 - e^{-k}$.

What is remarkable about this protocol is that Bob has learned nothing about Alice's coloring of G in the process. This can be argued along these lines: Suppose that Alice does have a legal 3-coloring, and the protocol is carried out. What does Bob see at each round, after all? Some randomly generated public keys, some probabilistic encryptions of colors.

Then he proposes an edge, he sees two decryption keys, and finally he finds out the two colors $\chi(u)$ and $\chi(v)$. But these colors are permuted versions of the original colors of Alice, and so they are nothing else but a randomly chosen pair of different colors. In conclusion, Bob sees nothing that he could not generate sitting by himself, fipping a fair coin for polynomial time, without Alice and her 3-coloring. We can conclude that zero knowledge was exchanged - in fact, a reasonable definition of zero knowledge goes roughly along these lines, namely that the interactions in the protocol form a random string drawn from a destribution that was available at the beginning of the protocol.

As a final note, it is handy that the zero knowledge protocol just described works for 3-COLORING, an NP-complete problem. Using reductions, it is possible to conclude all problems in NP have zero-knowledge proofs.

References

[1] O. Goldreich, Introduction to Complexity Theory - Lecture Notes, 1999.

[2] C. Papadimitriou, Computational Complexity, Addison Wesley, 1994.

[3] L. Babai, Trading group theory for randomness, In Proc. of th1 17th ACM Symposium on Theory of Computing, 1985.

[4] S. Goldwasser, M. Sipser. Private Coins versus Public Coins in Interactive Proof Systems, Advances in Computing Research: a research annual, Vol. 5 (Randomness and Computation, S. Micali, ed.), pages 73-90, 1989. Extended abstract in 18th STOC, pages 69-68, 1986.

[5] N. K. Vereshchagin. On the power of PP. In Proceedings 7th Structure in Complexity Theory, pages 138–143. IEEE Computer Society Press, 1992.