Chapter 12. RANDOMIZED PROOFS

In this chapter several types of randomized proofs are introduced and their power is analyzed.

WHAT IS A PROOF?

- A proof is whatever convinces me (M. Even).
- A nice proof makes us wiser (Yu. Manin).
- A proof is a sequence of statements each of them is either an axiom or follows from previous statements by am easy deduction rule - whether a to-be-proof is indeed a proof it should be cheackable by a computer. (A proof is therefore a computation process - a formalists' (Hilbert) view.)
- A proof of the existence of an object is a real proof only in case if the proof contains a method how such an object construct (a intuitionists' view).
- The question of *What is a proof* is one of major ones of the philosophy of science and mathematics.

FROM THE HISTORY of PROOFS

- The concept of the proof (of a theorem from axioms) was introduced during the first golden era of mathematics, in Greece, 600-300 BC.
- Most of their proofs were actually proofs of correctness of geometric algorithms.
- After 300 BC, Greek's ideas concerning proofs were actually ignored for 2000 years.
- During the second golden era of mathematics, in 17th century, the concept of the proof did not play very important role. Famous was encouragement of those times "Go on, God will be with you" whenever rigour of some methods or correctness of some theorem was questioned.
- An understanding that proofs are important has developed again at the end of 19th century and especially at the beginning of 20th century because
  - a lot of counter-intuitive phenomena have appeared in mathematics (for example a function that is everywhere continuous but has nowhere derivative);
  - paradoxes have appeared in the set theory. - For example, Does there exist a set of all sets?

STORY of GREEK MATHEMATICS

GREEK MATHEMATICS

- The Greek mathematics can be seen as dealing to a large extent with geometrical calculi.
- The goals of the proofs of theorems were actually aimed to show correctness of algorithms.
- One can say about that period that knowledge was mainly of practical nature, calculations were of chief interest. When some "theoretical" elements entered they were to a large extend (though not only) to facilitate techniques.
- Much of mathematics developed can be seen also as important attempts to understand the concept of a "process".
- In addition, three main problems of antique:
  - squaring of the circle;
  - duplication of the cube;
  - trisection of the angle.

  which had profound impact on the development of science, and led to the development of infinitesimal calculus, are actually algorithmic problems, unsolvability of which, with compass and straightedge alone, has been shown only in modern time.

IMPORTANT PARADOXES

Also more theoretical approaches of that period can be seen as being often deeply informatical in nature. For example, the attempts to understand the concept of a *process* and to deal with four famous Zeno paradoxes:

- Dichotomy paradox;
- Achilles paradox;
- Arrow paradox;
- Stade paradox

GREEK MATHEMATICS - CLASSICAL PERIOD (Pythagoreans, 600-300

- Greek mathematics was based on and helped to develop a new doctrine of nature - namely that nature is orderly and develops according to a plan. Old doctrine, but in Greek society actually dominating at that time, was that gods manipulate nature and men according their whims.
- Its protagonists were Thales and Pythagorean and it was highlighted by works of Eudoxus, Euclid, Plato and Aristotle.
- Greeks created, for the first time, mathematics as an organized, independent and reasoned discipline.
- Greeks made mathematics abstract - to see mathematical entities, numbers and geometrical objects as abstractions, sharply distinguished from physical objects.
- Greek made mathematics deductive, deriving truth in theorems by deduction from axioms.
- Greeks came with the idea to prove existence by construction.

- Their mathematics was mainly (well founded) geometry, (actually motivated by astronomy).
- Main goal of Mathematics was seen as to understand functioning of universe - they believed that mathematics is the key to comprehension of universe, for mathematical laws are the essence of its design.
- Greeks made mathematics to be a liberal art closely related (and a preparation) to philosophy.
- For Greeks arithmetic, geometry and astronomy were considered as *the art of the mind and music for the soul.*
- It is believed that it was the aesthetic appeal of the subject that caused Greek mathematicians to carry the exploration of particular topics beyond their use in the understanding of the physical world.
- Greeks made enormous contributions to the philosophy of science.
- Their mathematics was much inspired by the fact that phenomena that are much diverse from qualitative point of view exhibit identical mathematical properties.

- Their position was based on a belief that mind is capable to recognize truth, observation of physical world is not needed. As a consequences their outcomes were a combination of ingenious ideas, bold speculations and shrewd guesses.
- Greek mathematicians mixed deep and serious thoughts with what we could consider as fanciful, useless, and unscientific doctrines.
- Greek mathematicians "reduced" astronomy and music to numbers and therefore astronomy and music was considered to be a part of mathematics.
- Their classics (books) contained only formal deductive mathematics, no motivation - though one can expect astronomy was the main motivation.
- They believed in the power of mind to yield also the first principles.
- During the classical period, the doctrine of the mathematical design of nature was established and the search for its mathematical laws instituted.
- They believed that mathematical facts are not created by men, that they exists and can only be discovered.
- Their main contributions were practically forgotten or ignored for 2000 years

LIMITATIONS OF CLASSICAL GREEK MATHEMATICS

- They reduced mathematics to geometry dealing with simple curves, areas and bodies only.
- By insisting on a unity, completeness and simplicity, and by separating speculative thoughts from utility, they narrowed people's vision and closed their minds to new thoughts and methods.
- Their insistence on exact concepts and proofs was also a defect so far as creative mathematics is concerned.
- They were not able to accept irrational numbers in arithmetic.
- Their concept of proof was too restrictive concerning creative mathematics, and so was their concept of constructability.
- They were not able to accept infinity. Neither infinity of large not of small objects and not infinite processes.
- They could not accept continuity because of their emphasis on atomism.

WHY WAS GREEK MATHEMATICS IGNORED FOR 2000 YEARS?

- One of the most puzzling things in history of science is why ingenious Greek mathematics of its classical period was later practically ignored for about 2000 years.
- It was actually already ignored in the Alexandrian period (300 B.C. - 600 A.C.)
- One reason is nicely put together by Cicero: *The Greeks held the geometers in the highest honour; accordingly, nothing made more brilliant progress among them than mathematics. But we have established as the limit of this art its usefulness in measuring and counting.*
- Other reasons: it ignored computational needs of society; it was based on a wrong view of importance of observations and so it could hardly help other sciences; concerning exactness and deduction, it made too high and restrictive requirements for that period.
- Christianity decreased interest in physical world, preparation of the soul for after-life in the heaven was the main concern.

- Christianity brought a new belief concerning ways one seek for truth.
- Theology was seen as embracing all knowledge.
- New revival of the Classical Greek period appeared after a new doctrine was developed that saw God as the one creating mathematical nature and as seeing search for mathematical laws of nature as religious quest. A discovery of a mathematical law was seen as a further discovery of the greatness of the God - and therefore God was to be praised after each discovery of a simple law of nature, not the one who made the discovery. For example, Kepler wrote paeans to God after each of his discovery.

PLATO versus ARISTOTLE

- Concerning the development of the basic philosophical positions of main Greek period two men played the main role: Plato and Aristotle.
- Their views were quite different, and opposite in many sense, what later influenced much development of the science during the Renaissance.
- Plato was the most influential propagator of the doctrine that the reality and intelligibility of the physical world can be comprehended only through mathematics.
- He was convinced that the world was mathematically designed.
- Plato believed that physical world is but an imperfect copy of the ideal world, the one mathematicians and philosophers should study.
- He believed that mathematical laws, eternal and unchanging, are the essence of reality. Plato not only tried to understand nature through mathematics, he actually tried to substitute mathematics for nature itself.

- Aristotle believed in material things as the primary substance and source of reality.
- He believed that science must study the physical world to obtain truth.
- He believed that science must study the physical world to obtain truth.
- He believed that genuine knowledge is obtained from sense experience by intuition and abstraction.
- He distinguished sharply between physics and mathematics and assigned a minor role to mathematics.

PROOFS for NP-problems

Definition A language $L \subset \Sigma^*$ is in **NP** if and only if there exists a polynomial-bounded function p and a polynomial time deterministic Turing machine M with the following properties:

- For every $x \in L$, it holds that M accepts (x, y) for some string $y \in \Sigma^{p(|x|)}$ (called certificate or witness or proof);
- For every $x \notin L$, it holds that M rejects (x, y) for all strings $y \in \Sigma^{p(|x|)}$.

Quantum proofs

- A quantum proof is a quantum state that plays the role of a witness or certificate to a quantum computer that runs a verification procedure.
- All languages in **NP** have very short (logarithmic size) quantum proofs which can be verified provided that two unentangled copies are given.

Classes MA and AM

Definition A language $L \subset \Sigma^*$ is in **MA** if and only if there exists a polynomial-bounded function p and a polynomial time probabilistic Turing machine M with the following properties:

- For every $x \in L$, it holds that $\text{Prob}[M \text{ accepts } (x, y)] \geq \frac{2}{3}$ for some string $y \in \Sigma^{p(|x|)}$ (called a certificate or a witness or a proof of x);
- For every $x \notin L$, it holds that $\text{Prob}[M \text{ accepts } (x, y)] \leq \frac{1}{3}$ for all strings $y \in \Sigma^{p(|x|)}$.

Definition A language $L \subset \Sigma^*$ is in **AM** if and only if there exist a polynomial-bounded functions p and q and a polynomial time probabilistic Turing machine M with the following properties:

- For every $x \in L$ and at least $2/3$ of all strings $y \in \Sigma^{p(|x|)}$, there exists a string $z \in \Sigma^{q(|x|)}$ such that M accepts (x, y, z);
- For every $x \notin L$ and at least $2/3$ of all $y \in \Sigma^{p(|x|)}$, there are no $z \in \Sigma^{q(|x|)}$ such that M accepts (x, y, z).

INTERACTIVE PROOF PROTOCOLS

In an interactive proof system there are two parties:

- An (all powerful) Prover, often called Peggy (actually a randomized algorithm that uses a private random number generator);
- A not too much (polynomially) powerful Verifier, often called Vic (a polynomial time randomized algorithm using a private random number generator).

The prover knows some secret, or a knowledge, or a fact about a specific object, and wishes to convince the Verifier, through a communication with him, that he has the above knowledge.

For example, both Prover and Verifier posses an input x and Prover wants to convince Verifier that x has a certain properties and that (s)he – Prover – knows how to prove that.

The interactive prove consists of several rounds. In each round Prover and Verifier alternatively do the following.

1. Receive a message from the other party.
2. Perform a (private) computation.
3. Send a message to the other party.

Communication starts usually by a challenge from the Verifier and a response by the Prover.

At the end, the Verifier either accepts or rejects Prover's attempts to convince him. An interactive proof protocol is said to be an interactive proof system for a decision problem Π if the following properties are satisfied.

Completeness : If x is a yes-instance of Π, then the Verifier always accepts Prover's "proof".

Soundness : If x is a no-instance of Π, then the Verifier accepts Prover's "proof" only with a very small probability.

ZERO-KNOWLEDGE PROOFS - INFORMALLY

Very informally An interactive “proof” protocol at which a Prover tries to convince a Verifier about the truth of a statement, or about possession of a knowledge, is called “zero-knowledge” protocol if the Verifier does not learn from the communication with the Prover anything more except that the statement is true or that Prover has knowledge (secret) she claims to have.

Example The proof $n = 670592745 = 12345 \times 54321$is not a zero-knowledge proof that n is not a prime.

Informally: A zero-knowledge proof is an interactive proof protocol that provides **highly convincing evidence** that a statement is true or that Prover has certain knowledge (of a secret) and that the Prover knows a (standard) proof of it while providing **not a single bit of information** about the proof (knowledge or secret). (In particular, Verifier who got convinced about the correctness of a statement cannot convince the third person about that.)

More formally: A zero-knowledge proof of a theorem T is an interactive two party protocol, in which the Prover is able to convince the Verifier who follows the same protocol, by an overwhelming statistical evidence,

that T is true, if T is indeed true,

but no Prover is not able to convince Verifier that $T$is true, if this is not so.

In additions, during their interactions, the Prover does not reveal to the Verifier any other information, except whether T is true or not.

Consequently, whatever Verifier can do after he gets convinced, he can do just believing that T is true.

Age difference finding protocol

Alice and Bob wants to find out who is older without disclosing any other information about their age.

The following protocol is based on a public-key cryptosystem, in which it is assumed that neither Bob nor Alice are older than 100 years.

Protocol Age of Bob: j, age of Alice: i.
1. Bob choose a random x, computes $k = e_A(x)$ and sends Alice $s = k - j$.
2. Alice first computes the numbers $y_u = d_A(s+u); 1 \leq u \leq 100$, then chooses a large random prime p and computes numbers

$$z_u = y_u \bmod p, \quad 1 \leq u \leq 100(\star)$$

and verifies that for all $u \neq v$

$$|z_u - z_v| \geq 2 \text{ and } z_u \neq 0.(\star\star)$$

(If this it not the case, Alice choose a new p, repeats computations in $(\star)$ and checks $(\star\star)$ again.) Finally, Alice sends Bob the following sequence (order is important).

$$z_1, \dots, z_i, z_{i+1}+1, \dots, z_{100}+1, p$$

$z'_1, \dots, z'_i, z'_{i+1}, \dots, z'_{100}$
3. Bob checks whether j-th number in the above sequence is congruent to x modulo p. If yes, Bob knows that $i \geq j$, otherwise $i < j$.

Zero-knowledge proof for quadratic residua

Input: An integer $n = pq$, where p, q are primes and $x \in QR(n)$.

Protocol: Repeat $\lg n$ times the following steps:

1. Peggy chooses a random $v \in \mathbf{Z}_n^*$ and sends to Vic
$$y = v^2 \bmod n.$$
2. Vic sends to Peggy a random $i \in \{0, 1\}$.
3. Peggy computes a square root u of x and sends to Vic
$$z = u^i v \bmod n.$$
4. Vic checks whether
$$z^2 \equiv x^i y \pmod{n}.$$

Vic accepts Peggy's proof if he succeeds in Step 4 in each of $\lg n$ rounds.

Completeness is straightforward:
Soundness. If x is not a quadratic residue, then Peggy can answer only one of two possible challenges (only if $i = 0$), because in such a case y is a quadratic residue if and only if xy is not a quadratic residue. This means that Peggy will be caught in any given round of the protocol with probability $\frac{1}{2}$.

The overall probability that prover deceives Vic is therefore $2^{-\lg n} = \frac{1}{n}$.

Zero-knowledge proof for graph isomorphism

Input: Two graphs G_1 and G_2 with the set of nodes $\{1, \ldots, n\}$.
Repeat the following steps n times:

1. Peggy chooses a random permutation π of $\{1, \ldots, n\}$ and computes H to be the image of G_1 under the permutation π, and sends H to Vic.
2. Vic chooses randomly $i \in \{1, 2\}$ and sends it to Peggy. $\{$*This way Vic asks for isomorphism between* H *and* G_i.$\}$
3. Peggy creates a permutation ρ of $\{1, \ldots, n\}$ such that ρ specifies isomorphism between H and G_i and Peggy sends ρ to Vic.

   $\{$*If* $i = 1$ *Peggy takes* $\rho = \pi$; *if* $i = 2$ *Peggy takes* $\rho = \sigma \circ \pi$, *where* σ *is a fixed isomorphic mapping of nodes of* G_2 *to* G_1.$\}$
4. Vic checks whether H provides the isomorphism between G_i and H.

Vic accepts Peggy's "proof" if H is the image of G_i in each of the n rounds.

Completeness. It is obvious that if G_1 and G_2 are isomorphic then Vic accepts with probability 1.

Soundness: If graphs G_1 and G_2 are not isomorphic, then Peggy can deceive Vic only if she is able to guess in each round the i Vic chooses and then sends as H the graph G_i. However, the probability that this happens is 2^{-n}.

Observe that Vic can perform all computations in polynomial time.
However, why is this proof a zero-knowledge proof?

Why is the last "proof" a "zero-knowledge proof"?

Because Vic gets convinced, by the overwhelming statistical evidence, that graphs G_1 and G_2 are isomorphic, but he does not get any information ("knowledge") that would help him to create isomorphism between G_1 and G_2.

In each round of the proof Vic see isomorphism between H (a random isomorphic copy of G_1) and G_1 or G_2, (but not between both of them)!

However, Vic can create such random copies H of the graphs by himself and therefore it seems very unlikely that this can help Vic to find an isomorphism between G_1 and G_2.

Information that Vic can receive during the protocol, called *transcript*, contains:

- The graphs G_1 and G_2.
- All messages transmitted during communications by Peggy and Vic.
- Random numbers used by Peggy and Vic to generate their outputs.

Transcript has therefore the form

$$T = ((G_1, G_2); (H_1, i_1, \rho_1), \ldots, (H_n, i_n, \rho_n)).$$

The essential point, which is the basis for the formal definition of zero-knowledge proof, is that Vic can forge transcript, without participating in the interactive proof, that look like "real transcripts", if graphs are isomorphic, by means of the following forging algorithm called **simulator**.

GRAPH NON-ISOMORPHISM

A simple interactive proof protocol exists for computationally very hard graph non-isomorphism problem.

Input: Two graphs G_1 and G_2, with the set of nodes $\{1, \ldots, n\}$.

Protocol: Repeat n times the following steps:

1. Vic chooses randomly an integer $i \in \{1, 2\}$ and a permutation π of $\{1, \ldots, n\}$. Vic then computes the image H of G_i under the permutation π and sends H to Peggy.
2. Peggy determines a value j such that G_j is isomorphic to H, and sends j to Vic.
3. Vic checks if $i = j$.

Vic accepts Peggy's proof if $i = j$ in each of n rounds.

Completeness: If G_1 is not isomorphic to G_2, then the probability that Vic accepts is clearly 1.

Soundness: If G_1 is isomorphic to G_2, then Peggy can deceive Vic if and only if she correctly guesses n times the i Vic chosen randomly. Probability that this happens is 2^{-n}.

Observe that Vic's computations can be performed in polynomial time (with respect to the size of graphs).

ZERO-KNOWLEDGE PROOFS for NP-COMPLETE PROBLEMS

In 1986 Goldreich, Micali and Widgerson showed that if one-way functions exist, then zero-knowledge proofs exist for each **NP**-complete problem.

Since all **NP**-complete problems are reducible to each other, to prove the above statement it is sufficient to show the existence of zero-knowledge proof for one of them, for example for 3-coloring of graphs.

3-COLORING of GRAPHS

Let the Prover know a 3-coloring of a graph G. He can convince about it a Verifier if they perform n rounds of the following protocol.

The prover makes a 3-coloring of G, then permutes colors, encrypts each node-color using a special one-way function, permutes nodes and then sends to the Verifier the resulting graph with all nodes labeled by cryptotexts of their colors.

The Verifier chooses an edge and asks the Prover to disclose the corresponding one-way functions and colours of the edge's end-nodes. If the Verifier finds that two chosen nodes are indeed colored by different colors, the prover succeeded in that round.

In case the Prover succeeds in all n rounds the Verifier accepts as the fact that the Prover knows how to 3-color G. At the same time, the verifier got the slightest idea how to 3-color G.

Classes CZK and SZK

Zero-knowledge proofs of the graph non-isomorphism and of the graph 3-coloring are quite different.

In case of 3-coloring of the graph, the fact that the proof is zero-knowledge depends in the crucial way on the fact, a computational assumption, that one-way functions exist and therefore the polynomial time verifier does not have enough computational power to do encryptions of colors. Such zero-knowledge proofs are called computational zero-knowledge proofs and the class of problems with computational zero-knowledge proofs is denoted **CZK**.

In case of graph-non-isomorphism problem, the verifier cannot cheat no matter how much computational power he has. Such zero-knowledge proofs are called statistical zero-knowledge proofs and the class of the problems with such proofs is denoted **SZK**.

Clearly **SZK** $\subseteq$ **CZK**.

OPEN PROBLEM are the classes **CZK** and **SZK** equal?

It can be shown that if one-way functions exist, then **CZK=PSPACE**.

PROBABILISTICALLY CHECKEABLE PROOFS

The concept of probabilistic checkeable proofs (PCP), or *transparent* or *holographic* proof, is another great/shocking idea concerning proofs.

Informally, PCP proofs are proofs such that are written down in such a way that one needs to look only to (very) few randomly chosen bits of it in order to find out whether the proof is correct with (very) probability.

The hard task is to encode a given proof so randomized checking is possible.

Famous **PCP-Theorem** says that every **NP**-complete problem/language admits a probabilistically checkeable polynomially long proof.

This implies that every mathematical proof can be encoded in such a way that any error in the original proof translates into errors almost everywhere in the new proof.

PCP-THEOREM - STILL INFORMALLY

Intuitively, the PCP-theorem says that for some fixed (and universal) constant K, for every n, any mathematical proof of length n can be rewritten as a (different) proof of length $poly(n)$ that is formally verifiable on 99% by a randomized algorithm that makes only k queries to the proof.

One can also prove that each proof can be rewritten in such a way that it is enough to check 11 randomly chosen bit in order to verify the proof with probability at least $\frac{1}{2}$.

PCP-THEOREM - FORMALLY

Let PCP$[f, g]$ denote the class of languages (decision problems) with a transparent proof that uses $\mathcal{O}(f(n))$ random bits and checks $\mathcal{O}(g(n))$ bits of an n bit long proof.

It holds:

PCP Theorem $\mathbf{NP} = PCP[\lg n, \mathcal{O}(1)]$.

This result says that no matter how large an instance of an **NP**-problem is and how long its proof is, it is enough to look to a fixed number of (randomly) chosen bits of the proof in order to determine, with high probability, its validity.

Moreover, given an ordinary proof of membership for an **NP**-language, the corresponding transparent proof can be constructed in polynomial time in the length of the original classical proof.

Transparent proofs therefore have strong error-correcting properties.

PCP proof for the GRAPH NON-ISOMORPHISM

Given any two n-node non-isomorphic graphs G_0 and G_1 the Prover sends to the Verifier a specially encoded binary String proving that G_0 and G_1 are non-isomorphic.

What is in the String?

The Prover chooses some ordering of all n-node graphs and puts as the i-th bit of the String to 1 (to 0) if the i-th graph of the chosen ordering is isomorphic to G_1 (to G_0) - otherwise he puts as i-th bit of the String a randomly chosen bit.

How does the String proves to the Verifier that G_0 and G_1 are non-isomorphic?

VERY EASY (in a way): The Verifier flips the coin to choose G_0 or G_1, randomly permutes it to get a graph H. Then she queries the corresponding bit of the String and accepts if and only if the queried bit matches her randomly chosen bit.

The method works. Indeed, if graphs G_0 and G_1 are non-isomorphic, then the Verifier will always accept; if not, then probability of acceptance is at most 1/2.

PCP-THEOREM and APPROXIMATION ALGORITHMS

A surprising connection has been discovered between holographic proofs and highly practical problems of approximability of **NP**-complete problems.

It has been shown how any sufficiently good approximation algorithm for the clique problem can be used to test whether transparent proofs exist, and hence to determine membership in **NP**-complete languages.

On this basis it has been shown for the clique problem - and a variety of other **NP**-hard optimization problems, such as graph coloring - that there is a constant $\varepsilon > 0$ such that no polynomial time approximation algorithm for the clique problem for a graph with a set of $|V|$ of vertices can have a ratio bound less than $|V|^{\varepsilon}$ unless **P=NP**.