


Probabilistic Proof Systems

Oded Goldreich

Weizmann Institute of Science

Initial comments

- We shall consider proofs for mundane and totally formal theorems; e.g., a specific propositional formula is satisfiable, a specific system of quadratic equations (over a finite field) has a solution, a specific graph (or map) can be properly colored using three colors, etc. These theorems arise in various applications (e.g., Cryptography).
- Proof systems are defined in terms of their verification procedures. We seek efficient verification procedures.

Efficient procedures  efficient algorithms.
Efficient algorithms = polynomial-time algorithms.

- I will **not** mention the applications of these proof systems (to Cryptography and study of approximation), unless you ask...

Traditional proof systems (i.e., NP-proof systems)

- Efficient verification = verification procedure that runs in time that is polynomial in the length of the theorem being claimed.
- **Completeness:** If the claim is valid, then there exists a proof that is accepted by the verification procedure (i.e., “**verifier**”).
- **Soundness:** If the claim is invalid, then no alleged proof will be accepted by the verifier, who will always reject the claim.

The “prover” is implicit in the formulation (and inessential to it):

The **prover** is the person providing alleged proofs.

We implicitly consider a unidirectional communication from the prover to the verifier; the message sent is the alleged proof.

*) NP-proof systems correspond to the complexity class NP, conjectured to extend beyond P.

NP-proof systems: Examples

- The claim is that a given propositional formula is satisfiable.
The alleged proof is a satisfying assignment.
Verification amounts to substitution and calculation (or evaluation).
- The claim is that a given system of quadratic equations over a finite field (say $GF(2)$ or $GF(3)$) has a solution.
The alleged proof is a solution.
Verification amounts to substitution and calculation (or evaluation).
- The claim is that a given graph (map) is 3-colorable.
The alleged proof is a (legal) 3-coloring of the graph.
Verification amounts to checking that the endpoints of each edges are assigned different colors in $\{1, 2, 3\}$.

Interactive proof systems

New ingredients: Randomness and Interaction.

The verifier tosses coins and interacts with the prover.

A “proof” is no longer a static object, it is a process.

The “proof” carries an error probability, which is explicitly bounded.

- Efficient verification = a **probabilistic** and interactive procedure (verifier) that runs in time that is polynomial in the length of the claim.
- **Completeness:** If the claim is valid, then there exists a prover strategy that leads the verifier to accept with probability 1 (alternative: $\geq 2/3$).
- **Soundness:** If the claim is invalid, then no (“cheating prover”) strategy can lead the verifier to accept with probability greater than $\frac{1}{2}$ (alt: $1/3$).

The verifier and the prover are explicit in the formulation.

The **error probability** can be reduced by repetitions.

Interactive proof systems: Comments + Example

- If the verifier is deterministic, then IP-systems collapses to NP-systems.
“No point to interact with a predictable person who is computationally weaker.”
- Interactive proofs are akin to daily processes such as cross-examination in court, asking questions regarding a proof described in a lecture, and mental experiments that take place in traditional proofs (i.e., “For an arbitrary X , do $F(x)$ ”).
- THM [Goldwasser&Sipser]: Wlog, suffices to ask totally random questions.

One day on Olympus, bright-eyed Athena claimed that nectar poured from new silver-coated jars tasted less sweet than nectar poured from older gold-decorated jars. Mighty Zeus, who was forced to introduce the new jars by the practically minded Hera, was annoyed at the claim.

He ordered that Athena be served one hundred glasses of nectar, each poured at random either from an old jar or from a new one, and that she tell the source of the drink in each glass. To everybody's surprise, wise Athena correctly identified the source of each serving, to which the father of the gods responded, “My child, you are either right or extremely lucky.” Since all the gods knew that being lucky was not one of the attributes of Pallas-Athena, they all concluded that the impeccable goddess was right in her claim.

THM [G., Micali, Wigderson]: Graph Non-Isomorphism has an interactive proof system.

The power of interactive proof systems

THM [Lund, Fortnow, Karloff, and Nisan]: coNP is in IP .

Every set in coNP (i.e., the set of wrong claims for an NP -proof system) has an interactive proof system. E.g., one can prove (interactively) that

- A given propositional formula is NOT satisfiable.
- A given system of quadratic equations over a finite field (say $\text{GF}(2)$ or $\text{GF}(3)$) has NO solution.
- A given graph (map) is NOT 3-colorable.

“If the existence of something in some situation can be proved, then also its non-existence in this situation can be proved (interactively).”

THM [Shamir]: $\text{PSPACE} = \text{IP}$. (Corollary: $\text{coIP} = \text{IP}$.)

IP = all sets having interactive proof systems.

PSPACE = all sets that can be decided in polynomial amount of space.

Zero-knowledge proof systems

Typically, proofs yields much beyond their validity.

In contrast, ZK proofs yield nothing beyond.

“Whatever can be efficiently computed after interacting with the prover, can be efficiently computed assuming the claim is correct.”

Formally, for an interactive proof (P,V) and any (valid) claim x , we consider two distributions:

1. The output generated by V
(or even by any feasible “knowledge-seeking adversary”)
on input x after interacting with the prover strategy P .
2. The output of some efficient procedure (“simulator”) on input x .

We require that these distributions are identical / statistically-close / computationally-indistinguishable.

Zero-knowledge proof systems: Comments + Example

- Not possible with NP-proof systems: If an NP-proof system is zero-knowledge, then the verifier does not need the prover (i.e., can decide by itself).

THM [G., Micali, and Wigderson]:

There exists a (perfect) zero-knowledge proof system for Graph Isomorphism.

The prover sends the verifier a random isomorphic copy of the 1st input graph, and the verifier selects at random $i \in \{1,2\}$, sends i to the prover, who is required to respond with the isomorphism between the i^{th} input graph and the graph sent in step 1.

Soundness: If the input graphs are not isomorphic, the prover fails w.p. (at least) $\frac{1}{2}$.

Zero-knowledge: The simulator selects $j \in \{1,2\}$ at random, places a random isomorphic copy of the j^{th} input graph, and produces output if $i=j$. (Recall: i is chosen by the verifier.)

The power of zero-knowledge proof systems

THM [G., Micali, and Wigderson]: Assuming one-way functions, every NP-proof system can be transformed to a zero-knowledge proof system.

E.g., one can prove in zero-knowledge that

- A given propositional formula is satisfiable.
- A given system of quadratic equations over a finite field (say $GF(2)$ or $GF(3)$) has a solution.
- A given graph (map) is 3-colorable.

“If the existence of something in some situation can be proved, then also it can be proved in zero-knowledge.”

THM [Ben-Or et al]: Under same assumption, every interactive proof system can be transformed to a zero-knowledge proof system.

Probabilistically checkable proof (PCP) systems

[Feige, Goldwasser, Lovasz, Safra, and Szegedy] & [Arora and Safra]

**Back to NP-proofs, but in redundant form,
which are probed at few random locations.**

The (randomized) verifier has direct access to bits of the alleged proof.

- Super-fast verification = a **probabilistic** machine (verifier) that tosses logarithmically many coins and makes a constant number of probes (to an alleged proof of polynomial length).
- **Completeness:** If the claim is valid, then there exists a proof that makes the verifier accept with probability 1.
- **Soundness:** If the claim is invalid, then the verifier rejects with probability at least $\frac{1}{2}$, no matter which (false) proof is presented.

The **error probability** can be reduced by repetitions.

The power of PCP systems

THM [FGLSS, AS, ALMSS]: Every NP-proof system can be transformed to a PCP system. (Furthermore, an NP-proof for any valid claim can be efficiently transformed into a suitable proof for the PCP System.)

This yields **amplifying reductions** for many natural sets, including

- Satisfiable propositional 3CNF formula, where 3CNF = Conjunctive Normal Form with 3 literals in each clause.
- 3-colorable graphs (maps).

Def: f is an amplifying reduction for 3SAT if it satisfies:

- **Completeness:** If ϕ is satisfiable, then $f(\phi)$ is satisfiable.
- **Soundness:** If ϕ is not satisfiable, then each truth-assignment to $f(\phi)$ satisfies at most 99% of the clauses. (Can be improved to $7/8 + o(1)$.)

This means that approximating the fraction of clauses in a formula that can be simultaneously satisfied is as hard as determining whether the formula is satisfiable.

The proof(s) of the PCP Theorem

Both proofs use the notion of “proof composition”

$$\begin{aligned} & \text{“NP in } r\text{PCP}[r_{\text{out}}, q_{\text{out}}] \text{”} + \text{“P in PCPP}[r_{\text{in}}, q_{\text{in}}] \text{”} \\ & = \text{“NP in } r\text{PCP}[r_{\text{out}} + r_{\text{in}}(q_{\text{out}}), q_{\text{in}}(q_{\text{out}})] \text{”} \end{aligned}$$

1st proof [Arora, Lund, Motwani, Sudan, and Szegedy]:

- “NP in $r\text{PCPP}[\log, \text{polylog}]$ ”
- “NP in $r\text{PCPP}[\text{poly}, O(1)]$ ”

2nd proof [Dinur]: For some constant $c > 1$, and every $\rho < 1/c$

- $\text{PCP}_{\rho}[r, c]$ in $\text{PCP}_{4\rho}[r + O(1), O(1)]$ “Gap (i.e., detect.-prob.) Amplification”
- $\text{PCP}_{4\rho}[r', O(1)]$ in $\text{PCP}_{2\rho}[r' + O(1), c]$. (Uses composition with a $\text{PCPP}[\text{poly}, c]$ system)

*) $\text{PCP}[r, q]$ = PCP system with randomness complexity r (proof length $\exp(r)$) and query complexity q .

PCP_{ρ} indicates a system with false-detection probability ρ , where $q=1/n$ is trivial and we seek $\rho=1/2$.

Interactive proof systems, revisited: Doubly-Efficient interactive proofs systems

Efficient prover, very efficient verifier.

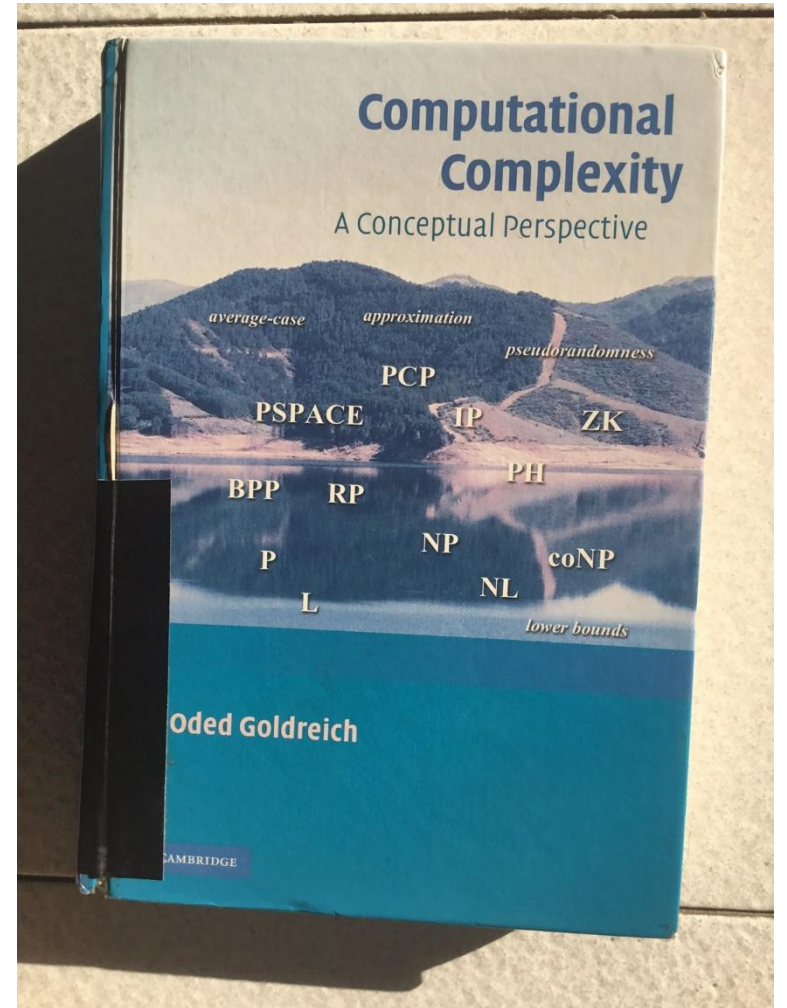
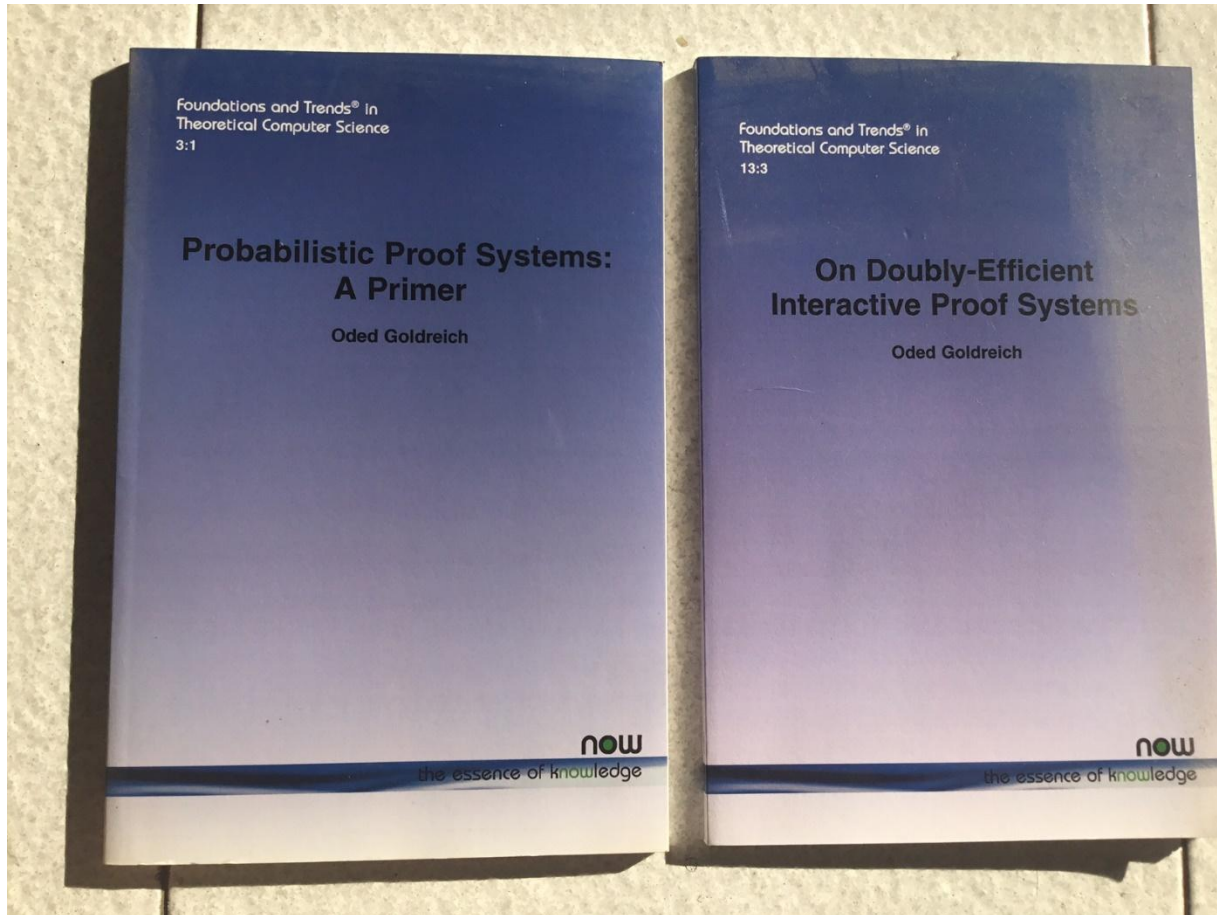
E.g., the prover runs in polynomial-time
but the verifier runs in almost-linear-time.

**So the prover is more powerful, and the verifier can still gain
(e.g., it may be more efficient than a decision procedure).**

The power of doubly-efficient IP systems:

- Cannot exist for claims that cannot be verified both in polynomial-time and in almost-linear-space.
- THM [Reingold, Guy and Ron Rothblum]: Do exist for claims that can be decided by a polynomial-time procedure that uses small (i.e., $n^{o(1)}$) space.

END



See <http://www.wisdom.weizmann.ac.il/~oded/pps.html> and [de-ip.html](http://www.wisdom.weizmann.ac.il/~oded/de-ip.html)

and [cc-book.html](http://www.wisdom.weizmann.ac.il/~oded/cc-book.html)