Probabilistically Checkable Proofs

Madhu Sudan
MIT CSAIL
Happy 75th Birthday, Appa!
Can Proofs Be Checked Efficiently?

The Riemann Hypothesis is true (12th Revision)

By

Ayror Sappen

# Pages to follow: 15783
Proofs and Theorems

- Conventional belief: Proofs need to be read carefully to be verified.

- Modern constraint: Don’t have the time (to do anything, leave alone) read proofs.

- This talk:
  - New format for writing proofs.
  - Efficiently verifiable probabilistically, with small error probability.
  - Not much longer than conventional proofs.
Outline of talk

- Quick primer on the **Computational perspective** on **theorems** and **proofs** (proofs can look very different than you’d think).

- Definition of **Probabilistically Checkable Proofs** (PCPs).

- Some overview of “ancient” (15 year old) and “modern” (3 year old) **PCP** constructions.
Theorems: Deep and Shallow

- **A Deep Theorem:**
  \[ \forall x, y, z \in \mathbb{Z}^+, n \geq 3, \quad x^n + y^n \neq z^n \]

  - **Proof:** (too long to fit in this section).

- **A Shallow Theorem:**
  - The number 3190966795047991905432 has a divisor between 25800000000 and 25900000000.

  - **Proof:** 25846840632.
Computational Perspective

- Theory of NP-completeness:
  - Every (deep) theorem reduces to shallow one.

  Given theorem $T$ and bound $n$ on the length (in bits) of its proof there exist integers $0 \leq A, B, C \leq 2^{n^c}$ such that $A$ has a divisor between $B$ and $C$ if and only if $T$ has a proof of length $T$.

- Shallow theorem easy to compute from deep. $A, B, C$ computable in poly$(n)$ time from $T$.

- Shallow proofs are not much longer.
**P & NP**

- **P =** Easy Computational Problems
  - Solvable in polynomial time
  - (E.g., Verifying correctness of proofs)

- **NP =** Problems whose solution is easy to verify
  - (E.g., Finding proofs of mathematical theorems)

- NP-Complete = Hardest problems in NP

- **Is P = NP?**
  - Is finding a solution as easy as specifying its properties?
  - Can we replace every mathematician by a computer?
  - Wishing = Working!
More Broadly: New formats for proofs

- New format for proof of $T$: Divisor $D$ ($A,B,C$ don’t have to be specified since they are known to (computable by) verifier.)

- Theory of Computation replete with examples of such “alternate” lifestyles for mathematicians (formats for proofs).
  - Equivalence: (1) new theorem can be computed from old one efficiently, and (2) new proof is not much longer than old one.

- Question: Why seek new formats? What benefits can they offer? Can they help?
Probabilistically Checkable Proofs

- How do we formalize “formats”?

- Answer: Formalize the Verifier instead. “Format” now corresponds to whatever the verifier accepts.

- Will define PCP verifier (probabilistic, errs with small probability, reads few bits of proof) next.
PCP Verifier

1. Reads Theorem
2. Tosses coins
3. Reads few bits of proof

\[ T \text{ Valid} \Rightarrow \exists P \text{ s.t. } V \text{ accepts w.p. } 1. \]

\[ T \text{ invalid} \Rightarrow \forall P, V \text{ accepts w.p. } \leq \frac{1}{2}. \]
Features of interest

- Number of bits of proof queried must be small (constant?).
- Length of PCP proof must be small (linear?, quadratic?) compared to conventional proofs.

- Optionally: Classical proof can be converted to PCP proof efficiently. (Rarely required in Logic.)

- Do such verifiers exist?

- PCP Theorem [Arora, Lund, Motwani, S., Szegedy, 1992]: They do; with constant queries and polynomial PCP length.

- [2006] – New construction due to Dinur.
Part II - Ingredients of PCPs
Essential Ingredients of PCPs

- **Locality of error:**
  - If theorem is wrong (and so “proof” has an error), then error in proof can be pinpointed **locally** (found by verifier that reads only few bits of proof).

- **Abundance of error:**
  - Errors in proof are **abundant** (easily seen in random probes of proof).

- How do we construct a proof system with these features?
Localy: From NP-completeness

- 3-Coloring is NP-complete:

Color vertices s.t. endpoints of edge have different colors.
3-Coloring Verifier:

- To verify: Picks an edge and verifies endpoints distinctly colored.
- Error: Monochromatic edge = 2 pieces of proof.
- Local! But errors not frequent.
Amplifying error: Algebraic approach

- Graph = $E : V \times V \rightarrow \{0,1\}$
  - Place $V$ in finite field $\mathbb{F}$
  - Convert $E$ to polynomial
    \[
    \hat{E} : \mathbb{F} \times \mathbb{F} \rightarrow \mathbb{F} \quad \text{s.t.} \quad \hat{E}|_{V \times V} = E
    \]

- Algebraize search:
  - Want $\chi : \mathbb{F} \rightarrow \mathbb{F}$ s.t.
    \[
    \chi(v) \cdot (\chi(v) - 1) \cdot (\chi(v) - 2) = 0, \quad \forall v \in V
    \]
    \[
    \hat{E}(u, v) \cdot \prod_{i \in \{-2, -1, 1, 2\}} (\chi(u) - \chi(v) - i) = 0, \forall u, v \in V
    \]
Algebraic theorems and proofs

- Theorem: Given $V \subseteq \mathbb{F}$, operators $A, B, C$; and degree bound $d$
  $\exists \chi$ of degree $d$ s.t. $A(\chi), B(\chi), C(\chi)$ zero on $V$

- Proof:
  - Evaluations of $\chi, A(\chi), B(\chi), C(\chi)$
  - Additional stuff, e.g., to prove zero on $V$

- Verification?
  - Low-degree testing (Verify degrees)
    - ~ “Discrete rigidity phenomena”?
  - Test consistency
    - ~ Error-correcting codes!
Say want to show $\chi \cdot (\chi - 1) \cdot (\chi - 2) = 0$ on $V$

$$\chi$$

$$\Gamma = \chi \cdot (\chi - 1) \cdot (\chi - 2)$$

$$\Delta = \frac{\Gamma}{(\prod_{u \in V}) (x - u)}$$

Checks: $\chi, \Gamma, \Delta$ are low-degree polynomials

$\chi(\alpha), \Gamma(\alpha), \Delta(\alpha)$ consistent
**Amplifying Error: Graphically**

- Dinur Transformation: **There exists a linear-time algorithm** $A$:

  - $A(G)$ 3-colorable if $G$ is 3-colorable
  - **Fraction of monochromatic edges in** $A(G)$ **is twice the fraction in** $G$
    (unless fraction in $G$ is $\geq \epsilon_0$).
Graphical amplification

- Series of applications of $A$:
  - Increases error to absolute constant
  - Yield PCP
- Achieve $A$ in two steps:
  - Step 1: Increase error-detection prob. By converting to (generalized) $K$-coloring
    - Random walks, expanders, spectral analysis of graphs.
  - Step 2: Convert $K$-coloring back to 3-coloring, losing only a small constant in error-detection.
  - Testing (~ “Discrete rigidity phenomenon” again)
Conclusion

- **Proof verification by rapid checks is possible.**
  - Does not imply math. journals will change requirements!
  - But not because it is not possible!
  - Logic is not inherently fragile!

- **PCPs build on and lead to rich mathematical techniques.**

- **Huge implications to combinatorial optimization**
  ("inapproximability")

- **Practical use?**
  - Automated verification of “data integrity”
  - Needs better size tradeoffs
  - ... and for practice to catch up with theory.
Thank You!