# Probabilistically Checkable Proofs

**Madhu Sudan**

MIT CSAIL

# Happy 75$^{th}$ Birthday, Appa!

# Can Proofs Be Checked Efficiently?

The Riemann Hypothesis is true (12$^{th}$ 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:

$$\forall x, y, z \in \mathbb{Z}^+, n \geq 3, \;\; x^n + y^n \neq z^n$$

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

- Shallow theorem easy to compute from deep.

$\mathbf{A, B, C}$ **computable in poly$(\mathbf{n})$ time from $\mathbf{T}$.**

- 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

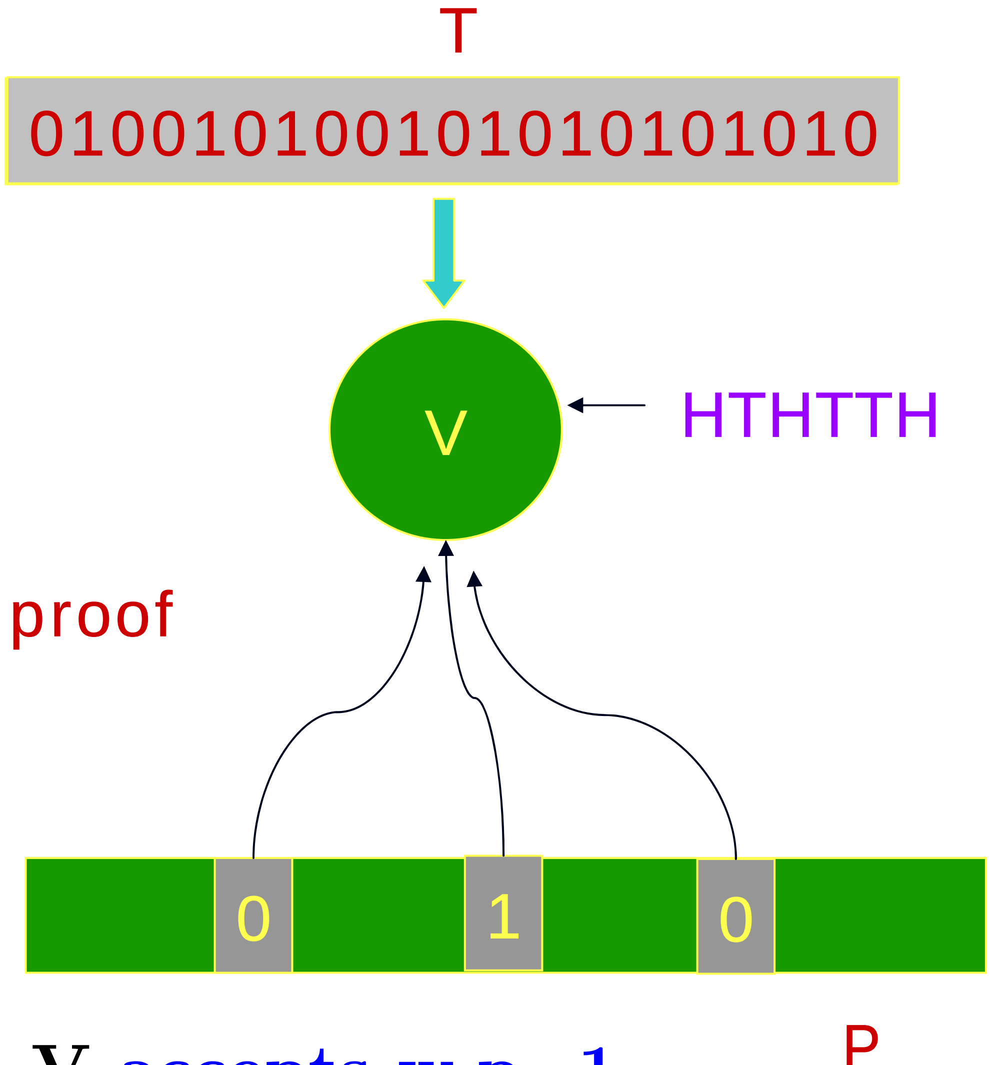
- How do we formalize “formats”?

- 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
4. Accepts/Rejects.




$$\mathbf{T}\ \textbf{Valid} \Rightarrow\ \exists\ \mathbf{P}\ \textbf{s.t.}\ \mathbf{V}\ \textbf{accepts w.p. } \mathbf{1}.$$

$$\mathbf{T}\ \textbf{invalid} \Rightarrow\ \forall\ \mathbf{P},\ \mathbf{V}\ \textbf{accepts w.p. } \leq \frac{\mathbf{1}}{\mathbf{2}}.$$

# 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

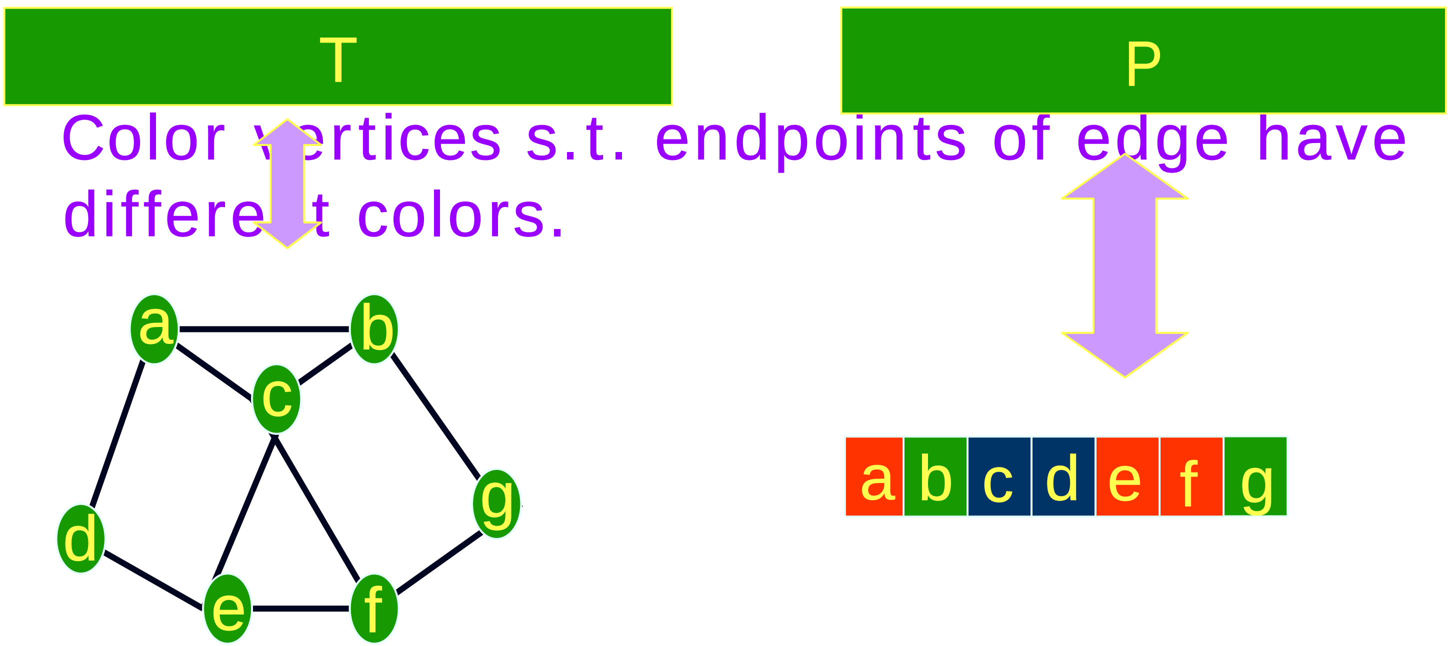
# Essential Ingredients of PCPs

- Locality of error:
  - If theorem is wrong (and so "proof" has an error), then error in proof can be pinpointed <u>locally</u> (found by verifier that reads only few bits of proof).
- Abundance of error:
  - Errors in proof are <u>abundant</u> (easily seen in random probes of proof).
- How do we construct a proof system with these features?

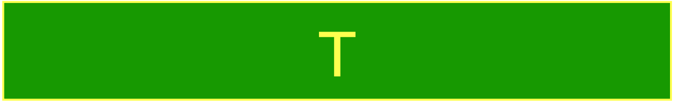
# Locality: From NP-completeness

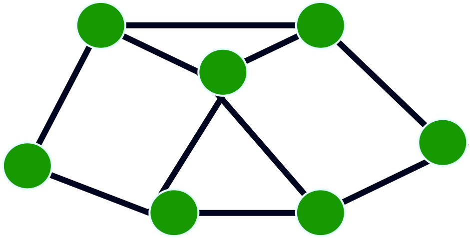
- 3-Coloring is NP-complete:

Color vertices s.t. endpoints of edge have different colors.



# 3-Coloring Verifier:

- To verify 

- Verifier constructs 
- Expects  as proof.
- 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 x V** $\rightarrow$ **{0,1}**

  **Place $\mathbf{V}$ in finite field $\mathbb{F}$**

  **Convert $\mathbf{E}$ to polynomial**

  $\hat{\mathbf{E}} : \mathbb{F} \times \mathbb{F} \rightarrow \mathbb{F}$ **s.t.** $\hat{\mathbf{E}}\big|_{\mathbf{V}\times\mathbf{V}} = \mathbf{E}$

- Algebraize search:

  **Want** $\chi : \mathbb{F} \rightarrow \mathbb{F}$ **s.t.**

$$\chi(\mathbf{v}) \cdot (\chi(\mathbf{v}) - \mathbf{1}) \cdot (\chi(\mathbf{v}) - \mathbf{2}) = \mathbf{0}, \quad \forall \mathbf{v} \in \mathbf{V}$$

$$\hat{\mathbf{E}}(\mathbf{u}, \mathbf{v}) \cdot \prod_{\mathbf{i} \in \{-\mathbf{2}, -\mathbf{1}, \mathbf{1}, \mathbf{2}\}} (\chi(\mathbf{u}) - \chi(\mathbf{v}) - \mathbf{i}) = \mathbf{0}, \forall \mathbf{u}, \mathbf{v} \in \mathbf{V}$$

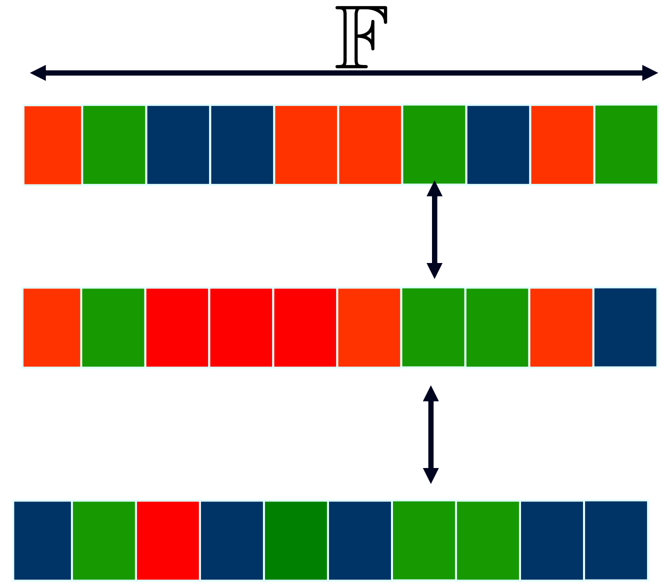
# Algebraic theorems and proofs

- Theorem: Given $\mathbf{V} \subseteq \mathbb{F}$, operators **A, B, C**; and degree bound d

  $\exists\chi$ **of degree d s.t.** $\mathbf{A}(\chi), \mathbf{B}(\chi), \mathbf{C}(\chi)$ **zero on V**
- Proof:
  - Evaluations of $\chi$, $\mathbf{A}(\chi), \mathbf{B}(\chi), \mathbf{C}(\chi)$
  - Additional stuff, e.g., to prove zero on **V**
- Verification?
  - Low-degree testing (Verify degrees)
    - ~ "Discrete rigidity phenomena"?
  - Test consistency
    - ~ Error-correcting codes!

# Some Details

**Say want to show** $\chi \cdot (\chi - \mathbf{1}) \cdot (\chi - \mathbf{2}) = \mathbf{0}$ **on** $\mathbf{V}$




$$\chi$$

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

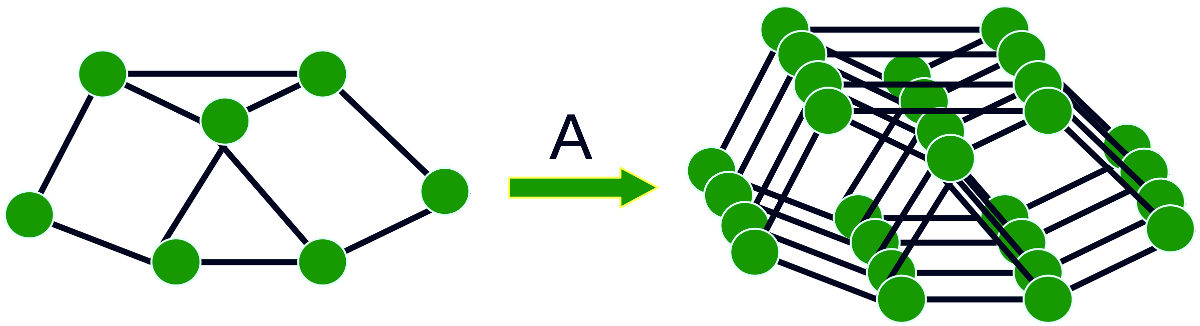
$$\Delta = \frac{\Gamma}{(\prod_{u \in V})(x - u)}$$

Checks: $\chi, \mathbf{\Gamma}, \mathbf{\Delta}$ **are low-degree polynomials**

$\chi(\alpha), \mathbf{\Gamma}(\alpha), \mathbf{\Delta}(\alpha)$ **consistent**

# Amplifying Error: Graphically

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




- $\mathbf{A(G)}$ $\mathbf{3}$-**colorable if** $\mathbf{G}$ **is** $\mathbf{3}$-**colorable**
- **Fraction of monochromatic edges in** $\mathbf{A(G)}$ **is twice the fraction in** $\mathbf{G}$ **(unless fraction in** $\mathbf{G}$ **is** $\geq \epsilon_{\mathbf{0}}$**).**

# 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!