Probabilistically Checkable Proofs

Madhu Sudan
MIT CSAIL

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).
- Overview of a new construction of PCPs due to Irit Dinur.

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 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 n.

- Shallow theorem easy to compute from deep one. (A, B, C computable in poly(n) time from T.)
- Shallow proofs are not much longer.

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.

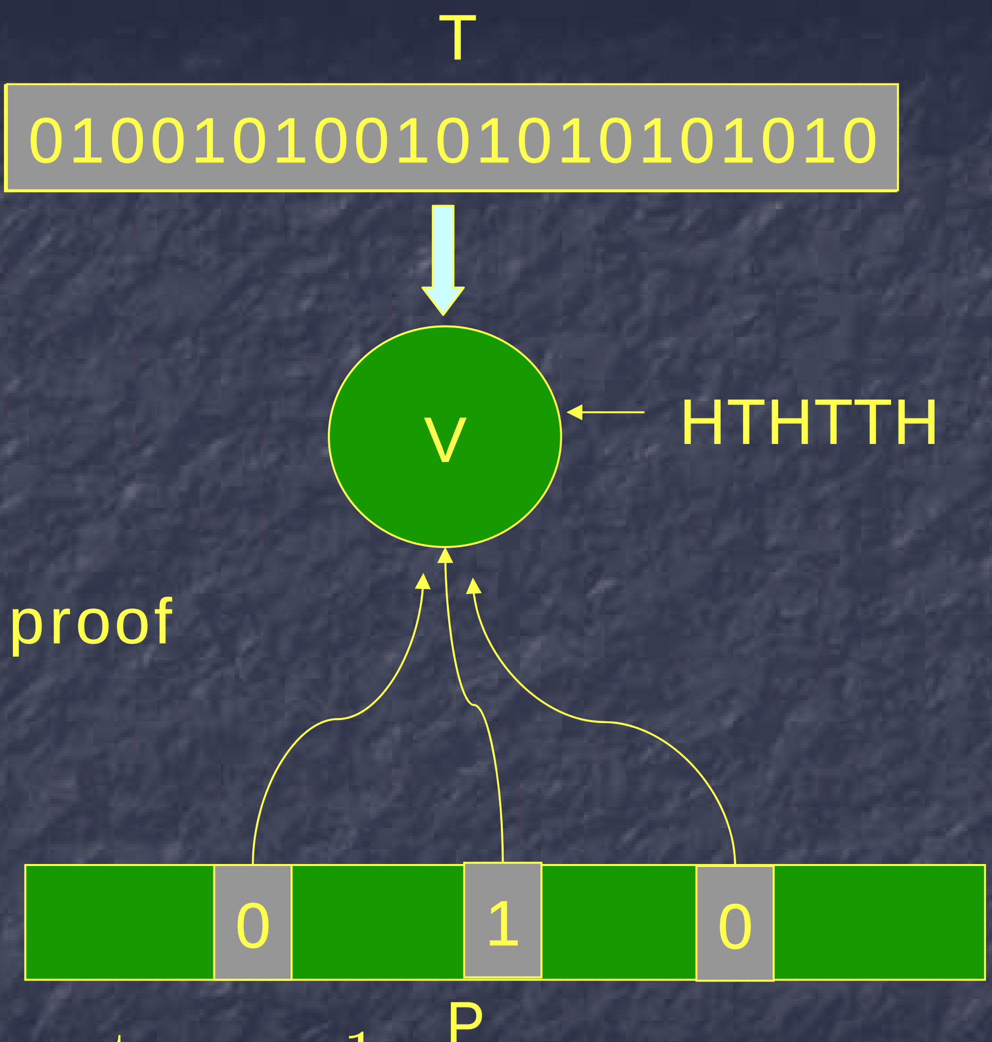
PCP Verifier

1. Reads Theorem

2. Tosses coins

3. Reads few bits of proof

4. Accepts/Rejects.



T Valid $\Rightarrow \exists\ P$ s.t. V accepts w.p. 1.

T invalid $\Rightarrow \forall\ P\ V$ 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 [1992]: They do.
- Today – New construction due to Dinur.

Part II – PCP Construction of Dinur

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 (since it is found by verifier that reads only few bits of proof).
- Abundance of error:
  - Errors in proof are abundant (i.e., easily seen in random probes of proof).
- How do we construct a proof system with these features?

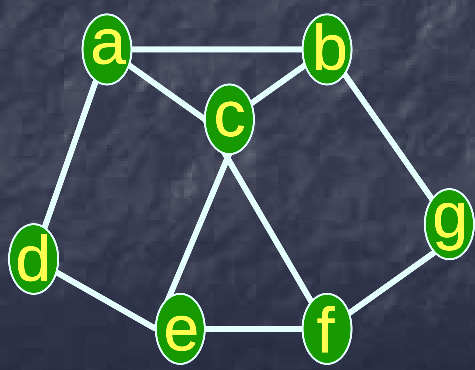
Locality: From NP-completeness

- 3-Coloring is NP-complete:

T

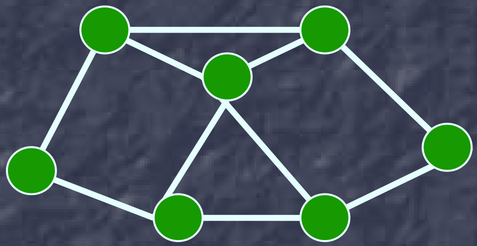
P

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



3-Coloring Verifier:

- To verify T
- Verifier constructs



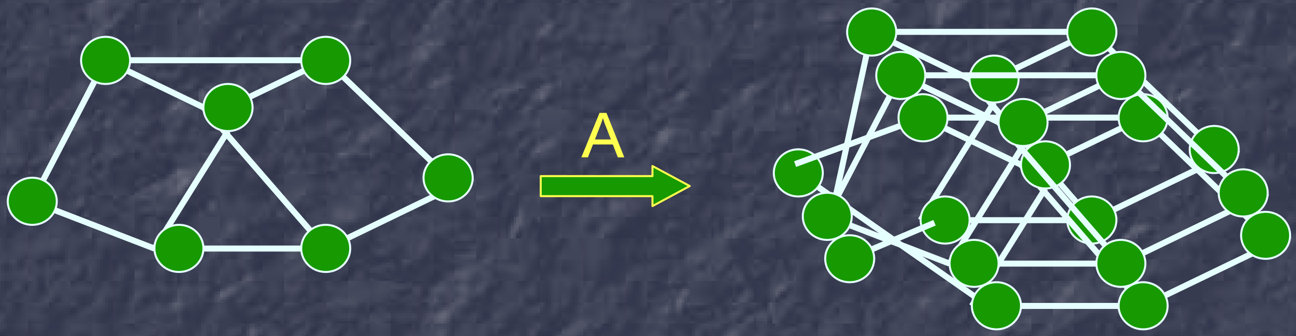
- Expects

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

- 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 $\geq \epsilon_0$).

Iterating the Dinur Transformation

- Logarithmically many iterations of the Dinur Transformation:
  - Leads to a polynomial time transformation.
  - Preserve 3-colorability (valid theorems map to valid theorems).
  - Convert invalid theorem into one where every proof has ε_0 fraction errors.

Details of the Dinur Transformation

- Step 1:

  "Gap Amplification": Increase number of available colors, but make coloring more restrictive.

  - Goal: Increase errors in this stage (at expense of longer questions).

- Step 2:

  "Color reduction": Reduce number of colors back to 3.

  - Hope: Fraction of errors does not reduce by much (fraction will reduce though).

- Composition of Steps yields Transformation.

Step 2: Reducing #colors

- Form of classical "Reductions": similar to task of reducing "k-coloring" to "3-coloring".
- Unfortunately: Classical reductions lose by factor k. Can't afford this.
- However: Prior work on PCPs gave a simple reduction: Lose only a universal constant, independent of k. This is good enough for Step 2.
- (So: Dinur does use prior work on PCPs, but the simpler, more elementary, parts.)

Step 1: Increasing Error

- Task (for example): Create new graph H (and coloring restriction) from G s.t. H is 3^c-color if G is 3-colorable, but fraction of "invalidly colored" edges in H is twice the fraction in G.

- One idea: Graph Products.

  - $V(H) = V(G) \times V(G)$
  - $(u, v) \leftrightarrow_H (w, x) \Leftrightarrow u \leftrightarrow_G w \; \& \; v \leftrightarrow_G x$
  - Coloring valid iff it is valid coordinatewise.

Graph Products and Gap Amplification

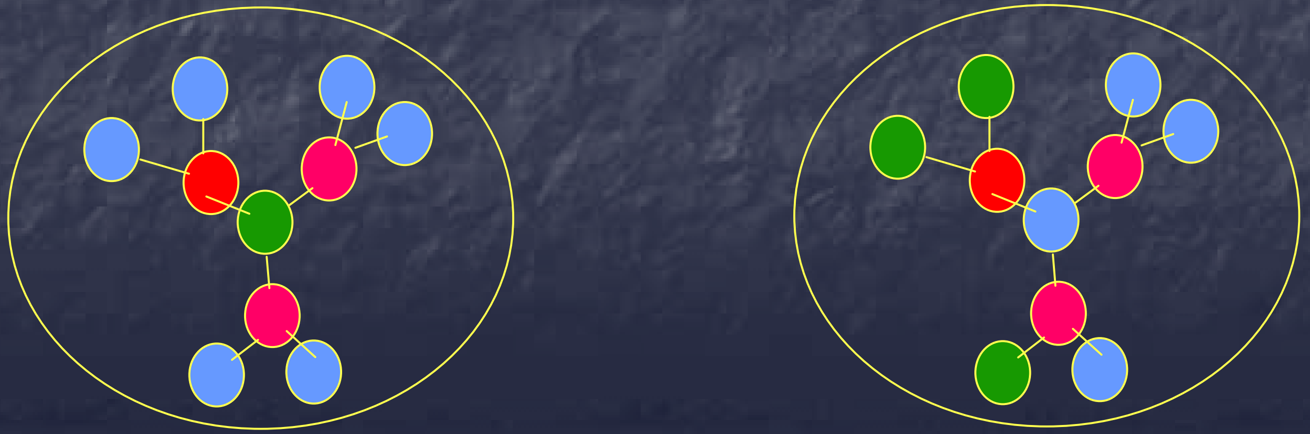
- Problem 1: Not clear that error amplifies. Non-trivial question. Many counter-examples to naïve conjectures. (But might work …)

- Problem 2: Quadratic-blow up in size. Does not work in linear time!!!

- Dinur’s solution: Take a “derandomized graph product”

Step 1: The final construction

- Definition of H (and legal coloring):
  - Vertices of H = Balls of radius t in G
  - Edges of H = Walks of length t in G
  - Legal coloring in H: Coloring to vertices in ball should respect coloring rules in G and two balls should be consistent on intersection.



Analysis of the construction.

- Does this always work?
  - No! E.g., if G is a collection of disconnected graphs, some 3-colorable and others not.
  - Fortunately, connectivity is the only bottleneck. If G is well-connected, then H has the right properties. (Intuition developed over several decades of work in “expanders” and “derandomization”.)
  - Formal analysis: Takes only couple of pages ☺

Conclusion

- A moderately simple proof of the PCP theorem. (Hopefully motivates you. Read original paper at ECCC. Search for “Dinur”, “Gap Amplification”).
- Matches many known parameters (but doesn’t match others).
- E.g., [Håstad] shows can verify proof by reading 3 bits, rejecting invalid proofs w.p. .4999…
- Can’t (yet) reproduce such constructions using Dinur’s technique.

Conclusions (contd.)

- PCPs illustrate the power of specifying a format for proofs.
- Can we use this for many computer generated proofs?

- More broadly: Revisits the complexity of proving theorems vs. verifying proofs.

- Is P=NP?