Probabilistically Checkable Proofs

Madhu Sudan

Microsoft Research

Can Proofs be Checked Efficiently?

Ayror Sappen

# Pages to
follow: 15783

Proofs and Theorems

- Conventional belief: Must read whole proof to verify it.
- Modern Constraint: Don’t have time to (do anything, leave alone to) read proofs.
- This talk:
  - New format for writing proofs.
  - Extremely efficiently verifiable probabilistically, with small error probability.
  - Not much longer than conventional proofs.

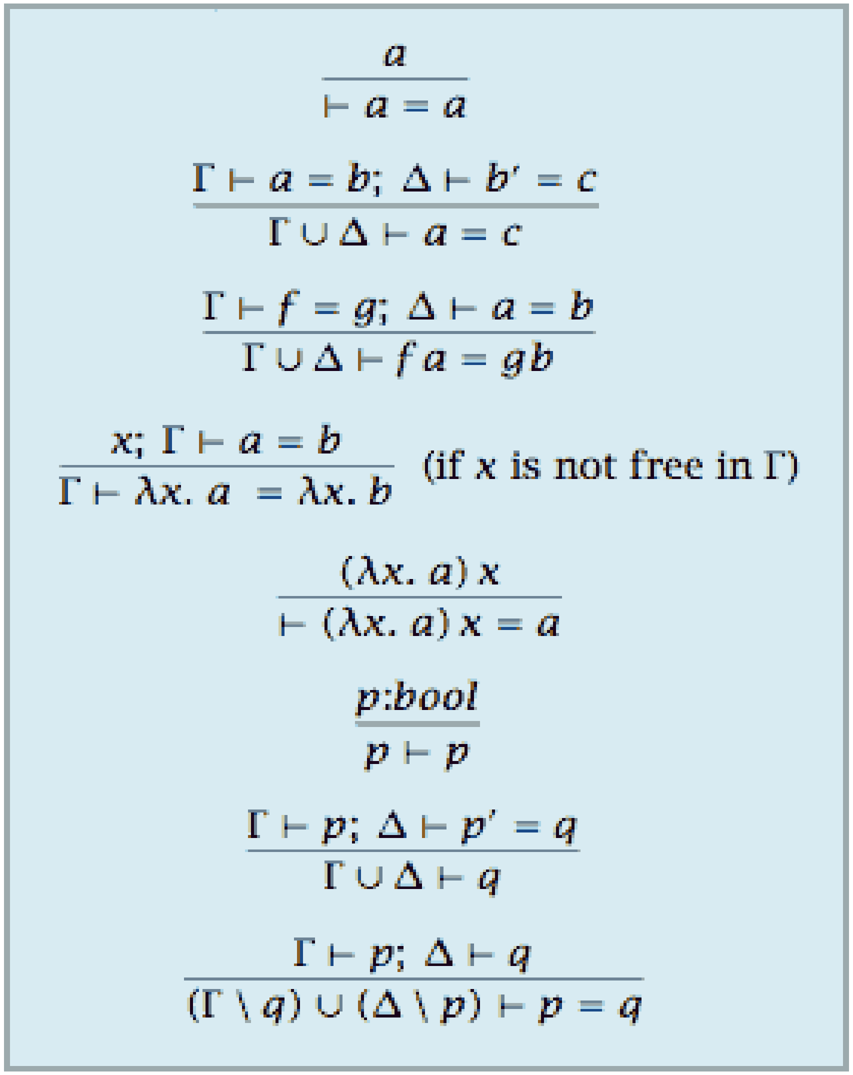
Outline of talk

- 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).
- Why (computer scientists) study proofs/PCPs.
- (Time permitting) Some overview of "ancient" (~25 year old) and "modern" (~10 year old) PCPs.

Part I: Primer

What is a proof?

$$\begin{aligned} a &= b \\ a^2 &= ab \\ a^2 - b^2 &= ab - b^2 \\ (a+b)(a-b) &= b(a-b) \\ a+b &= b \\ 2b &= b \\ 2 &= 1 \end{aligned}$$



$$\frac{a}{\vdash a = a}$$

$$\frac{\Gamma \vdash a = b;\ \Delta \vdash b' = c}{\Gamma \cup \Delta \vdash a = c}$$

$$\frac{\Gamma \vdash f = g;\ \Delta \vdash a = b}{\Gamma \cup \Delta \vdash f\,a = g\,b}$$

$$\frac{x;\ \Gamma \vdash a = b}{\Gamma \vdash \lambda x.\ a\ = \lambda x.\ b} \quad \text{(if } x \text{ is not free in } \Gamma\text{)}$$

$$\frac{(\lambda x.\ a)\,x}{\vdash (\lambda x.\ a)\,x = a}$$

$$\frac{p{:}bool}{p \vdash p}$$

$$\frac{\Gamma \vdash p;\ \Delta \vdash p' = q}{\Gamma \cup \Delta \vdash q}$$

$$\frac{\Gamma \vdash p;\ \Delta \vdash q}{(\Gamma \setminus q) \cup (\Delta \setminus p) \vdash p = q}$$

Philosophy & Computing - 101

- Theorems vs. Proofs?
  - Theorem: “True Statement”
  - Proof: “Convinces you of truth of Theorem”
  - Why is Proof more “convincing” than Theorem?
    - Easier to verify?
      - Computationally simple (mechanical, “no creativity needed”, deterministic?)
      - Computational complexity provides formalism!
      - Advantage of formalism: Can study alternate formats for writing proofs that satisfy basic expectations, but provide other features.

The Formalism

- Theorems/Proofs: Sequence of symbols.
- System of Logic ≡ Verification Procedure V.
  - (presumably V simple/efficient etc.)
- Proof P proves Theorem $T \Leftrightarrow V(T,P)$ accepts.
- T Theorem⇔There exists P s.t. $V(T,P)$ accepts.
- $V \equiv V'$ if both have same set of theorems.
  - But possible different proofs! Different formats!

Theorems: Deep and Shallow

- A Deep Theorem:

$$x, y, z, n \in \mathbb{Z} - \{0\}, n \geq 3 \;\Rightarrow\; x^n + y^n \neq z^n$$

  - Proof: (too long to fit this ~~margin)~~ *talk*.

- A Shallow Theorem:
  - The number 3190966795047991905432 has a divisor between 25800000000 and 25900000000.
  - Proof: 25846840632.

Deep $\leq$ Shallow

- Theory of NP-completeness [**Cook,Levin,Karp'70s**]:
  - Every deep theorem reduces to shallow one!

Given Theorem T and bound N on the length (#symbols) of a proof, there exist integers $0 \leq A, B, C \leq 2^{N^2}$ such that A has a divisor between B and C if and only if T has a proof of length $\leq N$ [Kilian'90s]

  - Shallow theorem easy to compute from deep one.
  - Proof not much longer ($N \rightarrow N^2$)
  - [Polynomial vs. Exponential growth important!]

Aside: P & NP

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

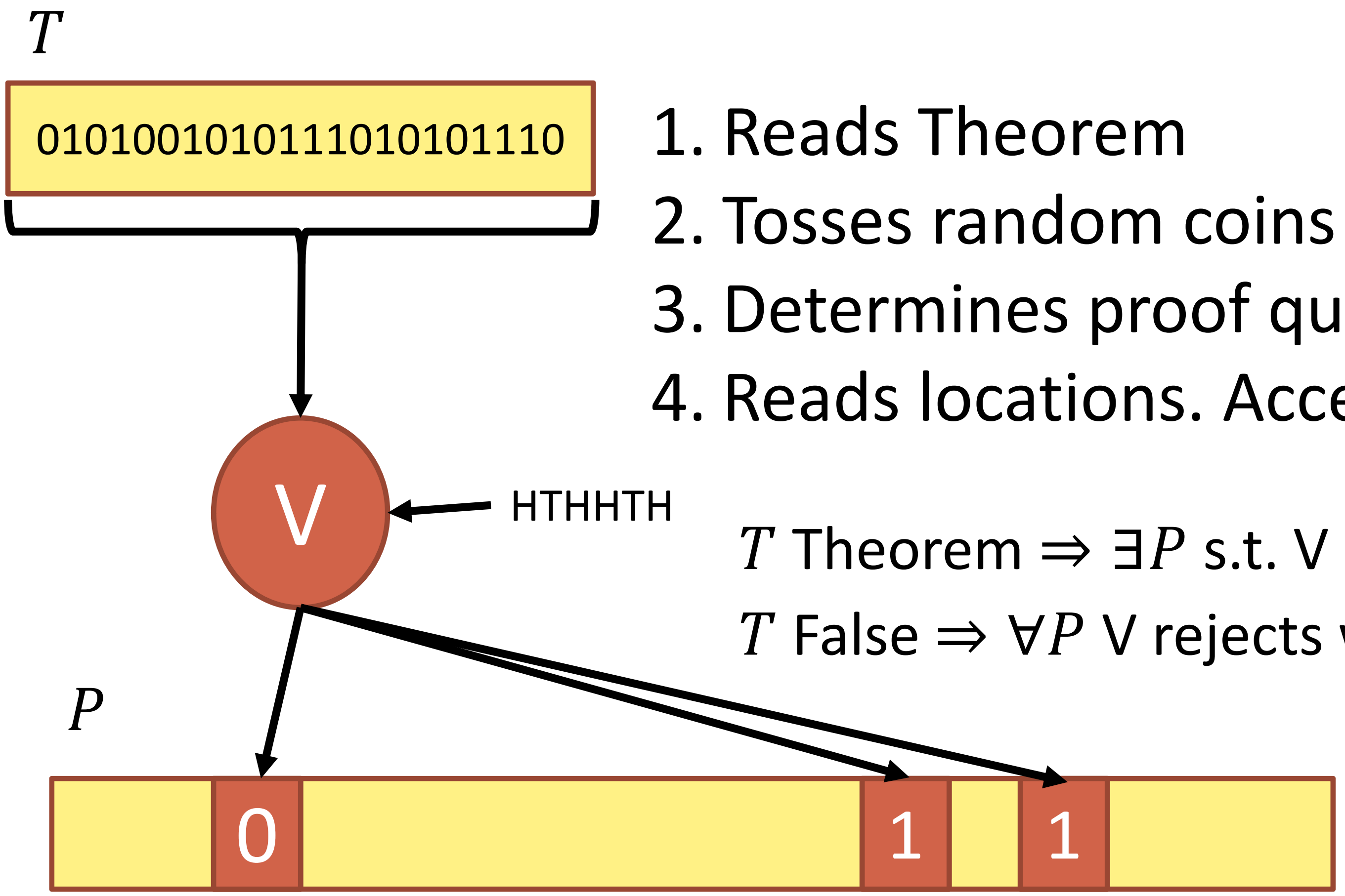
- NP = P   easy to verify
  - (E.g., F
- NP-Co   in NP
- Is P = NP?
  - Is finding a solution as easy as specifying its properties?
  - Can we replace every mathematician by a computer?
  - Wishing = Working!

New Formats for Proofs?

- New format for Proof:
  - “Theorem” T has “Proof” Divisor D
  - New Verifier:
    - Compute A, B, C from T;
    - Verify D divides A; and $B \leq D \leq C$.
- Theory of Computing:
  - Many alternate formats for proofs.
  - Can one of these help

Part II: Prob. Checkable Proof

PCP Format ≡ PCP Verifier



1. Reads Theorem
2. Tosses random coins
3. Determines proof query locations
4. Reads locations. Accepts/Rejects

T Theorem ⇒ $\exists P$ s.t. V accepts (always)

T False ⇒ $\forall P$ V rejects w. prob. 50%

Does such a PCP Verifier, making few queries, exist?

Features of interest

- #queries: Small! Constant? 3 bits?
- Length (compared to old proof):
  - Linear? Quadratic? ~~Exponential?~~
- Transformer: Old proofs => New Proofs?
  - (Not essential, but desirable)
- [Arora,Lund,Motwani,S.,Szegedy'92]: PCPs with constant queries exist.
- [Dinur'06]: New construction
- [Large body of work]: Many improvements (to queries, length)

Part III: Why Proofs/PCPs?

Complexity of Optimization

- Well-studied optimization problems:
  - Map Coloring: Color a map with minimum # colors so adjacent regions have different colors.
  - Travelling Salesman Problem: Visit n given cities in minmum time.
  - Chip Design: Given two chips, are they functionally equivalent?
  - Quadratic system: Does a system of quadratic equations in n variables have a solution?
- [Pre 1970s] All seem hard? And pose similar barriers
- [Cook,Levin,Karp'70s]: All are equivalent, and equivalent to automated theorem proving.
  - Given T, and length N, find proof P of length $\leq N$ proving T.

Approximation Algorithms

- When problem is intractable to solve optimally, maybe one can find approximate solutions?
  - Find a travelling salesman trip taking $\leq 10\%$ more time than minimum?
  - Find map coloring that requires few more colors than minimum?
  - Find solution that satisfies 90% of the quadratic equations?
- Often such approximations are good enough. But does this make problem tractable?

Theory of Approximability

- 70s-90s: Many non-trivial efficient approximation algorithms discovered.
  - But did not converge to optimum? Why?
- 90s-2015: PCP Theory + Reductions
  - Proved limits to approximability: For many problems gave a limit beyond which finding even approximate solutions is hard.
- PCP ⇒ Inapproximability?
  - PCP ⇒ Finding fairly correct proofs as hard as finding correct ones.

    Proof does not fit this margin
  - Analgous to “finding approximate solutions as hard as finding optimal ones”.

Part IV: PCP Construction Ideas

Aside: Randomness in Proofs

- Well explored in Computer Science community in 80s.
- Randomness+Interaction⇒ Many effects
  - Simple Proofs of complex statements
    - Pepsi vs. Coke – the blind taste test.
  - Proofs Revealing very little about its truth
    - Prove “Waldo” exists without ruining game.
  - Proof that some statement has no short proof!

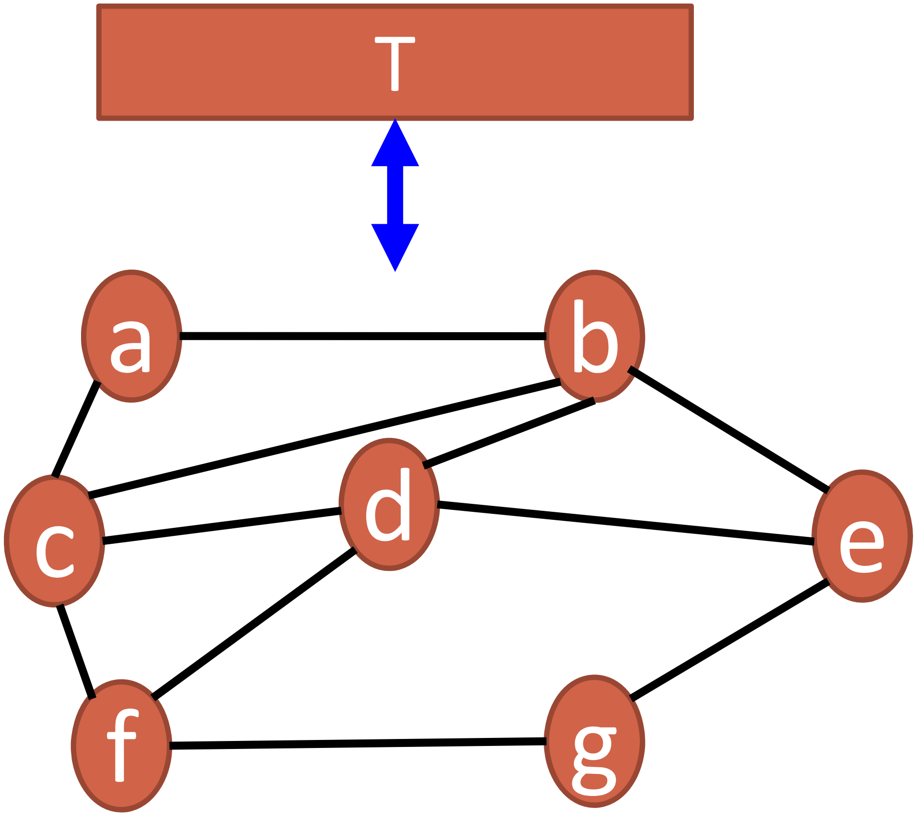
Essential Ingredient of PCPs

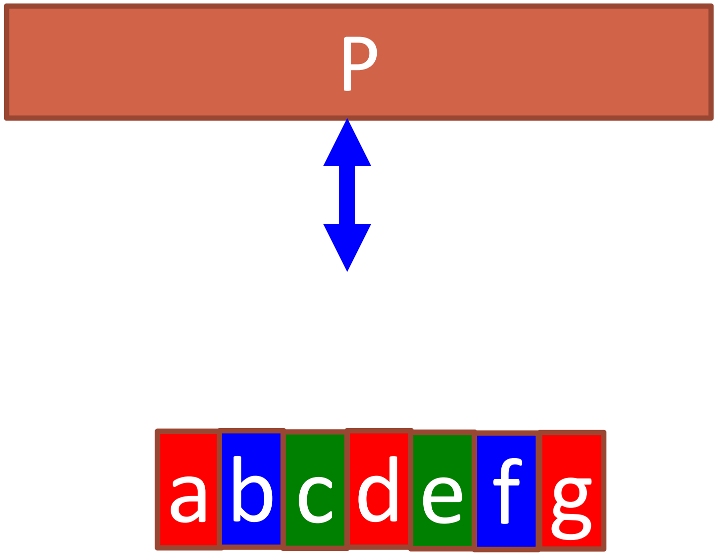
- Locality of error
  - Verifier should be able to point to error (if theorem is incorrect) after looking at few bits of proof.
- Abundance of error
  - Errors should be found with high probability.

- How do get these two properties?

Locality ⇐ NP-completeness

- 3Coloring is NP-complete:





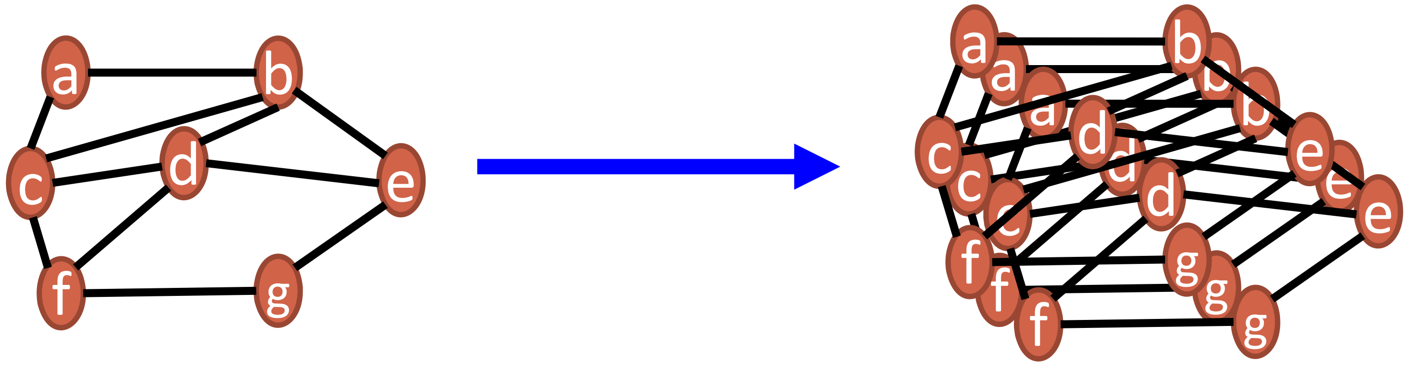
Error is 2-local
Abundance?

Abundance I: via Algebra

- Express (graph-coloring) via Algebra:
- Leads to problems of the form:
  - Given polynomial $A(x,y)$ find $B(x)$ and $C(x,y)$ such that $F(A,B,C) = 0$.
    - Example $F(A,B,C) = A(x,y)^2 - 3y^2C(x+1,y-1)B(x)C(3y)$
    - Actual example doesn't fit this margin ☹
- Advantage of polynomials:
  - Abundance of non-zeroes.
  - Non-zero polynomial usually evaluates to non-zero.
  - Can test for Polynomials

Abundance II: via Graph Theory

- [Dinur’06] Amplification:



- Constant Factor more edges
- Double fraction of violated edges (in any coloring)
- Repeat many times to get fraction upto constant.

Wrapping up

- PCPs
  - Highly optimistic/wishful definition
  - Still achievable!
  - Very useful
    - Understanding approximations (Hugely transformative)
    - Checking outsourced computations
    - Unexpected consequences: Theory of locality in error-correction

Back to Proofs: Philosophy 201

- So will math proofs be in PCP format?
- NO!
  - Proofs *never* self-contained.
    - Assume common language.
  - Proofs also rely on common context
    - Repeating things we all know is too tedious.
  - Proofs rarely intend to convey truth.
    - More vehicles of understanding/knowledge.
- Still PCP theory might be useful in some contexts:
  - Verification of computer assisted proofs?

Thank You!