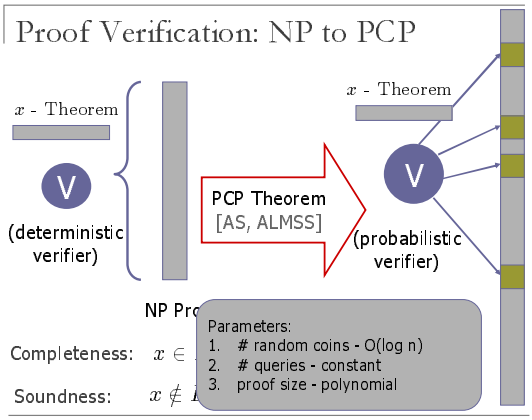
Short PCPs verifiable in Polylogarithmic Time

Eli Ben-Sasson, TTI Chicago & Technion
Oded Goldreich, Weizmann
Prahladh Harsha, Microsoft Research
Madhu Sudan, MIT
Salil Vadhan, Harvard

Proof Verification: NP to PCP



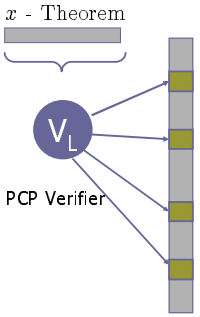
Study of PCPs

Initiated in the works of

- [BFLS]
  - positive result
- [FGLSS]
  - negative result

Very different emphases

BFLS: Holographic proofs



- Direct Motivation: Verification of Proofs
- Important Parameters
  - Proof Size
  - Verifier Running Time
    - randomness
    - query complexity

FGLSS: Inapproximability Connection

- Dramatic Connection
  - PCPs and Inapproximability
- Important Parameters
  - randomness
  - query complexity

Work since BFLS and FGLSS

- Almost all latter work focused on the inapproximability connection
  - improving randomness and query complexity of PCPs
- Very few works focused on PCP size
  - specifically, [PS, HS, GS, BSVW, BGHSV, BS]
- No latter work considered the verifier's running time
- This paper: revisit study of efficient PCPs

Short and Efficient PCPs?

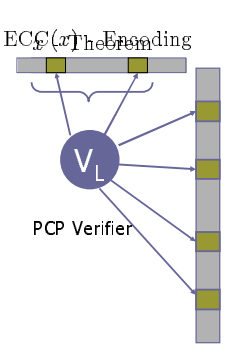
- Lower Bounds
  - Tightness of inapproximability results wrt to running time
- Upper Bounds
  - Future "practical implementations" of proof-verification
  - Coding Theory
    - Locally testable codes [GS, BSVW, BGHSV, BS]
    - Relaxed Locally Decodable Codes [BGHSV]
  - Cryptography
    - e.g.: non-blackbox techniques [Bar]

Motivation: short PCP constructions

- [BFLS] Blowup in proof size: n^{ε}
  Running time: poly log n
- Recent progress in short PCP constructions
  - [BGHSV] Blowup: exp $((\log n)^{\varepsilon})$
    # Queries: $O(1/\varepsilon)$
  - [BS] Blowup: poly log n
    # Queries: poly log n

  Can these improvements be accompanied with an efficient PCP verifier?

Sublinear Verification



Sublinear running time?
Not enough to read theorem !

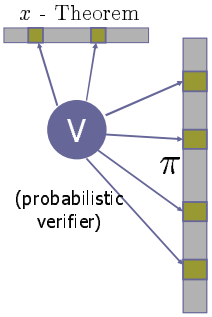
[BFLS] Assume theorem is encoded

Completeness:
$x \in L \Rightarrow$
$\exists \pi, \Pr[V^{Enc(x),\pi} = 1] = 1$

Soundness:
$y - \text{far from } Enc(L) \Rightarrow$
$\forall \pi, \Pr[V^{y,\pi} = 1] \leq \frac{1}{2}$

Important: # queries = sum of queries into encoded theorem + proof

PCP of Proximity (PCPP) [BGHSV, DR]



Completeness:
$x \in L \Rightarrow \exists \pi, \Pr[V^{x,\pi} = 1] = 1$

Soundness:
$x \notin L \Rightarrow$
$\Delta(x, L) > \delta \Rightarrow$
$\forall \pi, \Pr[V^{x,\pi}() = 1] \leq \frac{1}{2}$

- # queries = sum of queries into theorem + proof
- Theorem in un-encoded format
- δ – proximity parameter
- Assignment Testers of [DR]

Our Results: Efficient BS Verifier

- Theorem:
  Every $L \in$ NTIME(T(n)
  proximity with
  - Blowup in proof size:
  - # queries: poly log T(
  - Running time: poly log T(n)

Previous Constructions required polyT(n) time

- Corollary [efficient BS verifier]:
  Every $L \in$ NP has PCPPs with blowup at most poly log n and running time poly log n

Our Results: Efficient BGHSV Verifier

- Theorem:
  Every $L \in$ NTIME(T(n with
  - Blowup in proof size: e
  - # queries: $O(1/\varepsilon)$
  - Running time: poly log T(n)

Previous Constructions required polyT(n) time

- Corollary [efficient BGHSV verifier]:
  Every $L \in$ NP has PCPPs with blowup at most exp $((\log n)^{\varepsilon})$,
  # queries $O(1/\varepsilon)$ and running time poly log n

Efficient PCP Constructions

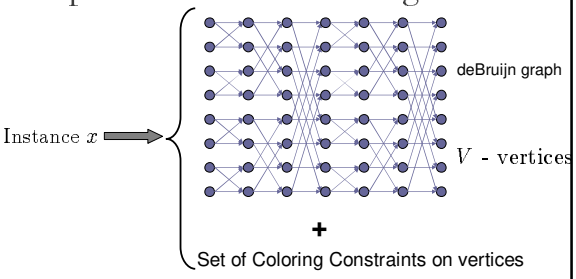
Efficient PCP Constructions

- Overview of existing short PCP constructions
  - specifically, construction of [BS]
- Why these constructions don't give efficient PCPs?
- Modifications to construction to achieve efficiency

PCP Constructions – An Overview

- Algebraic Constructions of PCP (exception: combinatorial const. of [DR])
  - Step 1: reduction to "nice" coloring CSP
  - Step 2: arithmetization of coloring problem
  - Step 3: zero testing problem
    - Note: Step 1 required only for short PCPs. Otherwise arithmetization can be directly performed on SAT. This however blowups the proof size.

Step 1: Reduction to Coloring CSP



- Size of graph $|V| \approx$ size of instance $|x|$
- Graph does not depend on x, depends only on $|x|$.
- Only coloring constraints depend on x

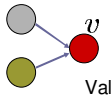
Step 1: Reduction (Contd)

- C – (constant sized) of colors
- Co

Coloring Constraints encode action of NTM on instance x

Coloring Constraint

$\mathrm{Con} : V \times C^3 \to \{0,1\}$



Valid?

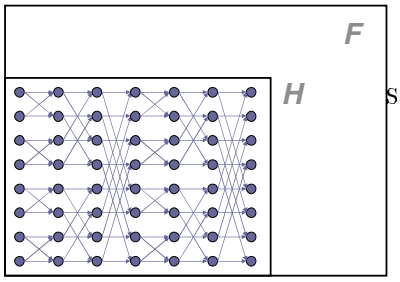
$$x \in L$$

$$\Updownarrow$$

$$\exists \text{ a coloring } \mathrm{Col} : V \longrightarrow C \text{ satisfying all the constraints.}$$

Proof of "$x \in L$": Coloring $\mathrm{Col} : V \to C$

Step 2: Arithmetization



Field F

Subset $H \subset F$

$|H| \approx |V|$

Embed de Bruijn graph in H:
Associate each vertex v with an element $x \in H$

Step 2: Arithmetization (Contd)

- Colors

$$C \longleftrightarrow \text{Constant sized subset of } F$$

- Coloring Constraint

$$\text{Con} : V \times C^3 \to \{0,1\} \longleftrightarrow \hat{\text{Con}} : F \times F^3 \to F$$

- Coloring

$$\text{Col} : V \to C \longleftrightarrow \hat{\text{Col}} : H \to F$$

low degree poly. $p : F \to F$

$x \in L \Leftrightarrow \exists$ a low-degree polynomial $p : F \to F$ such that $\hat{\text{Con}}(x, p(x), p(N(x)), p(N_2(x))) = 0, \forall x \in H$.

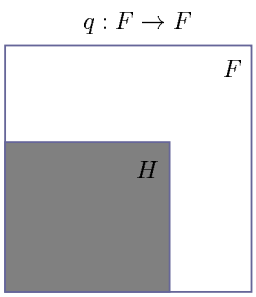
$\exists$ a coloring $\text{Col} : V \longrightarrow C$ satisfying all the constraints.

Proof of "$x \in L$": Polynomials $p,q : F \to F$

Step 3: Zero Testing

- Instance:
  - Field F and subset $H \subseteq F$
  - Function $q: F \to F$

    (specified implicitly as a table of values)
- Problem:
  - Need to check if q is close to a low-degree polynomial that is zero on H
    - Two functions are close if they differ in few points



Low Degree Testing

- Sub-problem of zero-testing
  - Instance:
    - Field F and subset $H \subseteq F$
    - Function $q: F \to F$ (specified implicitly as a table of values)
  - Problem:
    - Check if q is close to a low-degree polynomial.
- Most technical aspect of PCP constructions
- However, can be done efficiently (for this talk)

Step 3: Zero Testing (Contd)

$$\text{Let } Z_H(x) = \prod_{h \in H}(x - h)$$

- Obs: $q: F \to F$ is a low-degree polynomial that vanishes on H if there exists another low-degree polynomial r such that

$$q \equiv r \cdot Z_h$$

- Instance: $q: F \to F$ Proof: $r: F \to F$
  - (Both specified as a table of values)
- Testing Algorithm:
  - Check that both q and r are close to low-degree polynomials (low-degree testing)
  - Choose a random point $x \in_R F$, compute $Z_H(x)$ and check that $q(x) = Z_H(x) \cdot r(x)$

PCP Verifier

- Instance: x Proof: $p,q,r : F \to F$
  - Step 0: [Low Degree Testing]
    - Check that the functions p, q and r are close to low-degree poly.
  - Step 1: [Reduction to Coloring CSP]
    - Reduce instance x to the coloring problem. More specifically, compute the coloring constraint $\text{Con} : V \times C^3 \to \{0,1\}$
  - Step 2: [Arithmetization]
    - Arithmetize the coloring constraint Con to obtain the local rule B
    - Check that at a random point $q = B(p)$ is satisfied
  - Step 3: [Zero Test
    - Choose a random
    - Check that $p(x) =$

However, Steps 1,2 and 3 are NOT efficient in Verifier's running time

Step 3: Zero Testing – Efficient?

- Zero Testing involves computing $Z_H(x)$
- General H: Zero Testing – inefficient
  - Z_H has $|H|$ coefficients
  - Size of instance - $O(|H|)$
  - Hence, requires at least linear time
- Do there exist H for which $Z_H(x)$ can be computed efficiently
- YES!, if H is a subgroup of F instead of an arbitrary subset of F, then Z_H is a sparse polynomial

Facts from Finite Fields

- Fact 1

Let F be an extension field of GF(2) of size 2^q. Suppose $f : F \to F$ is an homomorphism (i.e., $f(x+y) = f(x) + f(y)$, for all $x, y \in F$), then f can be expressed as follows

$$f(x) = c_0 x + c_1 x^2 + c_2 x^4 + \cdots + c_{q-1} x^{2^{q-1}}$$

(i.e., f has a sparse polynomial representation)

- Fact 2

If H is a subgroup of F containing GF(2) (i.e., $x, y \in H \Rightarrow x + y \in H$), then Z_H is a homomorphism.

- Hence, Z_H is sparse (i.e, Z_H has only log $|H|$ coefficients). Moreover, these coeffs. Can be computed in poly log |H| time.

Fact 1: Homomorphisms are sparse

Let F be an extension field of GF(2) of size 2^q. Suppose $f : F \to F$ is an homomorphism (i.e., $f(x+y) = f(x) + f(y)$, for all $x, y \in F$), then f can be expressed as follows

$$f(x) = c_0 x + c_1 X^2 + c_2 x^4 + \cdots + c_{q-1} x^{2^{q-1}}$$

(i.e., f has a sparse polynomial representation)

Proof:

- Set of homomorphisms from F to F form a vector space over F of dimension q
- The functions $x, x^2, x^4, \ldots, x^{2^{q-1}}$ are homomorphisms
- The functions $x, x^2, x^4, \ldots, x^{2^{q-1}}$ are linearly independent

Hence, any homomorphism can be expressed as a linear combination of these functions ■

Fact 2: H subgroup $\Rightarrow Z_H$ homomorphism

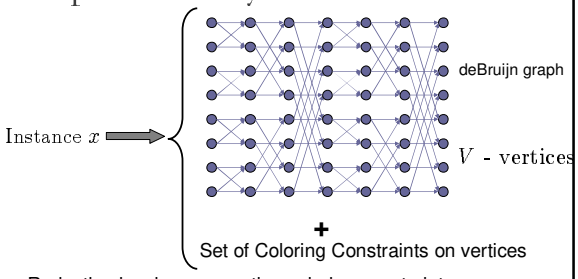
If H is a subgroup of F containing GF(2) (i.e., $x, y \in H \Rightarrow x + y \in H$), then Z_H is a homomorphism.

Proof:

- Need to show
  $p(x, y) \equiv Z_H(x + y) Z_H(x) Z_H(y), \forall x, y \in F$
- Degree of $p \leq |H|$
- If $x \in$ H or $y \in H$, then $p(x,y) = 0$
- Hence, number of zeros of p is $2|H||F| - |H|^2 > |H||F|$
- Fraction of zeros $> |H|/|F| \geq \deg(p)/|F|$

Hence, by Schwartz-Zippel, $p \equiv 0$ ■

Step 1: Efficiency of Reduction



- Reduction involves computing coloring constraint
  Con: $V \times C^3 \to \{0,1\}$
- Not efficient – requires poly $|x|$ time (each constraint needs to look at all of x)

Step 1: Succinct Coloring CSP

- Need to compute constraint without looking at all of x!
- Succinct description: For any node v, the coloring constraint at v can be computed in poly |v| time (by looking at only a few bits of x)
- Even this does not suffice (for arithmetization):
  - Further require that the constraint itself can be computed very efficiently (eg., by an NC_1 circuit)
- Gives a new NEXP-complete problem

Step 1: Succinct Coloring CSP (Contd)

- Succinct Coloring CSP: Same as before
  - DeBruijn graph + Coloring Constraints
  - Additional requirement: Coloring Constraint at each node described by an NC_1 circuit and furthermore given the node v, the circuit describing constraint at node v can be computed in poly $|v|$ time
- Reduction to Succinct CSP uses reduction of TM computations to ones on oblivious TMs [PF]
- Thus, Step 1 can be made efficient

Step 2: Arithmetization – Efficient?

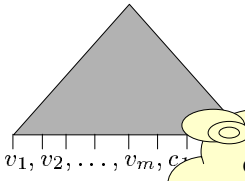
- Arithmetization of coloring constraint

$\mathrm{Con} : V \times C^3 \to \{0,1\} \longleftrightarrow \hat{\mathrm{Con}} : F \times F^3 \to F$

  - Obtained by interpolation
    - Time $O(|V|)=O(|H|)$
  - However, require that the arithmetization be computed in time poly log $|H|$
  - Non trivial !
    - All we know is Con is a small sized (NC_1) circuit when its input is viewed as a sequence of bits
    - Require arithmetization of Con to be small sized circuit when its inputs are now field elements and the only operations it can perform are field operations

Step 2: Efficient Arithmetization

$\mathrm{Con} : V \times C^3 \to \{0,1\}$



- Obs: The function extracting the bit v_i from the field element is a homomorphism $v_i: F \to F$
- Use Fact 1 (of finite fields) again: Homomorphisms are

Resulting algebraic circuit for Constraint
- Degree – $O(|H|)$
- Size – poly log $|H|$

Hence, efficient

- The remaining circ… …ner
  - AND $(x,y) \to x \cdot y$ (prod…
  - NOT$(x) \to (1\text{-}x)$

Putting the 3 Steps together…

- Plug the efficient versions of each step into PCP verifier to obtain the polylog PCP verifier
- Summarizing…
  - Efficient versions of existing short PCP constructions

The End

Thank You