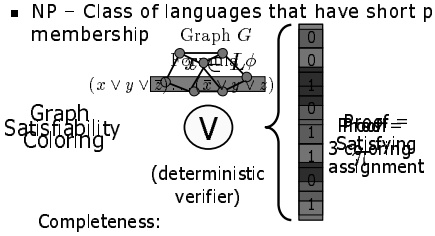
NP – Classical Proofs

- NP – Class of languages that have short proofs of membership



Completeness:

$$x \in L \Rightarrow \exists \pi, V(x, \pi) = \text{accept}$$

Soundness:

$$x \notin L \Rightarrow \forall \pi, V(x, \pi) = \text{reject}$$

Robust PCPs of Proximity

(Shorter PCPs, applications to Coding)

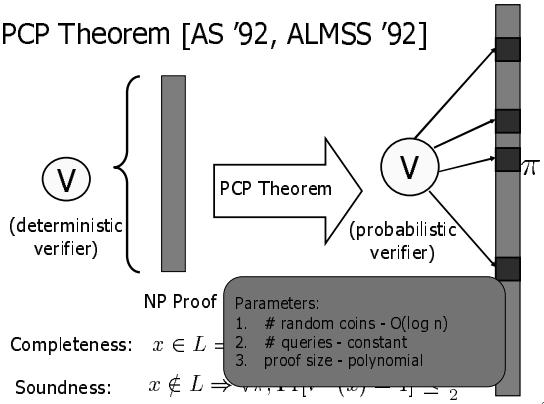
Eli Ben-Sasson (Radcliffe)
Oded Goldreich (Weizmann & Radcliffe)
Prahladh Harsha (MIT)
Madhu Sudan (MIT & Radcliffe)
Salil Vadhan (Harvard & Radcliffe)

PCPs - Significance

- Major impact on the study of combinatorial optimization
  - Consequence: For many NP-hard combinatorial optimization problems, finding near-optimal solutions is also NP-hard
    - Approximating MAXSAT to within a factor of $(8/7 - \varepsilon)$, for any $\varepsilon > 0$, is NP-hard

(Will not dwell into consequence on combinatorial optimization)

PCP Theorem [AS '92, ALMSS '92]



Completeness: $x \in L \Rightarrow$

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

Why Short PCPs? (Contd)

- Lower Bounds
  - Tightness of approximation algorithms with respect to running time
    - e.g.: If SAT has a PCP of size n^{α} then

$$\text{SAT} \in \text{TIME}\left(2^{\Omega(n)}\right)$$

$$\Downarrow$$

Approximating MAXSAT requires time at least $2^{n^{1/\alpha}}$

Short PCPs?

- How long is the new PCP proof?
  - Old NP proof – n ; New PCP proof - *?*
- Why Short PCPs?
  - Upper bounds
    - Cryptography
      - Computationally Sound Proofs and applications [Kil '92, Mic '94, CGH '98, Bar '01]
    - Coding Theory
      - Locally testable codes [GS '02, BSVW '03, this paper]
      - "Relaxed Locally Decodable Codes" [this paper]

Short PCPs vs Query Complexity

	queries	proof size
[HS '00]	17	$n^{3+\epsilon}$
[GS '02, BSVW '03]	$O(1)$	$n \cdot 2^{\sqrt{\log n}}$ $(= n^{1+o(1)})$
This paper	$O(\frac{1}{\epsilon})$	$n \cdot 2^{(\log n)^{\epsilon}}$
	$o(\log \log n)$	$n \cdot 2^{(\log \log n)^{c}}$

8

Short PCPs – Earlier Results

- [PS '94]
  - Proof Size = $n^{1+\varepsilon}$, query = $O(1/\varepsilon)$
    (Constant hidden in big-O $\approx 10^6$)
- [Hås '97]
  - Proof Size = $n^{1000000}$, query = 3;

7

Proof Techniques

- New Definition: Robust PCP of Proximity
  New Composition Theorem
  - Essential for short PCPs
  - simple, modular
- Building Block

10

Our Main Results

Main Theorem:
Satisfiability of circuits of size *n* can be probabilistically verified

- By probing a proof of length $n \cdot 2^{(\log \log n)^{c}}$ in $o(\log \log n)$ bit-locations.

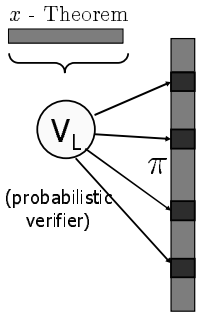
OR

- By probing a proof of length $n \cdot 2^{(\log n)^{\epsilon}}$ in $O(\frac{1}{\epsilon})$ bit-locations.

Previous PCPs required length $n \cdot 2^{\sqrt{\log n}}$ proof size even when reading $2^{\sqrt{\log n}}$ bit-locations [GS '02, BSVW '03]

9

PCP – Definition (Recall)



Completeness:

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

Soundness:

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

Parameters of Interest:

- size of proof ($|\pi|$)
- # queries (q)

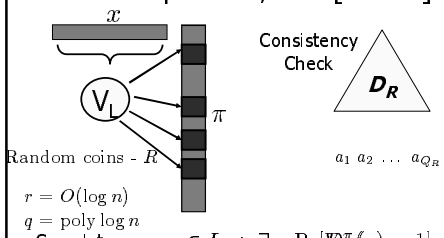
$|\pi| \leq q \cdot 2^{\text{rand}}$

12

Robust PCP of Proximity and Composition Theorem

11

Proof Composition, a la [AS '92]



Completeness: $x \in L \Rightarrow \exists \pi, \Pr[V_R^{x,\pi}(a_1 \ldots a_{Q_R}) = 1] = 1$

Soundness: $x \notin L \Rightarrow \forall \pi, \Pr[V_R^{x,\pi}(a_1 \ldots a_{Q_R}) = 1] \leq \frac{1}{2}$

Need to check if $a_1\ a_2\ \ldots\ a_{Q_R}$ satisfy consistency check D_R

Idea : Use a PCP verifier to check !

14

Why Composition?

- Don't know to build PCPs with $q = O(1)$ and size = poly(n) directly
- However,

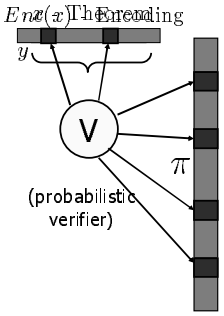
[AS '92, ALMSS '92] type of PCP:
size = poly(n)
q = poly log n
} Verifier V

[AS '92, ALMSS '92] "magically compose" verifier V with itself to obtain new verifier $V \oplus V$ with following parameters

size = poly(n)
q = poly log log n
} $V \oplus V$

13

[BFLS '91] PCP Verifier (Holographic Proofs)



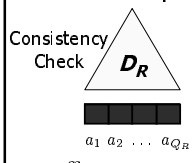
Completeness:
$x \in L \Rightarrow$
$\exists \pi, \Pr[V^{Enc(x),\pi} = 1] = 1$

Soundness:
y – 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

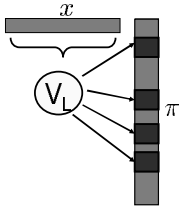
16

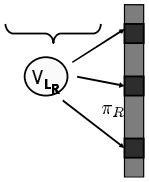
Proof Composition, Contd



Create language
$L_R = \{(a_1, \ldots, a_{Q_R}) | D_R \text{ accepts } (a_1, \ldots, a_{Q_R})\}$

Check if $(a_1, \ldots, a_{Q_R}) \in L_R$ using a PCP verifier

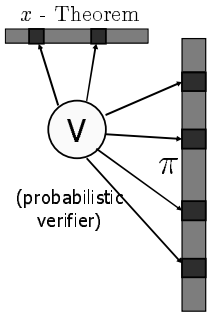




Problem: PCP verifier V_{L_R} needs to read all of theorem (input)
Key Observation:
- "PCP Verifier barely looks at Theorem"
- [BFLS '91] : Assume theorem is encoded and count #queries into theorem

15

PCP of Proximity (PCPP)



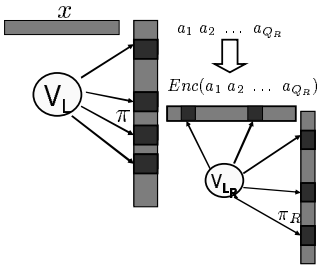
Completeness:
$x \in L \Rightarrow \exists \pi, \Pr[V^{x,\pi} = 1] = 1$

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

- # queries = sum of queries into theorem + proof
- Theorem in un-encoded format
- δ – proximity parameter

18

Proof Composition, Contd



Problem: Need to check $(a_1, \ldots, a_{Q_R})$ and $Enc(a_1\ a_2\ \ldots\ a_{Q_R})$ are consistent.

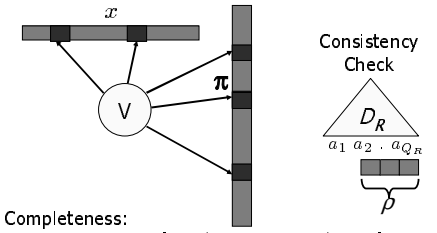
Semantics of arranging this is complex.

Earlier performed by "parallelization" – costly in randomness (large proof size)

Idea: Remove restriction that theorem is encoded !

17

Robust PCP of Proximity (Robust-PCPP) New!



Completeness:

$x \in L \Rightarrow \exists \pi, \Pr[D_R(a_1, \ldots, a_{Q_R}) = 1] = 1$

Robustness:

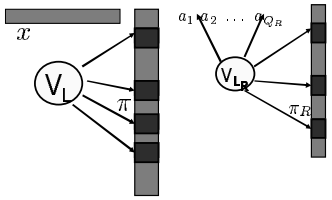
$\Delta(x, L) > \delta \Rightarrow \forall \pi, \Pr[D_R(a_1, \ldots, a_{Q_R}) = 1] \leq \frac{1}{2}$

$\forall \pi, \Pr[(a_1, \ldots, a_{Q_R}) \text{ is } \rho\text{-far from } L_R] > \frac{1}{2}$

ρ - robustness parameter of robust-PCPP

20

Composition again



Completeness: $x \in L \Rightarrow \exists \pi, \Pr[(a_1, \ldots, a_{Q_R}) \in L_R] = 1$

Soundness: $x \notin L \Rightarrow \forall \pi, \Pr[(a_1, \ldots, a_{Q_R}) \notin L_R] > \frac{1}{2}$

Problem:

Need to distinguish between $(a_1, \ldots, a_{Q_R}) \in L_R$ & $(a_1, \ldots, a_{Q_R}) \notin L_R$

PCPP distinguishes between $(a_1, \ldots, a_{Q_R}) \in L_R$ & $\delta -$ far from L_R

Strengthen soundness condition of verifier V_L

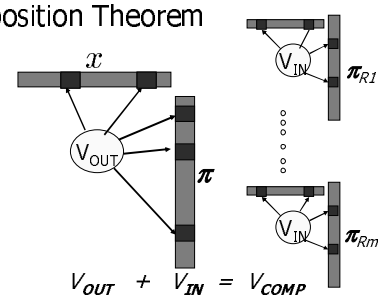
19

Advantages of PCPPs

- Give shortest known PCPs
- Allow natural self-composition
- Simpler constructions of PCPs (no parallelization)
- Coding applications:
  - Simple, highly efficient Locally Testable Codes
  - Simple, highly efficient Relaxed Locally Decodable Codes
- Any efficient property is locally testable (with a little bit of help)

22

Composition Theorem



$V_{OUT} + V_{IN} = V_{COMP}$

- Req. of Inner Verifier: δ_{IN} (proximity) < ρ_{OUT} (robustness)

Randomness: $r_{COMP} = r_{OUT} + r_{IN}$ Queries: $q_{COMP} = q_{IN}$

Proximity: $\delta_{COMP} = \delta_{OUT}$ Robustness: $\rho_{COMP} = \rho_{IN}$

New PCPP Proof for $V_{COMP} = (\pi, \pi_{R1}, \ldots, \pi_{Rm})$

21

Building Block

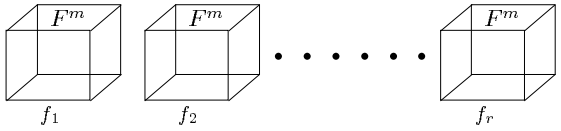
24

PCPPs – Brief History

- Holographic proofs - PCPPs where assignment x is encoded. [BFLS '91]
- PCPP - implicit in low-degree tests [RS '92, ALMSS '92]
- PCPPs - special case of "PCP Spot Checkers" [EKR '99]
- PCPP – extension of Property Testing [RS '92, GGR '96]
- Assignment Testers of [DR '03] similar to PCPPs.

23

Bird's eye-view of PCP construction



PCP Construction: Sequence of function evaluations,

$f_i : F^m \to F$

Checks performed by verifier

- Each function $f_i: F^m \to F$ is a low-degree polynomial
- Input Consistency: f_1 is input
- Each f_{i+1} is obtained consistently from f_i
- Input: Evaluation of function f at each point in F^m
- $f_{i+1}(x) = f_i(x) \cdot f_i(x-1)$
- f_r is close to the final function f_r is identically zero

i.e., $f_r \equiv 0$

How to test if a function is a low-degree polynomial ?

Need to check if evaluation of f is close to the

evaluation of a low-degree polynomial

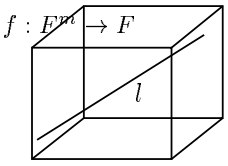
26

Robust PCPPs constructions

- Most existing PCP constructions can be modified to obtain robust PCPs of Proximity
- However, the parameters of such robust-PCPPs do not satisfy our needs
- So, build robust PCPP from scratch

25

Low Degree Test (LDT)



Input: Table of evaluations of f at each point of F^m

Output: Is f low-degree?

- Choose a random line l
- Read f along line l
- Check that restriction of f along l is a univariate low-degree polynomial

- Robust Soundness of LDT [RS '92, ALMSS '92]

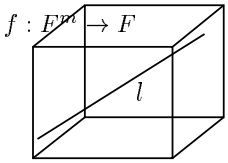
  $f: F^m \to F$ is δ-far from low degree, then

  $\Pr_l[f|_l$ is far from being low-degree] > $\Omega(\delta)$
- Amount of Randomness Required:
  - [RS '92, ALMSS '92] 2 points – 2 log $|F^m|$
  - [BSVW '03] derandomized set of lines $\approx$ log $|F^m|$

28

Low Degree Polynomials

- Main Tool – Low Degree Polynomial over Finite Fields (Reed-Muller Codes)
  - F - finite field, $f: F^m \to F$, m-variate polynomial over F, deg(f) = maximal degree of monomial in f



- [Schwartz-Zippel]

  If $f \neq g$ have degree < d, then

  $\Pr[f(x) = g(x)] \leq \frac{d}{|F|}$
- Fact:

  If deg(f) < d and l – line, then f restricted to line l is a univariate polynomial of degree < d.

27

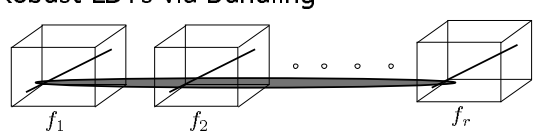
Building Block - Robust-PCPP

- Randomness: $\frac{1}{2}\log n + O(\log\log n)$
- # Queries : $\sqrt{n}\cdot\text{poly}\log n$
- Robustness Parameter
- Proximity Parameter

} constant

30

Robust LDTs via Bundling



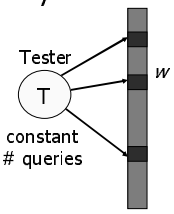
- Each LDT performed separately
  - Possible to cheat by having just one of f_i not low-degree

    ----- NOT ROBUST
- Bundle evaluations of diff. polys. together and perform LDTs in parallel (bundling)
  - PCPP π on query x returns $(f_1(x), f_2(x), \ldots, f_r(x))$

  Robust over larger alphabet F^r
- Can use error-correcting code to make robust over binary alphabet.
  - PCPP π on query x returns $ECC(f_1(x), f_2(x), \ldots, f_r(x))$

29

Locally Testable Codes



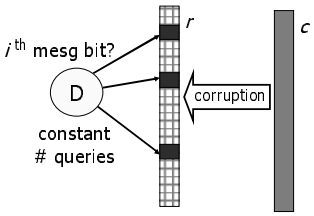
- w – codeword:
  Tester T accepts with probability 1
- w - far from codeword:
  Tester T accepts with low probability

- Lower Bounds
  - [BHR '03] Random LDPC Codes are not LTCs
- LTC Constructions
  - [GS '02, BSVW '03] $k \longrightarrow k \cdot 2^{\sqrt{\log k}}$
  - This paper $k \longrightarrow k \cdot 2^{(\log k)^{\epsilon}}$

Applications to Coding

- Locally Testable Codes
- Relaxed-Locally Decodable Codes

Relaxed Locally Decodable Codes New!



If less than γ n bits corrupted, for "most" message bits i

$\Pr[D^r(i) = m_i] \geq \frac{3}{4}$

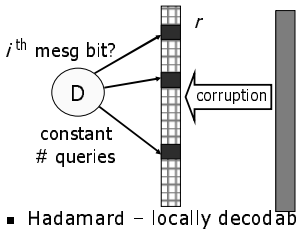
For remaining bits

$\Pr[D^r(i) = ?] \geq \frac{3}{4}$

- This paper: For every $\varepsilon > 0$, there exist relaxed-LDCs with

$$k \longrightarrow k^{1+\epsilon}$$

Locally Decodable Codes



If less than γ n bits corrupted, for all message bits i

$\Pr[D^r(i) = m_i] \geq \frac{3}{4}$

- Hadamard – locally decodable, but poor rate

- Upper Bound: [BIKR '02] $n \leq 2^{O(k)}$
- Lower Bound: [KT '00] $n \geq k^{\Omega(1)}$

The End

Summary of results

- Defined: Robust PCP of proximity
  - Strengthened definition of standard PCPs
- Composition Theorem
  - simple, modular
- Simpler constructions of PCPs
- Coding applications:
  - Locally Testable Codes
  - Relaxed Locally Decodable Codes