CMSC 39600: PCPs, codes and inapproximability	Oct 4, 2007

Lecture 4: $NP \subseteq PCP[\text{poly}, O(1)]$

Lecturer: Prahladh Harsha — *Scribe: Andrew Cotter*

In today's lecture, we will construct exponential sized PCPs for NP. More formally, we will show that the NP-Complete problem, Circuit-Satisfiability (Circuit-SAT) is in $PCP_{1,1-\delta}[O(n^2), O(1)]$ for some $0 < \delta < 1$. Recall that the PCP Theorem states that Circuit-SAT $\in PCP_{1,1-\delta}[O(\log n), O(1)]$. However, as we will see in later lectures, the exponential sized PCPs for Circuit-SAT will be used in proving the PCP Theorem. Actually, we will prove a slightly stronger result than Circuit-SAT $\in PCP_{1,1-\delta}[O(\log n), O(1)]$. We will actually construct a PCP of Proximity for a related problem, Circuit-Value (Circuit-VAL).

4.1 Walsh-Hadamard code (recap from last lecture)

Let us first recall the local testability and decodability of the Wlash-Hadamard code, discussed in the last lecture.

Definition 4.1 (Linearity test)**.** *The* BLR-Test *to test linearity of a function* $f : \{0,1\}^k \to \{0,1\}$ *is as follows:*

BLR-Test^f : " *1. Choose* $y, z \in_R \{0,1\}^k$
2. Accept if $f(y) + f(z) = f(y+z)$. "

Completeness: If f is linear, then $\Pr\left[\text{BLR-Test}^f \text{ accepts}\right] = 1$.

Soundness: If f is δ-far from linear, then $\Pr\left[\text{BLR-Test}^f \text{ rejects}\right] \geq \delta$.

Definition 4.2 (Local decodability)**.** *For any function* $f : \{0,1\}^k \to \{0,1\}$*, that is supposedly closed to some linear function* $l_z = \langle z, \cdot\rangle$*, the (probabilistic) local decoder* Dec^f : $\{0,1\}^k \to \{0,1\}$ *is defined as follows:*

Dec^f : "*On input* x,
1. Choose $r \in_R \{0,1\}^k$
2. Output $f(x+r) - f(r)$ "

Proposition 4.3 (Correctness of Walsh-Hadamard decoder)**.** *If* f *is* δ*-close to the Walsh-Hadamard code* l_z *for some* $z \in \{0,1\}^k$*, then*

$$\Pr\left[\text{Dec}^f(x) = l_z(x) = \langle z, x\rangle\right] \geq 1 - 2\delta.$$

4.2 Circuit-satisfiability is in $PCP[\text{poly}, O(1)]$

4.2.1 Circuit-SAT – Description

The NP-complete problem, Circuit-Satisfiability (Circuit-SAT) is specified as follows: Given a Boolean circuit C with n total gates labeled $1, 2, \ldots, n$ and $k \le n$ input gates (the first k gates are assumed to be input gates), is there an assignment $w \in \{0,1\}^k$ such that $C(w) = 1$. We may assume wlog that the gates of C are one of the following: AND (fan-in 2), NOT, INPUT and OUTPUT (all fan-in 1).

For each gate i (with inputs j, k), we can express the gate constraints as a quadratic function as follows:

$$P_i(z) = \begin{cases} z_i - z_j z_k & \text{if the } i\text{-th gate is an AND gate with inputs from gates } j \text{ and } k. \\ z_i - (1 - z_j) & \text{if the } i\text{-th gate is a NOT gate with input from gate } j. \\ 1 - z_j & \text{if the } i\text{-th gate is an output gate with input from gate } j. \\ 0 & \text{if the } i\text{-th gate is an input gate (i.e. } i \le m). \end{cases}$$

Recall the problem of Circuit-SAT: is there an assignment w to the k input gates such that C is satisfied? Or equivalently, is there an assignment z to all of the gates such that $\forall i, P_i(z) = 0$?

We will now construct a PCP Verifier to check that a given assignment $z \in \{0,1\}^n$ satisfies all the quadratic constraints P_i. To build some intuition, suppose that all the functions P_is are in fact linear. Then, if we want to check that $\forall i P_i(z) = 0$, we could let the proof be the Walsh-Hadamard code l_z for z. Recall that this code is the evaluation of all linear functions at z. Hence, we may verify using the following procedure:

1. Choose $\alpha_1, \alpha_2, \ldots, \alpha_n \in_R \{0,1\}$
2. Accept if $\sum_{i=1}^n \alpha_i P_i(z) = 0$

Of course, we could not do this exactly, as the prover might not give a Walsh-Hadamard code. But then we could check that it is close to a linear function and perform local-decoding. However, we will ignore this issue for now and assume that the proof is exactly the Walsh-Hadamard code of z. Then, clearly we will accept if the circuit is satisfied by the assignment z. Otherwise, we will reject with probability $\frac{1}{2}$.

Unfortunately, we cannot do this as the functions P_i's are not linear, but quadratic functions. We will do essentially the same as above, except that since our functions are quadratic, we must also include the quadratic equivalent of the Walsh-Hadamard code in our proof (i.e., the evaluation of all quadratic functions at z).

4.2.2 Quadratic Evaluations

Definition 4.4 ($quad_x$). *Define* $quad_x : \{0,1\}^{n \times n} \to \{0,1\}$ *as:*

$$quad_x(C) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{i,j}x_i x_j = x^T C x$$

Observe that:

- $quad_x$ is the evaluation of every quadratic function at x
- Given the truth table of $quad_x$, a single probe suffices to evaluate any quadratic function at x
- $quad_x$ is linear (in other words, $quad_x(C_1 + C_2) = quad_x(C_1) + quad_x(C_2)$)

Suppose we are given f and f' which are both linear functions, we first need to check that there exists a z such that $f = l_z$ and $f = quad_z$. We design the following consistency test for this purpose.

Definition 4.5 (Consistency test)**.** *Given both $f : \{0,1\}^n \to \{0,1\}$ (maybe $= l_z$) and $f' : \{0,1\}^{n\times n} \to \{0,1\}$ (maybe $= quad_z$). The consistency test will do the following:*

$\textsf{Quad-Consistency}^{f,f'}$: "*1. Choose $z_1, z_2 \in_R \{0,1\}^n$*
2. Accept if $f'\left(z_1 z_2^T\right) = f(z_1) f(z_2)$ "

where $z_1 z_2^T$ denotes the matrix whose $(i,j)^{th}$ entry is $(z_1)_i \cdot (z_2)_j$ (i.e., the $z_1 z_2^T$ is the outer product of z_1 and z_2).

This consistency test is based on Freivalds's matrix multiplication test [Fre79].

Proposition 4.6 (consistency test)**.** *The consistency test satisfies the following properties*

Completeness: *If $f = l_z$ and $f' = quad_z$ for some $z \in \{0,1\}^k$, then*

$$\Pr\left[\textsf{Quad-Consistency}^{f,f'} \textit{ accepts}\right] = 1.$$

Soundness: *If f and f' are linear functions, and*

$$\Pr\left[\textsf{Quad-Consistency}^{f,f'} \textit{ accepts}\right] > \frac{3}{4},$$

then there exists z such that $f = l_z$ and $f' = quad_z$.

Proof. The completeness follows from the following observations. Suppose $f = l_z$ and $f' = quad_z$ for some $z \in \{0,1\}^k$. Then, for any y, we have $f(y) = y^T z$. Furthermore,

$$\begin{aligned}
f'\left(z_1 z_2^T\right) &= \sum_{i,j} z_i z_j \left[z_1 z_2^T\right]_{i,j} \\
&= \sum_{i,j} z_i z_j [z_1]_i [z_2]_j \\
&= \left(\sum_i z_i [z_1]_i\right) \cdot \left(\sum_j z_j [z_2]_j\right) \\
&= f(z_1) \cdot f(z_2)
\end{aligned}$$

Now to prove soundness. Since we know f is linear, we know that $f = l_z$ for some z. Hence, $f(z_1) \cdot f(z_2) = (z_1^T z) \cdot (z^T z_2) = z_1^T (zz^T) z_2$. Furthermore, since f' is linear, there exists a matrix $B = \{b_{ij}\}$ such that $f'(z_1 z_2^T) = \sum b_{ij}(z_1)_i (z_2)_j = z_1^T B z_2$. We will now compare the matrices B and $C = zz^T$. If $B = C$, then it must be the case that $f' = quad_z$. Otherwise, we will show that the test rejects with probability at least $1/4$, thus proving soundness.

If $B \neq C$, then for a random vector $z_2 \in \{0,1\}^k$, we have $Bz_2 \neq Cz_2$ with probability at least $1/2$, i.e., $\Pr_{z_2}[Bz_2 \neq Cz_2 \mid B \neq C] \geq 1/2$. Furthermore if $y_1 \neq y_2 \in \{0,1\}^k$ then for a random z_1, we have $z_1^t y_1 \neq z_1^t y_2$ with probability exactly $1/2$. Setting $y_1 = Bz_2$ and $y_2 = Cz_2$, we have $\Pr_{z_1}\left[z_1^T B z_2 \neq z_1^T C z_2 \mid Bz_2 \neq Cz_2\right] = 1/2$. Putting both together we have $\Pr_{z_1,z_2}\left[z_1^T B z_2 \neq z_1^T C z_2 \mid B \neq C\right] \geq 1/4$. This completes the proof of soundness

□

The above consistency test checks that f and f' are the l_z and $quad_z$ respectively given that f and f' are both linear functions. However, we cannot guarantee that f and f' are linear functions, all we can guarantee that f and f' are close to linear functions using the linearity test. We then implement the following self-corrected version of the above quadratic-consistency test.

Definition 4.7 (Quadratic correction test)**.** *Given both $f : \{0,1\}^n \to \{0,1\}$ (maybe $= l_z$) and $f' : \{0,1\}^{n\times n} \to \{0,1\}$ (maybe $= quad_z$). The quadratic correction test is as follows:*

$\mathsf{Quad\text{-}Correction}^{f,f'}$: "*1. Choose $z_1, z_2 \in_R \{0,1\}^n$ and $M \in_R \{0,1\}^{n\times n}$*
2. Accept if $f'\left(z_1 z_2^T + M\right) - f'(M) = f(z_1)\, f(z_2)$ "

Proposition 4.8 (correction test)**.** *The quadratic correction test satisfies the following properties*

Completeness: *If $f = l_z$ and $f' = quad_z$ for some $z \in \{0,1\}^k$, then*

$$\Pr\left[\mathsf{Quad\text{-}Correction}^{f,f'} \text{ accepts}\right] = 1.$$

Soundness: *If f is δ-close to l_z for some $z \in \{0,1\}^n$ and f' is δ-close to some linear function, and*

$$\Pr\left[\mathsf{Quad\text{-}Correction}^{f,f'} \text{ accepts}\right] > \frac{3}{4} + 4\delta,$$

then in fact f' is δ-close to $quad_z$.

Proof. The completeness is obvious. For the soundness, let g be the linear function that is δ-close to f'. Since $z_1, z_2, z_1 z_2^T + M$ and M are uniformly random elements (but not independent) in their respective domains, we have that with probability at most 4δ, either $f(z_1) \neq l_z(z_1)$ or $f(z_2) \neq l_z(z_2)$ or $f'(z_1 z_2^T + M) \neq g(z_1 z_2^T + M)$ or $f'(M) \neq g(M)$. Since we have that $f(z_1) \cdot f(z_2) = f'(z_1 z_2^T + M) - f'(M)$ with probability at least $2/4 + 4\delta$, it must be the case that $l_z(z_1) \cdot l_z(z_2) = g(z_1 z_2^T + M) - g(M)$ with probability at least $3/4$. It now follows from the soundness of $\mathsf{Quad\text{-}Consistency}$ that $g = quad_z$. Thus, proved. □

4.2.3 PCP verifier

We are now ready to describe the PCP verifier for Circuit-SAT. The PCP Verifier expects as proof the functions $f : \{0,1\}^n \to \{0,1\}$ and $f' : \{0,1\}^{n\times n} \to \{0,1\}$ which are supposedly l_z and $quad_z$ where z is some satisfying assignment to the gates of the circuit C. The actions of the PCP verifier, which must be evident by now, are summarized below.

Definition 4.9. *PCP verifier for* Circuit-SAT *The proof given to the verifier will consist of functions $f : \{0,1\}^n \to \{0,1\}$ (maybe $= l_z$) and $f' : \{0,1\}^{n\times n} \to \{0,1\}$ (maybe $= quad_z$). On input a circuit C and oracle access to these functions, the verifier will perform the following steps:*

1. *Linearity test (for both f and f')*
   (a) *Choose $z_1, z_2 \in_R \{0,1\}^n$ and $M_1, M_2 \in_R \{0,1\}^{n\times n}$*
   (b) *Check that $f(z_1 + z_2) = f(z_1) + f(z_2)$ and $f'(M_1 + M_2) = f'(M_1) + f'(M_2)$*
2. *Quadratic correlation test*
   (a) *Choose $z_1, z_2 \in_R \{0,1\}$, and $M \in_R \{0,1\}^{n\times n}$*
   (b) *Check that $f(z_1) \cdot f(z_2) = f'\left(z_1 z_2^T + M\right) - f'(M)$*
3. *Circuit test*
   (a) *Choose $\alpha_1, \alpha_2, \ldots, \alpha_n \in_R \{0,1\}$, $z' \in_R \{0,1\}^n$ and $M \in_R \{0,1\}^{n\times n}$*
   (b) *Decompose the function $P(z) = \sum_{i=1}^{n} \alpha_i P_i(z)$ as the sum of a quadratic part $Q(z) = z^T B z$, a linear part $L(z) = y^T z$, and a constant c*
   (c) *Check that $[f'(M + B) - f(M)] + [f(y + z') - f(z')] + c = 0$*

We can now perform a analysis of PCP Verifier as follows:
Query Complexity: The number of queries is $14 = 6 + 4 + 4$ (linearity + correlation + circuit).
Randomness: The number of random coins tossed by the verifier is $\left(2n + 2n^2\right) + \left(2n + n^2\right) + \left(2n^2\right) = 6n + 4n^2 = O(n^2)$.
Proof Length: The length of the proof given to the verifier is $2^n + 2^{n^2}$.
Completeness: Clearly, if $f = l_z$, $f' = quad_z$ where z is a satisfying assignment for C, then the verifier accepts with probability exactly 1.

Claim 4.10 (Soundness)**.** *There exists a $\delta_0 > 0$ such that for all $\delta \leq \delta_0$, if the PCP Verifier accepts with probability at least $1 - \delta$, then there exists a satisfying assignment z for C, f is δ-close to l_z, and f' is δ-close to $quad_z$.*

Proof. This will be a proof by contradiction. Set $\delta_0 = 1/20$. Suppose the claim is false for this setting of δ_0. Then there exists a $\delta < \delta_0 = 1/20$ such that $\Pr[\text{ PCP Verifier accepts }] > 1 - \delta$, but there does not exist a satisfying assignment z such that f is δ-close to l_z and f' is δ-close to $quad_z$. Then, at least one of the following four possibilities must hold.

1. f is δ-far from linear

2. f' is δ-far from linear

3. f is δ-close to some l_z, and f' is δ-close to some linear g, but $g \neq quad_z$

4. f is δ-close to some l_z, f' is δ-close to $quad_z$ for some z, but z is not a satisfying assignment for C (equivalently, $\exists i P_i(z) \neq 0$)

We will now show that none of the above cases can happen. In cases 1 and 2, the verifier rejects with probability at least δ by the soundness of the linearity test. In case 3, the verifier rejects with probability at least $1/4 - 4\delta$ by the soundness of the Quad-correction test. In case 4, it must be the case that at least one of the polynomials $P_i(z) \neq 0$. Hence, with probability exactly $1/2$, $\sum \alpha_i P_i(z) \neq 0$. But with probability at at most 2δ, $f'(M+B) - f'(M) \neq z^T BZ$ and similarly withe probability at most 2δ, $f(y+z') - f(y) \neq l_z(y)$. Hence, with probability at least $1/2 - 4\delta$, $[f'(M+B) - f(M)] + [f(y+z') - f(z')] + c = \sum \alpha_i P_i(z) \neq 0$. Thus, the circuit test rejects with probability at least $1/2 - 4\delta$.

For δ sufficiently small ($\delta < 1/20$), both $\frac{1}{4} - 4\delta$ and $\frac{1}{2} - 4\delta$ are no smaller than δ. Hence, in each of the cases the verifier rejects with probability at least δ. This is a contradiction.

□

In essence, we have constructed an exponential sized PCP for Circuit-SAT.

Theorem 4.11. *[ALM+98]*

$$\textsf{Circuit-SAT} \in PCP_{1,1-\frac{1}{20}}[O(n^2), 14].$$

This theorem and the proof is due to Arora, Lund, Motwani, Sudan and Szegedy [ALM+98].

4.2.4 PCP of Proximity

In this section, we will show that the above construction of Arora et al. actually can be used to prove a slightly stronger statement, namely that Circuit-Value (Circuit-VAL) has a PCP of Proximity of exponential length. we will not explicitly define what a PCP of proximity is in this lecture, however we will prove this stronger statement.

The PCP verifier given above verifies the existence of a satisfying assignment z for the circuit C. The assignment w itself, however, only enters into the proof in l_z and $quad_z$. If we wished in addition to verify that some particular assignment w (to the input gates only), given implicitly via oracle access to the verifier, is close to a satisfying assignment of the circuit C, we could do the following additional test. Informally speaking, such a verifier that checks that checks that w is in the proximity of a satisfying assignment is called a "PCP of proximity (PCPP)" verifier.

Definition 4.12. *PCPP Verifier The verifier is given to inputs – the explicit input, the circuit C, which it can read in its entirety and the implicit input w, which it has oracle access to. As before the proof oracles are $f : \{0,1\}^n \to \{0,1\}$ and $f' : \{0,1\}^{n \times n} \to \{0,1\}$. The PCPP verifier performs all of the tests of the PCP verifier and the following additional test*

4. Proximity test

(a) *Choose* $i \in_R \{1, 2, \ldots, k\}$, $z' \in_R \{0,1\}^n$

(b) *Check that* $w_i = f(z' + e_i) - f(z')$

where by e_i *we refer to the axis-parallel unit-vector* $(0, \ldots, 0, 1, 0, \ldots, 0)$ *that has a* 1 *in its* i*-th co-ordinate and* 0 *elsewhere.*

The proof length is as before, the number of random coins increase by $\log_2(k) + n$, while the PCPP verifier makes three more queries than the PCP verifier resulting in a total of 17 queries.

Claim 4.13 (Completeness)**.** *If* $f = l_z$, $f' = quad_z$ *and* z *is a satisfying assignment for* C *which is equal to* w *in its first* k *bits, then the PCPP verifier accepts with probability exactly 1.*

Claim 4.14 (Soundness)**.** *There exists a* $\delta_0 > 0$ *such that for all* $\delta \le \delta_0$*, if the PCPP verifier accepts with probability at least* $1 - \delta$*, then there exists a satisfying assignment* z *for* C *which is equal to* w' *in its first* k *bits,* f *is* δ*-close to* l_z*, and* f' *is* δ*-close to* $quad_z$*, and furthermore* w' *is* 3δ*-close to* w*.*

Proof. Since the PCPP verifier accepts with probability at least $1 - \delta$, it follows from the soundness of the PCP Verifier that there there exists a satisfying assignment z for C, f is δ-close to l_z, and f' is δ-close to $quad_z$. We now analyze the proximity test as follows: Since z' is a random element of $\{0,1\}^n$, we have

- With probability at least $1 - \delta$, $f(z') = l_z(z')$
- With probability at least $1 - \delta$, $f(z' + e_i) = l_z(z' + e_i)$

Hence, with probability at least $1 - 2\delta$, $f(z' + e_i) - f(z') = l_z(z' + e_i) - l_z(z') = l_z(e_i) = z_i = (w')_i$. Since we also have that $f(z' + e_i) - f(z') = w_i$ with probability $1 - \delta$ (by the proximity test), we see that $w_i = (w')_i$ with probability at least $1 - 3\delta$. Hence, $\delta(w, w') \le 3\delta$. ∎

The notion of a PCP of proximity has appeared in several guises in the PCP literature, though it was formally defined independently by Ben-Sasson et al [BGH+06] and Dinur and Reingold [DR06]. The above construction appears in the work of Ben-Sasson et al [BGH+06].

References

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. *Proof verification and the hardness of approximation problems.* J. ACM, 45(3):501–555, May 1998. (Preliminary Version in *33rd FOCS*, 1992). doi:10.1145/278298.278306.

[BGH+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. *Robust PCPs of proximity, shorter PCPs and applications to coding.* SIAM J. Computing, 36(4):889–974, 2006. (Preliminary Version in *36th STOC*, 2004). doi:10.1137/S0097539705446810.

[DR06] Irit Dinur and Omer Reingold. *Assignment testers: Towards a combinatorial proof of the PCP Theorem*. SIAM J. Computing, 36:975–1024, 2006. (Preliminary Version in *45th FOCS*, 2004). doi:10.1137/S0097539705446962.

[Fre79] Rusins Freivalds. *Fast probabilistic algorithms*. In Jirí Becvár, ed., *Proc. 8th Symposium of Mathematical Foundations of Computer Science*, volume 74 of *Lecture Notes in Computer Science*, pages 57–69. Springer, Olomouc, Czechoslovakia, 3–7 September 1979. doi:10.1007/3-540-09526-8_5.