# Query Efficient PCPs with Perfect Completeness

Johan Håstad
Royal Institute of Technology, Stockholm
johanh@nada.kth.se *

Subhash Khot
Princeton University, NJ
khot@cs.princeton.edu †

## Abstract

*For every integer $k \geq 1$, we present a PCP characterization of NP where the verifier uses logarithmic randomness, queries $4k + k^2$ bits in the proof, accepts a correct proof with probability 1 (i.e. it is has perfect completeness) and accepts any supposed proof of a false statement with probability at most $2^{-k^2+1}$. In particular, the verifier achieves optimal amortized query complexity of $1 + \delta$ for arbitrarily small constant $\delta > 0$. Such a characterization was already proved by Samorodnitsky and Trevisan [15], but their verifier loses perfect completeness and their proof makes an essential use of this feature.*

*By using an adaptive verifier we can decrease the number of query bits to $2k + k^2$, the same number obtained in [15]. Finally we extend some of the results to larger domains.*

## 1. Introduction

The discovery of the PCP Theorem ([2],[1]) opened up a whole new fascinating direction for proving various inapproximability results. In the last eight years, quantitative improvement in the efficiency of PCP verifiers has led to (in many cases optimal) hardness results for many optimization problems ([3], [5], [9], [10], [15], [7]). For different applications, different aspects of the given PCP need to be optimized. For a detailed discussion of various parameters we refer to [3].

In the current paper we are mostly concerned with making efficient use of queries, i.e. to obtain very strong proofs where the verifier reads very few symbols in the proof.

Samorodnitsky and Trevisan [15] obtained very strong results along these lines, giving a PCP where the verifier reads $2k+k^2$ bits, almost always accepts a correct proof of a correct theorem and accepts a proof of an incorrect theorem with probability only marginally larger than $2^{-k^2}$. This is a very impressive result in that each read bit essentially decreases the probability of being fooled by a factor of 2. Formally, one defines *amortized query compleixty* as $\frac{q}{\log_2(1/s)}$ where $q$ is the number of query bits and $s$ is the probability of accepting a proof of an incorrect theorem. Samorodnitsky and Trevisan's verifier achieves amortized query complexity arbitrarily close to 1 which is optimal since PCPs with amortized query complexity less than 1 can recognize languages only in BPP (see [3]).

The fact that the verifier sometimes rejects a correct proof of a correct theorem is called imperfect completeness and in their construction Samorodnitsky and Trevisan make essential use of this property of the verifier. Their verifier accepts if the query bits satisfy certain linear constraints. Such a PCP cannot have perfect completeness (unless P=NP) since one can test the satisfiability of a system of linear constraints in polynomial time.

For many reasons it is preferable to have perfect completeness. Firstly it is natural to have a proof system where a correct proof of a correct theorem is always accepted. Secondly, perfect completeness is sometimes essential to obtain further results. In-approximability results for some problems such as coloring often make essential use of perfect completeness and when using a given protocol as a subprotocol in future protocols, perfect completeness, to say the least, simplifies matters.

Several results in the past have focused on achieving PCPs with perfect completeness and this task many times turns out to be much harder than obtaining corresponding PCPs without this property. For instance, Håstad shows that 3SAT is hard to approximate within ratio $\frac{8}{7} + \epsilon$. This result follows from the basic 3-bit PCP of [10] establishing hardness for approximating the number of satisfied linear equations mod 2. To extend this result to satisfiable instances requires a new construction and a technically much more complicated proof.

The main result of the current paper is to extend the result of Samorodnitsky and Trevisan to include perfect completeness.

*Work done while visiting Institute for Advanced Study, supported by NSF grant CCR-9987077.
†Work partially supported by S. Arora's David and Lucile Packard Fellowship.

**Theorem 1.1** *For every $k \geq 1$, NP has a PCP verifier that queries $4k + k^2$ bits, has perfect completeness and accepts a proof of an incorrect theorem with probability at most $2^{-k^2+1}$.*

Our result is based on a basic non-linear test which reads 5 bits $(b_1, b_2, b_3, b_4, b_5)$ from the proof and accepts if $b_1 = b_2 \oplus b_3 \oplus (b_4 \wedge b_5)$. We call this constraint Tri-Sum-And and let MAX-TSA be the problem of satisfying maximum number of such constraints. We have the following theorem.

**Theorem 1.2** *For any $\epsilon > 0$, it is NP-hard to distinguish satisfiable instances of Max-TSA from those where you can only satisfy a fraction $\frac{1}{2} + \epsilon$ of the constraints.*

Note that this is tight for Max-TSA in that a random assignment satisfies half the constraints. There are stronger results for other constraints on 5 bits and in particular Guruswami et al. [8] give a different predicate for which $\frac{1}{2}$ can be improved to $\frac{7}{16}$.

We then iterate this basic test in a way similar to Samorodnitsky and Trevisan iterate the basic 3-bit test by Håstad. We present two iterated tests : One which we call the "complete bipartite graph test", can be analyzed in a way analogous to Samorodnitsky-Trevisan and the other, which we call the "almost disjoint sets test", can be analyzed in a way analogous to how Håstad and Wigderson [11] analyzed the test of Samorodnitsky and Trevisan. We present the proof of only the second construction in this paper.

By a standard reduction the PCP results imply the following theorem.

**Theorem 1.3** *For all sufficiently large $k$, Boolean constraint satisfaction problem on $k$ variables is hard to approximate within ratio $2^{k-O(\sqrt{k})}$ on satisfiable instances.*

A test is called non-adaptive if which bits to read are decided before the first bit is read and hence this set is independent of the actual proof. All the above mentioned tests are non-adaptive which is in fact necessary to obtain Theorem 1.3.

If we allow adaptive tests then by making an iterated version of a test in [8] we can get essentially the same parameters as Samorodnitsky and Trevisan and simply gain perfect completeness.

**Theorem 1.4** *For every $k \geq 1$, NP has an adaptive PCP verifier that queries $2k + k^2$ bits, has perfect completeness and accepts a proof of an incorrect theorem with probability at most $2^{-k^2+1}$.*

This test might read $2k + 2k^2$ different bits and hence this result does not strictly dominate Theorem 1.1.

We extend some of our results to non-Boolean domains and in particular we have the following theorem.

**Theorem 1.5** *For every prime $p$, constraint satisfaction problem on $k$ variables over an alphabet of size $p$ is hard to approximate within ratio $p^{k-O(\sqrt{k})}$ on satisfiable instances.*

We hope that our results will be useful in future to prove strong hardness results for approximate graph coloring. One such result by Khot [12] is

**Theorem 1.6** *[12] For sufficiently large constants $k$, it is NP-hard to color a $k$-colorable graph with $k^{\Omega(\log k)}$ colors.*

Actually this result can be proved from the original form of the Samorodnitsky-Trevisan's result and perfect completeness is not strictly required. But using our PCP with perfect completeness, this result becomes more straightforward. On a related note one can observe that perfect completeness is essential in the hypergraph coloring results by Guruswami, Håstad and Sudan [7], and in general it is a subtle problem which coloring inapproximability results require perfect completeness in the underlying PCP.

Let us give a brief overview of the paper. Section 2 introduces techniques used in this paper. In Section 3 we give our results for the Boolean case: Section 3.1 gives our basic 5-bit test, and Section 3.2 gives our iterated tests. Appendix A extends some of the results of Section 3 to non-Boolean domains. Section 4 concludes with a few remarks.

## 2. The General Setup

In this section we provide the necessary background.

### 2.1. Notation

Throughout the paper, we have Boolean functions in $\pm 1$ notation with $-1$ as logical true. We use multiplication to denote exclusive-or, $\wedge$ for the logical AND function (which is *not* multiplication) while addition is used only over the real and complex numbers.

### 2.2. The 2-prover Protocol

Many efficient PCPs, such as the one given in [15] are conveniently analyzed using the formalism of an outer and inner verifier. This could also be done here, but to avoid too much formalism we give a more explicit analysis. Using the results of [1] (as explicitly done in [5]) one can prove that there is a constant $c < 1$ such that it is NP-hard to distinguish satisfiable 3-SAT formulas from those where only a fraction $c$ of the clauses can be satisfied by any assignment. This formula can furthermore have the property that any clause is of length exactly 3 and any variable appears in exactly 5 clauses.

Given a 3-SAT formula $\varphi = C_1 \wedge C_2 \ldots \wedge C_m$ which is either satisfiable or where one can only satisfy a fraction $c$

of the clauses one can design a two-prover interactive proof system with verifier $V$ as follows.

**Basic two-prover protocol**

1. $V$ chooses a clause $C_k$ uniformly at random and a variable $x_j$, uniformly at random from the 3 variables appearing in $C_k$. $V$ sends $k$ to prover $P_1$ and $j$ to prover $P_2$.

2. $V$ receives a value for $x_j$ from $P_2$ and values for all variables appearing in $C_k$ from $P_1$. $V$ accepts if the two values for $x_j$ agree and the clause $C_k$ is satisfied.

It is not difficult to see that if a fraction $c$ of the clauses can be satisfied simultaneously then the optimal strategy of $P_1$ and $P_2$ convinces $V$ with probability $(2+c)/3$. Thus it is NP-hard to distinguish the case when this probability is 1 and when it is some constant strictly smaller than 1.

To make the gap larger one runs this protocol $u$ times in parallel and in this protocol $u$ random clauses are sent to $P_1$, $u$ variables (one from each clause) are sent to $P_2$. The verifier accepts in this protocol if the assignments returned by the provers satisfy all the picked clauses and are consistent. By the fundamental result by Raz [14], the probability that the verifier accepts when only a constant fraction $c<1$ of the clauses can be simultaneously satisfied is bounded by $d_c^u$ for some absolute constant $d_c < 1$. Let us formulate these properties for future reference.

**Theorem 2.1** *Let 2PP$(u)$ be the $u$ parallel version of the basic two-prover protocol. Then if only a fraction $c<1$ of the clauses of $\varphi$ can be simultaneously satisfied, then no strategy of $P_1$ and $P_2$ can make the verifier accept with probability greater than $d_c^u$. Here $d_c<1$ is a constant that only depends on $c$.*

## 2.3. Long Codes

To turn the protocol 2PP$(u)$ into a written proof that can be checked very efficiently it is natural to, for each question to either $P_1$ or $P_2$, write down the answer in coded form. As many other papers we use the *long code* introduced by Bellare et al [3].

**Definition 2.2** *The* long code *of an assignment $x \in \{-1,1\}^t$ is obtained by for each function $f : \{-1,1\}^t \mapsto \{-1,1\}$ writing down the value $f(x)$.*

Thus the long code of a string of length $t$ is a string of length $2^{2^t}$. Note that even though a prover is supposed to write down a long code for an assignment we have no way to guarantee that a cheating prover does not write down a string which is not the correct long code of anything. We analyze such arbitrary tables by the Fourier-expansion and in the current situation this is given by

$$A = \sum_{\alpha \subseteq \{-1,1\}^t} \hat{A}_\alpha \chi_\alpha(f),$$

where

$$\chi_\alpha(f) = \prod_{x\in\alpha} f(x) \qquad \text{and} \qquad \widehat{A}_\alpha = E_f[A(f)\chi_\alpha(f)]$$

If $A$ is indeed a correct long code of a string $x^{(0)}$ then $\hat{A}_{\{x^{(0)}\}} = 1$ while all the other Fourier coefficients are 0. The Fourier coefficients of any table $A$ satisfy the Parseval's identity, i.e. $\sum_\alpha \widehat{A}_\alpha^2 = 1$.

We can, to a limited extent, put some restrictions on the tables produced by the prover. The following definitions are really motivated by their consequences which we state in Lemmas 2.5 and 2.6.

**Definition 2.3** *A supposed long code $A$ is folded over true if $A(f) = -A(-f)$ for any $f$.*

**Definition 2.4** *A supposed long code $A$ is conditioned upon a function $h : \{-1,1\}^t \mapsto \{-1,1\}$, if $A(f) = A(f \wedge h)$ for any $f$.*

As mentioned before, we now give the consequences of folding and conditioning for the Fourier coefficients. The proofs are easy and left to the reader but they can also be found in [9].

**Lemma 2.5** *If $A$ is folded over true and $\hat{A}_\alpha \neq 0$ then $|\alpha|$ is odd and in particular $\alpha$ is non-empty.*

**Lemma 2.6** *If $A$ is conditioned upon $h$ and $\hat{A}_\alpha \neq 0$ then for every $x \in \alpha$, $h(x)$ is true.*

To make sure that an arbitrary long code is folded we access the table as follows. For each pair $(f,-f)$ we choose (in some arbitrary but fixed way) one representative. If $f$ is chosen, then if the value of the table is required at $f$ it is accessed the normal way by reading $A(f)$. If the value at $-f$ is required then in this case also $A(f)$ is read but the result is negated. If $-f$ is chosen from the pair the procedures are reversed.

Similarly we can make sure that a given table is properly conditioned by always reading $A(f \wedge h)$ when the value for $f$ is needed. Folding over true and conditioning can be done at the same time.

We now describe efficient PCPs that follow the basic setup of the two-prover protocol but replace the final check by some checks on the long codes of the corresponding answers.

## 3. Efficient PCPs for Boolean Domains

In this section let $U$ be a random set of $u$ variables which would have been sent to $P_2$ in the two-prover protocol. Let $A$ be a supposed long code on $U$ in a written proof. We assume that $A$ is folded over true.

Similarly let $W$ be the set of variables in the $u$ clauses sent to $P_1$ and let $\varphi_W$ be the conjunction of the clauses chosen. We let $B$ be a supposed long code on $W$ in a written proof. We assume that $B$ is folded over true and conditioned upon $\varphi_W$.

Note that $U$ is a subset of $W$ and we let $\pi$ denote the projection operator. This operator is defined for assignments by taking subassignments and it is naturally extended to sets of assignments by taking the projection of all the elements of a given set. For sets we also need the operation $\pi_2$ and $x \in \pi_2(\beta)$ iff there is an odd number of elements in $\beta$ that project onto $x$.

### 3.1. Our Basic Test

We have the following basic test, defined using the conventions above.

1. $V$ chooses two random functions $f$ and $f'$ on $U$.
2. $V$ chooses two random functions $g$ and $g'$ on $W$ and defines a third function $h$ by setting for each $y \in \{-1,1\}^W$, $h(y) = g(y)f(\pi(y))(g'(y) \wedge f'(\pi(y)))$.
3. $V$ accepts iff $B(h) = B(g)A(f)(B(g') \wedge A(f'))$.

We have the basic completeness lemma.

**Lemma 3.1** *The completeness of the basic test is 1.*

**Proof:** In a correct proof of a correct theorem each value of a long code is the evaluation of a given global assignment. If we denote this assignment by $z$ then $B(h) = h(\pi^W(z))$ where $\pi^W$ is the projection onto $W$ and similarly for the other involved functions. The completeness now follows from the definition of $h$. ∎

The main problem is to establish soundness.

**Lemma 3.2** *If the verifier in the basic test accepts with probability $(1+\delta)/2$ then there exists a strategy for $P_1$ and $P_2$ in 2PP($u$) that makes the verifier accept with probability $\delta^{O(1)}$.*

**Proof:** The hypothesis of the lemma implies that

$$E_{f,f',g,g',U,W}[B(h)B(g)A(f)(B(g') \wedge A(f'))] = \delta. \quad (1)$$

Fix $U, W, f'$ and $g'$ and let us study

$$E_{f,g}[B(h)B(g)A(f)].$$

Replacing each function by its Fourier expansion we see that this equals

$$\sum_{\beta_1,\beta_2,\alpha} \hat{B}_{\beta_1}\hat{B}_{\beta_2}\hat{A}_\alpha E_{f,g}[\chi_{\beta_1}(fg(f' \wedge g'))\chi_{\beta_2}(g)\chi_\alpha(f)].$$

The inner expected value is 0 unless $\beta_1 = \beta_2 = \beta$ and $\pi_2(\beta) = \alpha$ and thus the expected value equals

$$\sum_\beta \hat{B}_\beta^2 \hat{A}_{\pi_2(\beta)}\chi_\beta(f' \wedge g').$$

Hence we need to analyze

$$E_{f',g'}[\chi_\beta(f' \wedge g')(B(g') \wedge A(f'))].$$

We have $a \wedge b = \frac{1}{2}(1+a+b-ab)$ and using this we should analyze

$$E[\chi_\beta(f' \wedge g')], E[\chi_\beta(f' \wedge g')B(g')], E[\chi_\beta(f' \wedge g')A(f')]$$

and $E\,[\chi_\beta(f' \wedge g')B(g')A(f')]$.

Fix the value of $f'$ and let

$$\beta' = \{y \mid y \in \beta \ \wedge \ f'(\pi(y)) = -1\}.$$

When averaging over $g'$, the first and third expected values are 0 unless $\beta' = \emptyset$ while the second and the fourth expected values equal $\hat{B}_{\beta'}$ and $\hat{B}_{\beta'}A(f')$, respectively. The probability, over the choice of $f'$ that $\beta'$ is empty is $2^{-|\pi(\beta)|}$ while Cauchy-Schwartz inequality implies

$$E_{f'}\left[|\hat{B}_{\beta'}|\right] = 2^{-|\pi(\beta)|}\sum_{\alpha \subseteq \pi(\beta)} |\hat{B}_{\beta \cap \pi^{-1}(\alpha)}|$$

$$\leq 2^{-|\pi(\beta)|/2}\left(\sum_{\alpha \subseteq \pi(\beta)} \hat{B}^2_{\beta \cap \pi^{-1}(\alpha)}\right)^{1/2} \leq 2^{-|\pi(\beta)|/2}. \quad (2)$$

This implies that we get an overall upper bound on the left hand side of (1) as

$$E_{U,W}\left[\sum_\beta \hat{B}_\beta^2 |\hat{A}_{\pi_2(\beta)}|(2^{-|\pi(\beta)|} + 2^{-|\pi(\beta)|/2})\right]$$

$$\leq E_{U,W}\left[\sum_\beta \hat{B}_\beta^2 |\hat{A}_{\pi_2(\beta)}| 2^{1-|\pi(\beta)|/2}\right], \quad (3)$$

and hence this expression is at least $\delta$. We use this to establish a good strategy for $P_1$ and $P_2$. We first establish that some parts of the given sum are small. We have the following lemma from [9].

**Lemma 3.3** *There is a constant $c > 0$ such that $E_U[|\pi(\beta)|^{-1}] \leq |\beta|^{-c}$. The value $c = \frac{1}{35}$ is acceptable.*

Let $S_\delta = (4(6+2\log\delta^{-1})/\delta)^{1/c}$ and consider any $\beta$ of size at least $S_\delta$. Since $E[|\pi(\beta)|^{-1}] \le \delta/4(6+2\log\delta^{-1})$, we conclude that the probability that $|\pi(\beta)| \le (6+2\log\delta^{-1})$ is bounded by $\delta/4$. Thus for any such $\beta$ we have

$$E_U[2^{1-|\pi(\beta)|/2}] \le \frac{\delta}{4} + \frac{\delta}{4} = \frac{\delta}{2}$$

and hence discarding terms with $\beta$ of size at least $S_\delta$ in (3) still keeps a sum of expected value at least $\delta/2$.

Furthermore since $\sum_\beta \hat{B}_\beta^2 = 1$ we can discard any term with $|\hat{A}_{\pi_2(\beta)}| \le \delta/4$ and not reduce the sum by more than $\delta/4$. We conclude that the sum which is the right hand side of (3) is at least $\delta/4$ even if we restrict summation to $\beta$ of size at most $S_\delta$ and such that $|\hat{A}_{\pi_2(\beta)}| \ge \delta/4$.

Now consider the following strategy for the provers $P_1$ and $P_2$. On receiving $W$, $P_1$ chooses $\beta$ with probability $\hat{B}_\beta^2$ and returns a random $y \in \beta$. Similarly on receiving a $U$, $P_2$ chooses $\alpha$ with probability $\hat{A}_\alpha^2$ and returns a random $x \in \alpha$. We note that since $A, B$ are folded over true, by Lemma 2.5, the sets $\alpha$ and $\beta$ selected by the provers are always nonempty. Also, since $B$ is conditioned upon $\varphi_W$, by Lemma 2.6, every $y \in \beta$ satisfies the formula $\varphi_W$. The success-probability of the given strategy is at least

$$E_{U,W}[\sum_\beta \hat{B}_\beta^2 \hat{A}_{\pi_2(\beta)}^2 |\beta|^{-1}] \qquad (4)$$

If we restrict summation to $|\beta| \le S_\delta$ and $|\hat{A}_{\pi_2(\beta)}| \ge \delta/4$, (4) is at least

$$S_\delta^{-1}\delta/4 \; E_{U,W}[\sum_{\beta;|\beta|\le S_\delta, |\hat{A}_{\pi_2(\beta)}|\ge\delta/4} \hat{B}_\beta^2 |\hat{A}_{\pi_2(\beta)}|]$$

and by the above reasoning this expected value is at least $\delta/4$ and we get a lower bound $S_\delta^{-1}(\delta/4)^2$ for the success probability of the provers. This completes the proof of the lemma. ∎

The basic test reads 5 bits $(b_1, b_2, b_3, b_4, b_5)$ of the proof and checks whether $b_1 b_2 b_3 (b_4 \wedge b_5) = 1$ which is same as $b_1 = b_2 \oplus b_3 \oplus (b_4 \wedge b_5)$ in $\{0,1\}$ notation. Theorem 1.2 now follows by a standard procedure of replacing the bits in the proof by variables and asking for a proof that maximizes the acceptance probability.

## 3.2. Iterated Tests

We now extend our basic test in a query efficient way. We pick one set $U$ and on it we pick $k$ functions $(f_i)_{i=1}^k$ and $k$ functions $(f'_j)_{j=1}^k$ and $k$ sets $(W_l)_{l=1}^k$ each with its pair of functions $(g_l, g'_l)$. We perform the basic test for a certain set of quadruples $(f_i, f'_j, g_l, g'_l)$. We can give a strong analysis in two cases each utilizing $k^2$ quadruples. One is given by the constraint $i = j$ and can be analyzed very much as Samorodnitsky and Trevisan [15] analyzed their tests. We call it the “complete bipartite test”. We omit the analysis of this test.

The other set of $k^2$ quadruples is given by all triples $(i,j,l)$ such that $i+j+l = 0 \bmod k$. The key property of this set of triples is that any two different triples have at most one coordinate in common. Hence we call it the “almost disjoint sets test”. This analysis can be done in the style of Håstad and Wigderson [11] and is substantially simpler. We present a complete analysis of this test.

In either case we get a test that reads $4k + k^2$ bits, has perfect completeness and soundness only marginally higher than $2^{-k^2}$, proving Theorem 1.1. This theorem follows from either Theorem 3.4 or Theorem 3.5 below.

### 3.2.1 The Almost Disjoint Sets Test

We first define the test which is an iteration of the basic test studied in the last section. Define

$$h_{ijl} = g_l f_i (f'_j \wedge g'_l)$$

and check that

$$B_l(h_{ijl}) = B_l(g_l) A(f_i) (A(f'_j) \wedge B_l(g'_l))$$

for all $i,j,l$ with $i+j+l \equiv 0 \bmod k$. Let $Z_0$ be this set of triples. Let us call this the “Almost disjoint sets test”. We have the following theorem.

**Theorem 3.4** *The almost disjoint sets test has completeness* $1$ *and soundness* $2^{-k^2} + d_c^{\Omega(u)}$*, where* $d_c$ *is the constant from Theorem 2.1.*

**Proof:** The completeness follows from that of the basic test and we need to analyze the soundness. To this end let $\mathrm{Acc}(i,j,l)$ be a variable that indicates whether the test given by the triple $(i,j,l)$ accepts, taking the value $1$ if it does and $-1$ otherwise. Note that in fact

$$\mathrm{Acc}(i,j,l) = B_l(h_{ijl}) B_l(g_l) A(f_i) (A(f'_j) \wedge B_l(g'_l)).$$

Consider

$$\prod_{(i,j,l)\in Z_0} \frac{1+\mathrm{Acc}(i,j,l)}{2} = 2^{-k^2} \sum_{S \subseteq Z_0} \prod_{(i,j,l)\in S} \mathrm{Acc}(i,j,l) \qquad (5)$$

This number equals $1$ if the test accepts and is $0$ otherwise and thus its expected value is the probability that the test accepts. The term with $S = \emptyset$ is $1$ and to establish the theorem it is sufficient to establish that any other term is bounded by $d_c^{\Omega(u)}$. Let $T_S$ be the expected value of the term corresponding to $S$. We go on to establish a strategy for $P_1$ and $P_2$ which makes the verifier in 2PP($u$) accept with probability $|T_S|^{O(1)}$.

Suppose that $(i_0, j_0, l_0) \in S$ and let us fix the values of $f_i$, $i \neq i_0$, $f'_j$, $j \neq j_0$ and $(W_l, g_l, g'_l)$ for $l \neq l_0$ in such a way as not to decrease $|T_S|$. Consider any triple $(i, j, l) \in S$ other than $(i_0, j_0, l_0)$. It intersects $(i_0, j_0, l_0)$ in at most one place. If $i \neq i_0, j \neq j_0, l \neq l_0$, after the fixings, $Acc(i, j, l)$ reduces to a constant $\pm 1$. Similarly, if $i = i_0, j \neq j_0, l \neq l_0$, $Acc(i, j, l)$ reduces to $X(f_{i_0})$ where $X$ is some function of $f_{i_0}$. An important point is that $X$ depends only on the choice of $U$. If $i \neq i_0, j = j_0, l \neq l_0$, $Acc(i, j, l)$ reduces to some function $Y(f'_{j_0})$ depending only on $U$. Finally if $i \neq i_0, j \neq j_0, l = l_0$, $Acc(i, j, l)$ reduces to some function $Z(g_{l_0}, g'_{l_0})$ depending both on $W_{l_0}$ and $U$.

On the other hand, using $x \wedge y = \frac{1+x+y-xy}{2}$ one can write

$$Acc(i_0, j_0, l_0) = B_{l_0}(h_{i_0 j_0 l_0}) B_{l_0}(g_{l_0}) A(f_{i_0}) \cdot \frac{1 + A(f'_{j_0}) + B_{l_0}(g'_{l_0}) - A(f'_{j_0}) B_{l_0}(g'_{l_0})}{2}$$

Altogether we can write $T_S$ as a sum of 4 terms of the form

$$B_{l_0}(h_{i_0 j_0 l_0}) A'(f_{i_0}) A''(f'_{j_0}) C(g_{l_0}, g'_{l_0}) \qquad (6)$$

each with a coefficient $1/2$. Here $A', A'', C$ are Boolean functions that "absorb" all the functions of type $X, Y, Z$ respectively. We again stress that $A'$ and $A''$ only depend on $U$ and hence can be used to extract strategies for $P_2$. $B_{l_0}$ is the original long code on $W_{l_0}$ and hence is useful for extracting strategies for $P_1$. Finally $C$ is a Boolean function that depends on both $U$ and $W_{l_0}$ and is not useful for extracting strategies.

Let us now discard the indices for readability writing (6) as

$$B(h) A'(f) A''(f') C(g, g'),$$

We want to compute the expected value of this expression over random choices of $f$, $f'$, $g$ and $g'$. Expanding all factors except $A''(f')$ by the Fourier transform we get

$$\sum_{\alpha,\beta,\gamma,\gamma'} \hat{A}'_\alpha \hat{B}_\beta \hat{C}_{\gamma,\gamma'} E[\chi_\alpha(f) \chi_\beta(gf(f' \wedge g') \chi_\gamma(g) \chi_{\gamma'}(g') A''(f')]. \qquad (7)$$

Now taking the expected value over $f$ we see that unless $\alpha = \pi_2(\beta)$ the term is 0. Similarly we need $\beta = \gamma$, and fixing $f'$ we see that unless $\gamma' = \beta \cap \pi^{-1}(f'^{-1}(-1))$ we also get a zero expected value. This implies that the expression reduces to

$$\sum_\beta \hat{A}'_{\pi_2(\beta)} \hat{B}_\beta \hat{C}_{\beta, \beta \cap \pi^{-1}(f'^{-1}(-1))} A''(f'). \qquad (8)$$

We have

$$\begin{aligned} &| E_{f'}[\hat{C}_{\beta, \beta \cap \pi^{-1}(f'^{-1}(-1))} A''(f')] | \\ \leq \;& E_{f'}[\, |\hat{C}_{\beta, \beta \cap \pi^{-1}(f^{-1}(-1))}| \,] \\ \leq \;& 2^{-|\pi(\beta)|} \textstyle\sum_{\alpha' \subseteq \pi(\beta)} |\hat{C}_{\beta, \beta \cap \pi^{-1}(\alpha')}| \\ \leq \;& 2^{-|\pi(\beta)|/2} (\textstyle\sum_{\alpha' \subseteq \pi(\beta)} \hat{C}^2_{\beta, \beta \cap \pi^{-1}(\alpha')})^{1/2}. \end{aligned}$$

Substituting this estimate into (8) and using Cauchy-Schwartz inequality over $\beta$ we get the upper estimate

$$\left( \sum_\beta \hat{B}^2_\beta \hat{A}'^2_{\pi_2(\beta)} 2^{-|\pi(\beta)|} \right)^{1/2} \left( \sum_{\beta,\beta_1} \hat{C}^2_{\beta,\beta_1} \right)^{1/2}$$

$$\leq \left( \sum_\beta \hat{B}^2_\beta \hat{A}'^2_{\pi_2(\beta)} 2^{-|\pi(\beta)|} \right)^{1/2}$$

for $|T_S|$. The rest of the proof now follows along the same lines as for the basic test. We define the same strategies for the provers and the analysis is almost identical. We omit the details. ∎

### 3.2.2 The Bipartite Graph Test

Let

$$h_{il} = g_l f_i (g'_l \wedge f'_i)$$

and test if

$$B_l(h_{il}) = B_l(g_l) A(f_i) (B_l(g'_l) \wedge A(f'_i)) \qquad (9)$$

for all pairs $i, l$. We call this the bipartite graph test. We state the following theorem without proof.

**Theorem 3.5** *The bipartite graph test has completeness* $1$ *and soundness* $2^{-k^2} + d_c^{\Omega(u)}$.

### 3.2.3 Adaptive Tests

Guruswami et al. [8] gave an adaptive test reading three bits that has perfect completeness and soundness $\frac{1}{2} + \epsilon$ for any $\epsilon > 0$. The nonadaptive version of this test has the same parameters except that it reads 4 bits. The natural iterated test based on this test reads $2k + k^2$ bits in adaptive setting and $2k + 2k^2$ bits in the nonadaptive setting. It has perfect completeness and it turns out that soundness is essentially $2^{-k^2}$ also for this test. Thus its parameters, when adaptive, are the same as those of the test of Samorodnitsky and Trevisan while achieving perfect completeness.

We now describe this test which we call "the adaptive iterated test". The most complicated part is a probability distribution $D_\delta$ that we specify below.

1. $V$ chooses random functions $(f_i)_{i=1}^k$ on $U$ and reads the bits $A(f_i)$.

2. $V$ chooses random sets $W_j$ and a random function $g_j$ on each $W_j$, for $j = 1, 2, \dots k$.

3. For $i, j \in [k]$, $V$ picks a function $h_{ij}$ on $W$ from the distribution $D_\delta$. If $A(f_i) = 1$, $V$ checks that $B_j(g_j(f_i \wedge h_{ij})) = B_j(g_j)$ and otherwise $V$ checks that $B_j(g_j(-f_i \wedge h_{ij})) = B_j(g_j)$.

The distribution $D_\delta$ is the same distribution as used in [8] and [9]. It is defined as follows.

1. Set $t = \lceil \delta^{-1} \rceil$, $\epsilon_1 = \delta$ and $\epsilon_i = (\delta^{1+2/c} 2^{-1/c})^{i-1} \delta$, for $i = 2, 3, \dots t$, where $c$ is the constant from Lemma 3.3.

2. Pick a random $i$, $1 \le i \le t$ with uniform probability and let $h$ be a function that takes value 1 with probability $\epsilon_i$ and value $-1$ with probability $1 - \epsilon_i$, independently for each input.

Completeness is straightforward.

**Lemma 3.6** *The adaptive iterated test has perfect completeness.*

**Proof:** Fix an $i$ and $j$. Suppose that we have a correct proof of a correct theorem based on the global assignment $z$. If $A(f_i) = 1$ then $f_i(\pi^U(z)) = 1$ and we have

$$B_j(g_j(f_i \wedge h_{ij})) = g_j(\pi^W(z))(f_i(\pi^U(z)) \wedge h_{ij}(\pi^W(z)))$$
$$= g_j(\pi^W(z)) = B_j(g_j).$$

The case $A(f_i) = -1$ is similar. ∎

We next turn to soundness.

**Lemma 3.7** *Given a proof that makes the verifier in the adaptive iterated test with parameter $\delta$ accept with probability $2^{-k^2} + 2\delta$, we can find strategies for $P_1$ and $P_2$ in 2PP$(u)$ that makes the verifier of that protocol accept with probability $2^{-O(\delta^{-1})}$.*

**Proof:** The proof follows along the same lines as the result for the protocol with $k = 1$ given in [8]. That proof in its turn is very much based on the proof that 3-Sat is inapproximable for satisfiable instances in [9] and we use the notation of that proof. Let

$$\mathrm{Acc}(i,j) = \frac{1 + A(f_i)}{2} B_j(g_j) B_j(g_j(f_i \wedge h_{ij})) +$$
$$\frac{1 - A(f_i)}{2} B_j(g_j) B_j(g_j(-f_i \wedge h_{ij}))$$

which is 1 if the test given by $(i, j)$ accepts and $-1$ otherwise. We have an expansion like (5) and we assume we have a term corresponding to a nonempty $S$ that is at least $2\delta$.

We may assume that $(1,1) \in S$. Fixing the values of $g_j$, and $f_i$ for $i, j \ge 2$ and for $h_{ij}$ for $(i,j) \ne (1,1)$, we conclude that there is a term of the form

$$A'(f_1) B(g_1(f_1 \wedge h_{11})) C(g_1) \tag{10}$$

or

$$A'(f_1) B(g_1(-f_1 \wedge h_{11})) C(g_1) \tag{11}$$

that is at least $\delta$ in absolute value. Here $A'$, $B$ and $C$ are Boolean functions where $B$ is the original $B_1$ and $A'$ is a function only depending on $U$. Since we are not putting any conditions on $A'$, $f_1$ is equivalent to $-f_1$ and hence we might as well study (10).

Replacing each function by its Fourier expansion, we, arrive at the condition

$$|E_{U,W,i} \sum_{\beta, \alpha \subseteq \pi(\beta)} \hat{A}'_\alpha \hat{B}_\beta \hat{C}_\beta p_i(\alpha, \beta)| \ge \delta,$$

where $i \in [t]$ as defined in the definition of $D_\delta$ and

$$p_i(\alpha, \beta) = \prod_{x \in \alpha \cap \pi(\beta)} (\frac{1}{2}(1 - (2\epsilon_i - 1)^{s_x}))$$
$$\prod_{x \in \pi(\beta) \setminus \alpha} (\frac{1}{2}(1 + (2\epsilon_i - 1)^{s_x})),$$

where $s_x$ is the cardinality of $\pi^{-1}(x) \cap \beta$. The functions $p_i$ are, upto sign, equal to the function $p$ used in the corresponding proof of [9] when the value of $\epsilon$ is equal to $\epsilon_i$. The difference in sign is of no importance since all estimates involve $p^2$.

Thus we have essentially derived the condition (42) on page 46 of [9]. We have some minor differences.

1. We have replaced $A$ by $A'$ and the latter cannot be assumed to be folded.

2. We have replaced $\hat{B}_\beta^2$ by $\hat{B}_\beta \hat{C}_\beta$.

The key property of the numbers $\hat{B}_\beta^2$ is that they sum to one and this remains true in the form that

$$\sum_\beta |\hat{B}_\beta \hat{C}_\beta| \le \left( \sum_\beta \hat{B}_\beta^2 \right)^{1/2} \left( \sum_\beta \hat{C}_\beta^2 \right)^{1/2} \le 1. \tag{12}$$

Note also that $B$ is still folded over true and hence any $\beta$ appearing in the above sum is of odd size.

The differences are small however and can be handled without problems. Let us give a sketch.

To handle the first difference one simply observes that terms corresponding to $\hat{A}'_\emptyset$ are similar to the terms without the factor $A$ in [9] which are covered by Lemma 6.7 of [9].

The fact (12) makes the second difference inessential except in the chain of inequalities on the bottom of page 47 of [9]. There instead of getting the factor

$$\sum_{\beta,\alpha\subseteq\pi(\beta)} \hat{B}_\beta^2 p^2(\alpha,\beta)$$

we get the factor

$$\sum_{\beta,\alpha\subseteq\pi(\beta)} \hat{C}_\beta^2 p^2(\alpha,\beta)$$

which is also bounded by 1 and hence we get the same bound. The rest of the proof is unaffected. ∎

## 4. Conclusions

We have established that the query efficient test of Samorodnitsky and Trevisan can be extended to include perfect completeness in several different ways. The tests are simple and the analysis only moderately complicated, in particular the proofs using the approach of [11] are fairly straightforward.

All this taken together gives us good hope that we, in the not too distant future will see more powerful PCPs with even more applications to in-approximability of NP-hard optimization problems. In particular the fact that we can include perfect completeness gives hope that stronger lower bounds for coloring of graphs of small chromatic index could be possible. Clearly to obtain such results obstacles of other nature need also be overcome. We note that some progress for constant colorable graphs has already occurred [12], but getting strong results for 3-colorable graphs seems to require new ideas.

## 5. Acknowledgements

We would like to thank Sanjeev Arora for several helpful discussions.

## References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[2] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[3] M. Bellare, O. Goldreich and M. Sudan. Free bits, PCP's and non-approximability – towards tight results. *SIAM Journal on Computing*, 27(3):804-915, 1998.

[4] L. Engebretsen. Lower Bounds for non-Boolean Constraint Satisfaction, *ECCC report TR00-042*.

[5] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, vol 45, 1998, pp 634–652.

[6] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 1996, Vol 43:2, pp 268–292.

[7] V. Guruswami, J. Håstad, M. Sudan. Hardness of approximate hypergraph coloring. *In Proceedings of the 41st IEEE Symposium of Foundations of Computer Science*, 2000, pp 149-158.

[8] V. Guruswami, D. Lewin, M. Sudan, L. Trevisan. A new characterization of NP with 3 query PCPs. *In Proceedings of 39th Annual IEEE Symposium of Foundations of Computer Science*, 1998.

[9] J. Håstad. Some optimal inapproximability results. *Technical Report TR97-37, Electronic Colloquium on Computational Complexity*, 1997. Preliminary version in *Proceedings 29th Annual ACM Symposium on Theory of Computation*, 1997, pp 1–10. Final version accepted for publication in Journal of the ACM. The exact references refers to the version accepted to Journal of the ACM available from the authors homepage.

[10] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, Vol. 182, 1999, pp 105–142.

[11] J. Håstad and A. Wigderson. Simple analysis of graph tests for linearity and PCP, accepted for presentation at 2001 Computational Complexity conference.

[12] S. Khot. Improved inapproximability results for maxclique, chromatic number and approximate graph coloring. Accepted for presentation at the 42nd Annual Symposium on Foundations of Computer Science, 2001.

[13] M. Kiwi. Probabilistically Checkable Proofs and the Testing of Hadamard-like codes, Ph.D-thesis, MIT, 1996.

[14] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[15] A. Samorodnitsky and Luca Trevisan. A PCP characterization of NP with optimal amortized query complexity. *In proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pp 191-199, 2000.

## A. The Case of Larger Domains

In many cases, the results on PCPs established for the binary case readily extend to the case of larger domains. This is true for the linearity test ([13], see also [11]) and for the test of Samorodnitsky and Trevisan [4] getting close to optimal soundness as a function of the number of symbols read. This is also true in our current situation.

We write $Z_p$ as the multiplicative group of the $p^{th}$ roots of unity. Let $\zeta = e^{2\pi i/p}$ be the basic $p^{th}$ root of unity. We define an operation $mult( \, , \, )$ as :

$$mult(\zeta^i, \zeta^j) = \zeta^{ij}.$$

We have the following useful lemma

**Lemma A.1** *For $x$ and $y$ being $p^{th}$ roots of unity we have*

$$mult(x, y) = \frac{1}{p} \sum_{i=0}^{p-1} \sum_{j=0}^{p-1} x^i y^j \zeta^{-ij}.$$

**Proof:** Suppose $y = \zeta^{i_0}$. Fix $i$ and consider the inner sum. For $i \neq i_0$ the value is 0 while for $i = i_0$ it is $p$. This implies that the total sum equals $x^{i_0}$ which is in fact $mult(x, y)$. ∎

The written PCP is similar to the binary case except in that instead of the long codes we have long $p$ codes where the positions are indexed by functions $f : \{-1, 1\}^t \mapsto Z_p$. The value in such a position in a correct proof is given by $f(x)$.

We again let $A$ and $B$ be supposed long $p$ codes on $U$ and $W$ respectively. We need to define rules for accessing these and then the Fourier transform.

**Definition A.2** *A supposed long $p$-code $A$ is folded over true if $A(\zeta^a f) = \zeta^a A(f)$, for $0 \le a \le p-1$ and all $f$.*

**Definition A.3** *A supposed long $p$-code $A$ respects exponentiation if $A(f^a) = A(f)^a$ for $0 \le a \le p-1$ and all $f$.*

**Definition A.4** *A supposed long $p$-code $A$ is conditioned upon a function $h : \{-1, 1\}^t \mapsto \{1, \zeta\}$ (1 represents false and $\zeta$ represents true), if $A(f) = A(mult(f, h))$ for all $f$.*

Now we briefly explain Fourier analysis of long $p$-codes. For every function $\alpha : \{-1, 1\}^t \mapsto GF(p)$, there is a character $\chi_\alpha$ defined as

$$\chi_\alpha(f) = \prod_{x \in \{-1,1\}^t} f(x)^{\alpha(x)}$$

Note that $\alpha$ is a "function" rather than a "set" as in binary case and that the transform might take complex values. We denote by $N(\alpha)$ the set on which $\alpha$ takes nonzero values i.e. $N(\alpha) = \{x | \alpha(x) \neq 0\}$. Every table $A$ can be written as $A(f) = \sum_\alpha \hat{A}_\alpha \chi_\alpha(f)$ with $\sum_\alpha |\hat{A}_\alpha|^2 = 1$. We can assume that tables are folded or conditioned upon a given function by using appropriate access mechanisms. Following are easy consequences of folding and conditioning.

**Lemma A.5** *If $A$ is folded over true and $\hat{A}_\alpha \neq 0$, then $\sum_{x \in \{-1,1\}^t} \alpha(x) = 1$ (in GF(p)). In particular $N(\alpha)$ is a nonempty set.*

**Lemma A.6** *If $A$ is conditioned upon a function $h : \{-1, 1\}^t \mapsto \{1, \zeta\}$ and $\hat{A}_\alpha \neq 0$, then for every $x \in N(\alpha)$, $h(x)$ is true, i.e. $h(x) = \zeta$.*

## A.1. Iterated Tests

The tests for long $p$-codes are very similar to the tests for boolean case in Section 3. Hence we directly present the iterated test without first going through the basic test. We have only attempted the simpler analysis of almost disjoint sets test and this is what we present here.

### A.1.1 The Almost Disjoint Sets Test

We assume that tables $A, B$ are folded over true and respect exponentiation. The $B$ tables (supposed long $p$-codes on $W$s) are conditioned upon the CNF formula $\varphi_W$.

The verifier picks one set $U$, and on it picks $k$ functions $(f_i)_{i=1}^k$ and $k$ functions $(f'_j)_{j=1}^k$. It picks $k$ sets $W_1, \dots, W_k$ and on set $W_l$, picks a pair of functions $(g_l, g'_l)$. Define

$$h_{ijl} = g_l \; f_i \; mult(f'_j, g'_l)$$

and check that

$$B_l(h_{ijl}) = B_l(g_l) \; A(f_i) \; mult(A(f'_j), B_l(g'_l))$$

again for all $i, j, l \in Z_0$, i.e. for $i + j + l \equiv 0 \bmod k$. Let us call this the "Almost disjoint sets test in $Z_p$". We have the following theorem.

**Theorem A.7** *The almost disjoint sets test in $Z_p$ has completeness 1 and soundness $p^{-k^2} + d_c^{\Omega(u)}$, where $d_c$ is the constant from Theorem 2.1.*

**Proof:** The completeness is obvious and we need to analyze the soundness. To this end let

$$\mathrm{Acc}(i, j, l) = B_l(h_{ijl})^{-1} B_l(g_l) A(f_i) \; mult(A(f'_j), B_l(g'_l)).$$

Let $S \in GF(p)^{k^2}$ be a vector whose coordinates are indexed by the triples in $Z_0$. So $S(i, j, l)$ is the coordinate corresponding to the triple $(i, j, l) \in Z_0$. Consider the expression :

$$\prod_{(i,j,l) \in Z_0} \frac{\sum_{a=0}^{p-1} (\mathrm{Acc}(i, j, l))^a}{p}$$

$$= p^{-k^2} \sum_{S \in GF(p)^{k^2}} \prod_{(i,j,l) \in Z_0} (\mathrm{Acc}(i, j, l))^{S(i,j,l)}$$

This expression equals 1 if the test accepts and is 0 otherwise and thus its expected value is the probability that the test accepts. The term with $S \equiv 0$ is 1 and to establish the theorem it is sufficient to establish that any other term is bounded by $d_c^{\Omega(u)}$. Let $T_S$ be the expected value of the term corresponding to $S$. We go on to establish a strategy for $P_1$ and $P_2$ which makes the verifier in 2PP($u$) accept with probability $|T_S|^{O(1)}$.

Suppose that $r = S(i_0, j_0, l_0) \neq 0$ and let us fix the values of $f_i, i \neq i_0$, $f_j, j \neq j_0$ and $(W_l, g_l, g'_l)$ for $l \neq l_0$ in such a way as not to decrease $|T_S|$. From the fact that any other triple $(i, j, l)$ intersects the triple $(i_0, j_0, l_0)$ in at most one place, we can "reduce" $Acc(i, j, l)$ to a function of type $X(f_{i_0}), Y(f'_{j_0})$ or $Z(g_{l_0}, g'_{l_0})$ in a manner similar to Section 3.2.1. By Lemma A.1, we can expand $Acc(i_0, j_0, l_0)$ into a sum of $p^2$ terms. Thus after the above fixings, $T_S$ can be written as the sum of $p^2$ terms of the form

$$B(h)^{-r} A'(f) A''(f') C(g, g'),$$

each with a coefficient $1/p$, where we discard the indices $i_0, j_0, l_0$ for readability. Here $A'$ and $A''$ are functions that only depend on $U$ and hence might be used to extract strategy for $P_2$. $B$ is the original long $p$-code on $W = W_{l_0}$ and hence is useful for extracting strategy for $P_1$. Finally $C$ is a function that depends on both $U$ and $W_{l_0}$ and is not useful for extracting strategies.

We now want to compute the expected value of this expression over random choices of $f$, $f'$, $g$ and $g'$. Expanding all factors except $A''(f')$ by the Fourier transform we get

$$\sum_{\alpha,\beta,\gamma,\gamma'} \hat{A}'_\alpha \hat{B}_\beta \hat{C}_{\gamma,\gamma'} E[\chi_\alpha(f) \chi_{-r\beta}(gf\, mult(f', g')) \chi_\gamma(g) \chi_{\gamma'}(g') A''(f')]. \quad (13)$$

Now taking the expected value over $f$ we see that unless $\alpha = r\pi_p(\beta)$ the term is 0 where $\pi_p$ is a generalization of $\pi_2$ defined by

$$\pi_p(\beta)(x) = \sum_{y \in \pi^{-1}(x)} \beta(y) \ \ (\text{in } GF(p))$$

Similarly we need $\gamma = r\beta$. Fix $f'$ and define $\beta'$ as follows : for every $y$, $\beta'(y) = re(y)\beta(y)$ where $f'(\pi(y)) = \zeta^{e(y)}$. With this definition, we have

$$\chi_{-r\beta}(mult(f', g')) = \chi_{-\beta'}(g')$$

Thus unless $\gamma' = \beta'$, the expectation is 0. The expression reduces to

$$\sum_\beta \hat{A}'_{r\pi_p(\beta)} \hat{B}_\beta \hat{C}_{\beta,\beta'} A''(f') \quad (14)$$

Note that $\beta'$ depends on $\beta$ and $f'$, and $f's$ which are different on $\pi(N(\beta))$ give different $\beta's$. Let $\Delta_\beta$ be the set of all possible $\beta'$s. We have $|\Delta_\beta| = p^{|\pi(N(\beta))|}$ and over all the choices of $f'$, every $\beta' \in \Delta_\beta$ occurs equally often. This implies that

$$\begin{aligned} & |\, E_{f'}[\hat{C}_{\beta,\beta'} A''(f')] \,| \ \leq E_{f'}[|\hat{C}_{\beta,\beta'}|] \\ = \ & p^{-|\pi(N(\beta))|} \textstyle\sum_{\beta' \in \Delta_\beta} |\hat{C}_{\beta,\beta'}| \\ \leq \ & p^{-|\pi(N(\beta))|/2} (\textstyle\sum_{\beta' \in \Delta_\beta} |\hat{C}_{\beta,\beta'}|^2)^{1/2}. \end{aligned} \quad (15)$$

Substituting this estimate into (14) and using Cauchy-Schwartz inequality over $\beta$ we get the upper estimate

$$\left( \sum_\beta |\hat{B}_\beta|^2 |\hat{A}'_{r\pi_p(\beta)}|^2 p^{-|\pi(N(\beta))|} \right)^{1/2} \left( \sum_{\beta, \beta' \in \Delta_\beta} |\hat{C}_{\beta,\beta'}|^2 \right)^{1/2} \leq \left( \sum_\beta |\hat{B}_\beta|^2 |\hat{A}'_{r\pi_p(\beta)}|^2 p^{-|\pi(N(\beta))|} \right)^{1/2}$$

for $|T_S|$. Now we can define strategies for provers in usual manner which make the verifier accept in 2PP($u$) with probability $|T_S|^{O(1)}$. A minor difference is that now $\alpha$ ( $\beta$ ) are functions and not sets. The provers pick $\alpha$ ( $\beta$ ) with probability $\hat{A}_\alpha^2$ ( $\hat{B}_\beta^2$ ) and pick a random $x \in N(\alpha)$ (a random $y \in N(\beta)$ ).

■