# The PCP Theorem by Gap Amplification

Irit Dinur[*]

April 22, 2005

**Abstract**

Let $\mathcal{C} = \{c_1, \ldots, c_n\}$ be a set of constraints over a set of variables. The *satisfiability-gap* of $\mathcal{C}$ is the smallest fraction of unsatisfied constraints, ranging over all possible assignments for the variables.

We prove a new combinatorial amplification lemma that doubles the satisfiability-gap of a constraint-system, with only a linear blowup in the size of the system. Iterative application of this lemma yields a self-contained (combinatorial) proof for the PCP theorem.

The amplification lemma relies on a new notion of "graph powering" that can be applied to systems of constraints. This powering amplifies the satisfiability-gap of a constraint system provided that the underlying graph structure is an expander.

We also apply the amplification lemma to construct PCPs and locally-testable codes whose length is *linear up to a polylog* factor, and whose correctness can be probabilistically verified by making a *constant* number of queries. Namely, we prove $SAT \in PCP_{\frac{1}{2},1}[\log_2(n \cdot \text{poly} \log n), O(1)]$. This answers an open question of Ben-Sasson et al. (STOC '04).

## 1 Introduction

Let $\mathcal{C} = \{c_1, \ldots, c_n\}$ be a set of constraints over a set of variables $V$. The *satisfiability-gap* of $\mathcal{C}$, denoted $\overline{\text{SAT}}(\mathcal{C})$, is the smallest fraction of unsatisfied constraints, over all possible assignments for $V$. Clearly $\mathcal{C}$ is satisfiable if and only if $\overline{\text{SAT}}(\mathcal{C}) = 0$. Also, if $\mathcal{C}$ is not satisfiable then $\overline{\text{SAT}}(\mathcal{C}) \geq 1/n$.

**Background** The PCP Theorem is equivalent to stating that gap-3SAT is NP-hard, in the following sense: for some $\alpha > 0$, given a set $\mathcal{C}$ of constraints such that each is an conjunction of three literals, it is NP-hard to distinguish between $\overline{\text{SAT}}(\mathcal{C}) = 0$ and $\overline{\text{SAT}}(\mathcal{C}) > \alpha$. Historically, the PCP Theorem has been formulated through interactive proofs and the concept of a probabilistic verifier that can check an NP witness by randomly probing it at only $O(1)$ bit locations. The [FGL+96, ALM+98] connection between this formulation and inapproximability, as in the gap-3SAT formulation stated above, came as a big surprise. Together with the proof of the PCP Theorem by [AS98, ALM+98], it brought about a revolution of the field of inapproximability. The proof of the theorem followed an exciting sequence of developments in interactive proofs. The proof techniques were mainly algebraic including low-degree extension, low-degree test, parallelization through curves, a sum-check protocol, and the Hadamard and quadratic functions encodings.

[*] Hebrew University. Email: dinuri@cs.huji.ac.il. Supported by the Israel Science Foundation.

**Gap Amplification** In this paper we take a different approach for proving the PCP Theorem. Our approach is quite natural in the context of inapproximability. We start with a 3SAT system $\mathcal{C}$, so it is NP-hard to decide if $\mathcal{C}$ is satisfiable or not. Namely, it is NP-hard to distinguish between the cases (i) $\overline{\mathrm{SAT}}(\mathcal{C}) = 0$ and (ii) $\overline{\mathrm{SAT}}(\mathcal{C}) \geq 1/n$. Now repeatedly apply the amplification lemma to $\mathcal{C}$, doubling the satisfiability gap at each iteration. The outcome $\mathcal{C}'$ is a constraint system for which in the first case still $\overline{\mathrm{SAT}}(\mathcal{C}') = 0$, and in the second case $\overline{\mathrm{SAT}}(\mathcal{C}') \geq \alpha$ for some $\alpha > 0$. This gives a reduction from 3SAT to gap-3SAT, thus proving the PCP Theorem.

What makes the gap double? Let us restrict ourselves to systems of constraints over two variables. Satisfiability is still NP-complete for such systems, for some constant-size (non-Boolean) alphabet. A two-variable constraint system naturally defines an underlying graph, in which the variables are vertices, and two variables are adjacent iff there is a constraint over them. We call this a *constraint graph*. In order to amplify the gap of a constraint graph we simply raise it to the power $t$, for some $t = O(1)$. The *graph powering* operation is defined as follows: The new underlying graph is the $t$-th power of the original graph (with the same vertex-set, and an edge for each length-$t$ path). Each vertex will hold a value over a larger alphabet, that describes its own value plus it's "opinion" about the values of all of its neighbors at distance $\leq t/2$. The constraint over two adjacent vertices $u, v$ in the new graph will be satisfied iff the values and opinions of $u$ and $v$ are consistent with an assignment that satisfies all of the constraints induced by $u, v$ and their neighborhoods.

Our main lemma asserts that the gap is multiplied by a factor of roughly $\sqrt{t}$, as long as the initial underlying graph is sufficiently "well-structured".

The main advantage of this operation is that it *does not increase* the number of variables in each constraint (which stays 2 throughout). Moreover, when applied to $d$-regular graphs for $d = O(1)$, it only incurs a *linear* blowup in the size (the number of edges is multiplied by $d^{t-1}$), and an affordable increase in the alphabet size (which goes from $\Sigma$ to $\Sigma^{d^{t/2}}$). Combined with an operation that reduces the alphabet back to $\Sigma$, we get an inductive step that can be repeated $\log n$ times until a constant gap is attained.

**Composition** Reducing the alphabet size is an easy task assuming we have at our disposal a PCP reduction $\mathcal{P}$. A PCP reduction is an algorithm that takes as input a single large-alphabet constraint, and outputs a system of (perhaps many) constraints over a smaller alphabet. Indeed, all we need to do is to run $\mathcal{P}$ on each of the constraints in our system[1]. This results in a new constraint system with a similar gap, and over a smaller alphabet. At first read, this argument may appear to be circular, as the reduction $\mathcal{P}$ sounds very much like our end-goal. The point is that since in our setting the input to $\mathcal{P}$ always has *constant size*, $\mathcal{P}$ is allowed to be extremely inefficient. This relaxation makes $\mathcal{P}$ significantly easier to construct, and one can choose their favorite implementation, be it Long-code based or Hadamard-code based. In fact, $\mathcal{P}$ can be found by exhaustive search, provided we have proven its existence in an independent fashion. Composition with $\mathcal{P}$ is direct and simple, relying on the relatively recent 'modularized' notion of composition using "assignment-testers" [DR04] or "PCPs of proximity" [BGH+04].

Thus, our proof of the PCP Theorem roughly takes the following form: Let $G$ encode a SAT instance. Fix $t = O(1)$, set $G_0 = G$, and repeat the following step $\log n$ times:

$$G_{i+1} = (G_i)^t \circ \mathcal{P}$$

**Related Work** This construction is inspired by the zig-zag construction of expander graphs due to [RVW] and by Reingold's proof for $SL = L$ [Rei05]. Reingold shows how one iteration of powering /

[1] While ensuring consistency between the many invocations of $\mathcal{P}$.

zigzagging, increases the spectral gap of any graph; so after $\log n$ iterations the initial graph becomes an expander.

Our proof has the same overall structure, where each iteration consists of powering and composition. In this proof it is the satisfiability gap, rather than the spectral gap, that is increased steadily in each step.

The steady increase of the satisfiability gap is inherently different from the original proof of the PCP Theorem. There, a constant satisfiability gap (using our terminology) is generated by one powerful transformation, and then a host of additional transformations are incorporated into the final result to take care of other parameters.

This work follows [GS97, DR04] in the attempt to find an alternative proof for the PCP Theorem that is combinatorial and/or simpler. In [DR04], a quasi-polynomial PCP Theorem was proven combinatorially. While our proof is different, we do rely on the modular notion of composition due to [BGH$^+$04, DR04], and in particular on composition with a bounded-input assignment-tester, which has already served as an ingredient in the constructions of [DR04].

**Short PCPs and Locally Testable Codes** The goal of achieving extremely-short Probabilistically Checkable Proofs and Locally-Testable Codes (LTCs) has been the focus of several works [PS94, HS01, GS02, BSVW03, BGH$^+$04, BS05]. The shortest PCPs/LTCs are due to [BGH$^+$04] and [BS05], each best in a different parameter setting. For the case where the number of queries is constant, the shortest construction is due to [BGH$^+$04], and the proof-length is $n \cdot 2^{(\log n)^\varepsilon}$. The construction of [BS05] has shorter proof-length, $n \cdot \text{poly}\log n$, but the number of queries it requires is $\text{poly}\log n$. Our result combines the best parameters from both of these works. Our starting point is the construction [BS05]. We first transform this construction into a two-query constraint system $\mathcal{C}$ whose size is $n \cdot \text{poly}\log n$, such that if the input was a 'no' instance, then $\overline{\text{SAT}}(\mathcal{C}) \geq \frac{1}{\text{poly}\log n}$. Then, by applying our amplification lemma $O(\log\log n)$ times, we raise the gap to a constant, while increasing the size of the system by only another polylogarithmic factor. Namely, we show that $SAT \in PCP_{\frac{1}{2},1}[\log_2(n \cdot \text{poly}\log n),\ O(1)]$.

**Organization** Section 2 contains some preliminaries, including a formal definition of constraint graphs, and some basic facts about expander graphs. In Section 3 we describe the operations on constraint graphs on which we base our construction. In Section 4 we prove the PCP Theorem. The proof of the gap amplification lemma is given in Section 5. In Section 6 we describe a concrete (and inefficient) construction of an assignment tester $\mathcal{P}$ based on the Long-Code, so as to make our result self-contained. In Section 7 we prove $SAT \in PCP_{\frac{1}{2},1}[\log_2(n \cdot \text{poly}\log n), O(1)]$.

## 2 Preliminaries

### 2.1 Constraint Graphs

In this paper we are interested in systems of constraints, as well as in the graph structure underlying them. We restrict our attention to systems of two-variable constraints, whose structure is captured by 'constraint graphs', defined as follows:

**Definition 2.1 (Constraint Graph)** *$G = \langle (V,E), \Sigma, \mathcal{C}\rangle$ is called a constraint graph, if*

1. *$(V,E)$ is an undirected graph, called the underlying graph of $G$.*

2. *The set $V$ is also viewed as a set of variables assuming values over alphabet $\Sigma$*

3. *Each edge $e \in E$, carries a constraint $c^e : \Sigma^2 \to \{\mathsf{T}, \mathsf{F}\}$, and $\mathcal{C} = \{c^e\}_{e\in E}$.*

An assignment is a mapping $\sigma : V \to \Sigma$ that gives each vertex in $V$ a value from $\Sigma$. For any assignment $\sigma$, define

$$\textsc{sat}_\sigma(G) = \Pr_{(u,v)\in E}[c^e(\sigma(u),\sigma(v)) = \mathsf{T}] \quad \text{and} \quad \textsc{sat}(G) = \max_\sigma \textsc{sat}_\sigma(G)\,.$$

Also define $\overline{\textsc{sat}}_\sigma(G) = 1 - \textsc{sat}_\sigma(G)$ and $\overline{\textsc{sat}}(G) = 1 - \textsc{sat}(G)$. We call $\overline{\textsc{sat}}(G)$ the **satisfiability-gap** of $G$, or just the gap of $G$ for short. We denote by $size(G)$ the size of the description of $G$, so $size(G) = \theta(|V| + |E| \cdot |\Sigma|^2)$.

**Proposition 2.1 (Constraint-Graph Satisfiability)** *Given a constraint graph $G = \langle (V,E),\Sigma,\mathcal{C}\rangle$ with $|\Sigma| \le 7$, it is NP-hard to decide if* $\textsc{sat}(G) = 1$.

**Proof:** Reduce from 3SAT. Put a vertex for each clause and let the alphabet be $\{1,\ldots,7\}$, standing for all possible assignments that satisfy that clause. Put a consistency constraint for every pair of clauses that share a variable. ■

We sometimes use the same letter $G$ to denote the constraint graph and the underlying graph. In particular, we refer to the degree of $G$ and we write $\lambda(G)$ to signify the second largest eigenvalue value of the adjacency matrix of the graph underlying $G$.

## 2.2 Expander Graphs

**Definition 2.2** *Let $G = (V,E)$ be a $d$-regular graph. Let $E(S,\bar{S}) = \left|(S\times\bar{S})\cap E\right|$ equal the number of edges from a subset $S \subseteq V$ to its complement. The edge expansion is defined as*

$$h(G) = \min_{S,|S|<|V|/2} \frac{E(S,\bar{S})}{|S|}\,.$$

**Lemma 2.2 (Expanders)** *There exist $d_0 \in \mathbb{N}$ and $h_0 > 0$, such that there is a polynomial-time constructible family $\{X_n\}_{n\in\mathbb{N}}$ of $d_0$-regular graphs $X_n$ on $n$ vertices with $h(X_n) \ge h_0$.* ■

**Proof:** It is well-known that a random constant-degree graph on $n$-vertices is an expander. For a deterministic construction, one can get expanders on $2^k$ vertices for any $k$ from the construction of [RVW]. For $n = 2^k - n'$ ($n' < 2^{k-1}$), one can, for example, merge $n'$ pairs of vertices. To make this graph regular one can add arbitrary edges to the non-merged vertices. Clearly, the edge expansion is maintained up to a constant factor. ■

The following relation between the edge expansion and the second eigenvalue is known, see, e.g., [LW03],

**Lemma 2.3** *Let $G$ be a $d$-regular graph, and let $h(G)$ denote the edge expansion of $G$. Then*

$$\lambda(G) \le d - \frac{h(G)^2}{d}\,.$$

■

Finally, we prove the following (standard) estimate on the random-like behavior of a random-walk on an expander.

**Proposition 2.4** *Let $G = (V,E)$ be a $d$-regular graph with second largest eigenvalue $\lambda$. Let $F \subseteq E$ be a set of edges. The probability $p$ that a random walk that starts at a random edge in $F$ takes the $i+1$st step in $F$ as well, is bounded by $\frac{|F|}{|E|} + \left(\frac{|\lambda|}{d}\right)^i$.*

**Proof:** Let $K$ be the distribution on vertices induced by selecting a random edge in $F$, and then a random vertex in it[2]. Let $B$ be the support of $K$. Let $A$ be the normalized $n \times n$ adjacency matrix of $G$ so $A_{ij}$ equals $q/d$ where $q$ is the number of edges between vertices $i$ and $j$. The first and second eigenvalues of $A$ are $1$ and $\tilde{\lambda} = \lambda/d$ respectively.

Let $x$ be the vector corresponding to the distribution $K$, i.e. $x_v = \Pr_K[v]$ equals the fraction of edges touching $v$ that are in $F$, divided by 2. Since the graph is $d$-regular, $\Pr_K[v] \le \frac{d}{2|F|}$. Let $y_v$ be the probability that a random step from $v$ is in $F$, so $y = \frac{2|F|}{d}x$. The probability $p$ equals the probability of landing in $B$ after $i$ steps, and then taking a step in $F$,

$$p = \sum_{v \in B} y_v (A^i x)_v = \sum_{v \in V} y_v (A^i x)_v = \langle y, A^i x \rangle .$$

Let $\mathbf{1}$ be the all $1$ vector. Write $x = x^\perp + x^{||}$ where $x^{||} = \frac{1}{n}\mathbf{1}$, is an eigenvector of $A$ with eigenvalue $1$, and $x^\perp = x - x^{||}$. The vector $x^\perp$ is orthogonal to $x^{||}$ since $\mathbf{1} \cdot x^\perp = \sum_v \Pr_K[v] - \sum_v \frac{1}{n} = 1 - 1 = 0$. Denote $\|x\| = \sqrt{\sum_v x_v^2}$. Observe that $\|x\|^2 \le (\sum_v |x_v|) \cdot (\max_v |x_v|) \le 1 \cdot (\max_v |x_v|) \le \frac{d}{2|F|}$. Clearly,

$$\|A^i x^\perp\|_2 \le |\tilde{\lambda}|^i \|x^\perp\|_2 \le |\tilde{\lambda}|^i \|x\|_2 \le |\tilde{\lambda}|^i (\max_v x_v)^{1/2} \le |\tilde{\lambda}|^i \sqrt{\frac{d}{2\,|F|}}$$

By Cauchy-Schwartz

$$\langle y, A^i x^\perp \rangle \le \|y\| \cdot \|A^i x^\perp\| \le \frac{2\,|F|}{d} |\tilde{\lambda}|^i \|x\|^2 \le |\tilde{\lambda}|^i$$

Combining the above we get the claim,

$$\langle y, A^i x \rangle = \langle y, A^i x^{||} \rangle + \langle y, A^i x^\perp \rangle \le \frac{2\,|F|}{dn} + |\tilde{\lambda}|^i = \frac{|F|}{|E|} + \left(\frac{|\lambda|}{d}\right)^i$$

■

## 3 Operations on Constraint Graphs

Our main theorem is proven by performing three operations on constraint graphs:

- Preprocessing: This simple operation preserves both the gap (roughly) and the alphabet size, but makes the constraint graph more nicely structured.
- Powering: The operation which amplifies the gap, at the expense of increasing the alphabet size.
- Composition: The operation which reduces the alphabet size, while maintaining the gap (roughly).

These operations are described in Sections 3.1, 3.2 and 3.3 respectively.

[2] Let us adopt the convention that a self-loop is "half" an edge, and its probability is of being selected is defined accordingly. In the application $F$ will contain no self-loops so this whole issue can be safely ignored.

### 3.1 Preprocessing

We describe how to (easily) turn any constraint graph into a 'nicely-structured' one. By 'nicely-structured' we mean regular, constant-degree, and expanding.

**Lemma 3.1 (Preprocessing)** *There exist constants $0 < \lambda < d$ and $\beta_1 > 0$ such that any constraint graph $G$ can be transformed into a constraint graph $G'$, denoted $G' = prep(G)$, such that*

- *$G'$ is $d$-regular with self-loops, and $\lambda(G') \leq \lambda < d$.*
- *$G'$ has the same alphabet as $G$, and $size(G') = O(size(G))$.*
- *$\beta_1 \cdot \overline{\text{SAT}}(G) \leq \overline{\text{SAT}}(G') \leq \overline{\text{SAT}}(G)$.*

Note that the third item implies that completeness is maintained, i.e., if $\text{SAT}(G) = 1$ then $\text{SAT}(G') = 1$. We prove this lemma in two steps, summarized in the next two lemmas.

**Lemma 3.2 (Constant degree)** *Any constraint graph $G = \langle (V,E), \Sigma, \mathcal{C} \rangle$ can be transformed into a $(d_0+1)$-regular constraint graph $G' = \langle (V',E'), \Sigma, \mathcal{C}' \rangle$ such that $|V'| = 2|E|$ and*

$$\overline{\text{SAT}}(G)/c \leq \overline{\text{SAT}}(G') \leq \overline{\text{SAT}}(G)$$

*for some global constants $d_0, c > 0$.*

This lemma is a well-known 'expander-replacement' transformation, due to [PY91]. We include a proof for the sake of completeness. The idea is to split each vertex $v$ into $deg(v)$ new vertices that are interconnected via a constant-degree expander, placing equality constraints on the new edges. Intuitively, this maintains $\overline{\text{SAT}}(G)$ because the expander edges "penalize" assignments for the new graph that do not assign the same value to all copies of $v$; hence assignments for the new graph behave just like assignments for $G$.

**Proof:** For each $n$, let $X_n$ be a $d_0$-regular expander on $n$ vertices with edge expansion $h(X_n) \geq h_0$, as guaranteed by Lemma 2.2. Fix $d = d_0 + 1$. We replace each vertex $v$ with a copy of $X_{d_v}$ where $d_v$ denotes the degree of $v$ in $G$. Denote the vertices of $X_{d_v}$ by $[v]$ and let $[V] = \cup_v [v]$. Denote the union of the edges of $X_{d_v}$ for all $v$ by $E_1$, and place equality constraints on these edges.

In addition, for every edge $(v,w) \in E$ we will put an edge between one vertex in $[v]$ and one vertex in $[w]$ so that each vertex in $[V]$ sees exactly one such external edge. Denote these edges $E_2$. Altogether $G' = ([V], \mathbf{E} = E_1 \cup E_2)$ is a $d$-regular graph, and $|\mathbf{E}| = d|E|$.

We analyze $\overline{\text{SAT}}(G')$. The (completeness) upper bound $\overline{\text{SAT}}(G') \leq \overline{\text{SAT}}(G)$ is easy: An assignment $\sigma : V \to \Sigma$ can be extended to $[V]$ by assigning each $x \in [v]$ the value $\sigma(v)$. This will cause the same number of edges to reject, which can only decrease as a fraction.

For the (soundness) lower bound, let $\sigma' : [V] \to \Sigma$ be a 'best' assignment, i.e. violating the fewest constraints, $\overline{\text{SAT}}_{\sigma'}(G') = \overline{\text{SAT}}(G') \overset{\triangle}{=} \alpha$. Define $\sigma : V \to \Sigma$ according to plurality of $\sigma'$, i.e., let $\sigma(v)$ be the most popular value among $(\sigma'(x))_{x \in [v]}$. Let $S \subseteq [V]$ be the set of vertices whose value disagrees with the plurality. Let $E^* \subseteq E$ be the edges that reject $\sigma$, and let $\mathbf{E}^*$ be the edges that reject $\sigma'$. The external edge corresponding to an $e \in E^*$ either rejects $\sigma'$, or has at least one endpoint in $S$. So

$$|\mathbf{E}^*| + |S| \geq |E^*| = \alpha |E|$$

If $|\mathbf{E}^*| \geq \frac{\alpha}{2}|E|$ we are done since $\frac{\alpha}{2}|E| = \frac{\alpha}{2d}|\mathbf{E}|$ and so $\overline{\text{SAT}}(G') \geq \overline{\text{SAT}}(G)/2d$. So assume that $|S| \geq \frac{\alpha}{2}|E|$. Focus on one $[v]$, and let $S^v = [v] \cap S$. We can write $S^v$ as a disjoint union of sets

$S_a = \{x \in S^v \mid \sigma'(x) = a\}$. Since $S$ is the set of vertices disagreeing with the plurality value, we have $|S_a| \le \frac{1}{|\Sigma|} \le 1/2$, so by the edge expansion of the appropriate expander $X_{d_v}$, $E(S_a, \bar{S_a}) \ge h_0 \cdot |S_a|$. All of these edges carry equality constraints that reject $\sigma'$. So there are at least $h_0 \sum_v |S \cap [v]| = h_0 |S| \ge \frac{\alpha h_0}{2} |E|$ edges that reject $\sigma'$. Since $|E| = |\mathbf{E}|/d$, we get $\overline{\text{SAT}}(G') \ge \frac{h_0}{2d}\overline{\text{SAT}}(G)$. We have completed the proof, with $c = \min(\frac{1}{2d}, \frac{h_0}{2d})$. ■

**Lemma 3.3 (Expanderizing)** *Let $d_0, h_0 > 0$ be some global constants. Any $d$-regular constraint graph $G$ can be transformed into $G'$ such that*

- *$G'$ is $(d + d_0 + 1)$-regular, has self-loops, and $\lambda(G') \le d + d_0 + 1 - \frac{{h_0}^2}{d+d_0+1} < deg(G')$.*
- *$size(G') = O(size(G))$*
- *$\frac{d}{d+d_0+1} \cdot \overline{\text{SAT}}(G) \le \overline{\text{SAT}}(G') \le \overline{\text{SAT}}(G)$*

**Proof:** The idea is to add to $G$ self-loops and edges of an expander and put void constraints on these new edges (i.e., constraints that are satisfied always). By convention, a self loop adds 1 to the degree of a vertex. Let $X = (V, E')$ be a $d_0$-regular expander on $|V|$ vertices, with $h(X) \ge h_0$ (again, as guaranteed by Lemma 2.2). Let $E_{loop} = \{(v, v) \mid v \in V\}$. Let $G' = (V, E \cup E' \cup E_{loop})$, where the constraints associated with non-$E$ edges are void constraints (satisfied always). Clearly the degree is $d + d_0 + 1$. To bound $\lambda(G')$ we rely on the following well-known inequality (see Lemma 2.3),

$$\lambda(G) \le d(G) - \frac{h(G)^2}{d(G)} .$$

Clearly $h(G') \ge h(X) \ge h_0$, so plugging $G'$ in the above yields $\lambda(G') \le d + d_0 + 1 - \frac{{h_0}^2}{d+d_0+1} < d + d_0 + 1$.

Finally, since the new edges are always satisfied and since we increased the total number of edges by factor $c' = \frac{d+d_0+1}{d}$, the fraction of unsatisfied constraints drops by at most $c'$. ■

**Proof:**(of Lemma 3.1) First apply Lemma 3.2 on $G$, and then apply Lemma 3.3 on the result. The lemma is proven with $\beta_1 = c \cdot \frac{d}{d+d_0+1}$. ■

### 3.2 Powering

This operation is a new operation on constraint systems, and it is the one that gains us the gap. Let $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ be a constraint graph , and let $t \in \mathbb{N}$. We define $G^t = \langle (V, \mathbf{E}), \Sigma^{d^{\lceil t/2 \rceil}}, \mathcal{C}^t \rangle$ to be the following constraint graph:

- The vertices of $G^t$ are the same as the vertices of $G$.
- Edges: $u$ and $v$ are connected by $k$ edges in $\mathbf{E}$ if the number of $t$-step paths from $u$ to $v$ in $G$ is exactly $k$.
- Alphabet: The alphabet of $G^t$ is $\Sigma^{d^{\lceil t/2 \rceil}}$, where every vertex specifies values for all of its neighbors reachable in $t/2$ steps. One may think of this value as describing $v$'s opinion of its neighbors' values.
- Constraints: The constraint associated with an edge $\mathbf{e} = (u, v) \in \mathbf{E}$ is satisfied iff the assignments for $u$ and $v$ are consistent with an assignment that satisfies all of the constraints induced by the $t/2$ neighborhoods of $u$ and $v$.

If $\textsc{sat}(G) = 1$ then clearly $\textsc{sat}(G^t) = 1$. More interestingly,

**Lemma 3.4 (Amplification Lemma)** *Let $\lambda < d$, and $|\Sigma|$ be arbitrary constants. There exists a constant $\beta_2 = \beta_2(\lambda, d, |\Sigma|) > 0$, such that for every $t \in \mathbb{N}$ and for every $d$-regular constraint graph $G = \langle (V,E), \Sigma, \mathcal{C} \rangle$ with self-loops and $\lambda(G) \le \lambda$,*

$$\overline{\textsc{sat}}(G^t) \ge \beta_2 \sqrt{t} \cdot \min\left( \overline{\textsc{sat}}(G)\, , \, \frac{1}{t} \right) .$$

This is our main technical lemma, and its proof is given in Section 5.

### 3.3 Composition

In this section we describe a transformation on constraint graphs that reduces the alphabet size, while roughly maintaining the gap. We rely on *composition* which is an essential component in PCP constructions, described next. To understand composition let us ignore the underlying graph structure of a constraint graph $G$, and view it simply as a system of constraints.

Let us step back for a moment and recall our overall goal of proving the PCP Theorem. What we seek is a reduction from (say) 3SAT to gap-3SAT. Such a reduction is a polynomial-time algorithm that inputs some 3SAT formula on $n$ Boolean variables, and generates a new system of 3CNF clauses with the following gap property: Satisfiable inputs translate to satisfiable 3SAT systems, and unsatisfiable inputs translate to 3SAT systems that are only $1 - \alpha$ satisfiable (i.e. any assignment can satisfy only $1 - \alpha$ of the clauses), for some $\alpha > 0$.

With these “gap-generating” reductions in mind, one can imagine how to make use of composition. Suppose we had such a gap-generating reduction whose output size is exponential in the input size[3]. We could potentially use it as a subroutine in a (polynomial-time) gap-generating reduction, making sure to run it on inputs that are sufficiently small ($\le \log n$). This is the basic idea of composition.

How would this work with constraint graphs? Assume we have a gap-generating reduction $\mathcal{P}$ as above, and let $G$ be a constraint graph. We can put each constraint of $G$ in 3SAT form, and then feed it to $\mathcal{P}$. The output would be a system of 3SAT clauses, which can be easily viewed as constraints over a small alphabet $\le 2^3 = 8$. The new system would be the union of the 3SAT systems output by $\mathcal{P}$ over all of $G$’s constraints. Thus we have achieved our goal of reducing the size of the alphabet, from $\Sigma$, $|\Sigma| = O(1)$, to $|\Sigma_0| \le 2^3$. The main point is that as long as $|\Sigma| = O(1)$, $\mathcal{P}$ can be allowed to be as inefficient as needed, and still this composition would only incur a linear overhead.

There is one subtle point that has been ignored so far. It is well-known that for composition to work, consistency must be established between the many invocations of $\mathcal{P}$. This point has been handled before in a modular fashion by adding additional requirements on the reduction $\mathcal{P}$. Such more-restricted reductions are called PCPs of Proximity in [BGH+04] or Assignment Testers in [DR04]. We describe these formally below. For an exposition as to why these objects are well-suited for composition, as well as a proof of a generic composition theorem, please see [BGH+04, DR04].

The following is a stripped-down version of the definition of [DR04], that suffices for our purposes:

**Definition 3.1 (2-Query Assignment Tester)** *A 2-Query Assignment Tester with parameters $\delta_0, \Sigma_0$ is a reduction algorithm $\mathcal{P}$ whose input is a Boolean constraint $\varphi$ over a set of Boolean variables $X$. $\mathcal{P}$ outputs a system of constraints $\Psi$ over variables $X$ and auxiliary variables $Y$ such that*

- *The variables in $Y$ take values in an alphabet $\Sigma_0$.*

[3] The implicit assumption here is that such reductions are significantly easier to construct, see e.g. Section 6.

- *Each $\psi \in \Psi$ is over two variables from $X \cup Y$.*
- *For every assignment $a : X \to \{0,1\}$,*
  1. *[Completeness:] If $a$ satisfies $\varphi$ then there exists an assignment $b : Y \to \Sigma_0$ such that $a \cup b$ satisfies every constraint in $\Psi$.*
  2. *[Soundness:] For every $\delta \le \delta_0$, if $a$ is $\delta$-far[4] from every satisfying assignment for $\varphi$, then for every assignment $b : Y \to \Sigma_0$, at least $\Omega(\delta)$ of the constraints in $\Psi$ reject $a \cup b$.*

Notice that no restriction is imposed on the running time of $\mathcal{P}$ or on $|\Psi|$. In particular, we ignored the format of the input to $\mathcal{P}$, which may as well be a truth table. We describe an explicit construction of such an algorithm in Section 6 (see Lemma 6.2). As mentioned earlier, such an algorithm (that works only on inputs of some fixed bounded size) can also be found by exhaustive search, provided we have proven its existence independently. Our main lemma in this section is the following,

**Lemma 3.5 (Composition)** *Assume the existence of a 2-query assignment tester $\mathcal{P}$, with $\delta_0 > 0$ and alphabet $\Sigma_0$, $|\Sigma_0| = O(1)$. There exists $\beta_3 > 0$ that depends only on $\mathcal{P}$, such that any constraint system $G = \langle V, \Sigma, \mathcal{C} \rangle$ can be transformed into a constraint system $G' = \langle V', \Sigma_0, \mathcal{C}' \rangle$, denoted $G \circ \mathcal{P}$, such that $size(G') = M(|\Sigma|) \cdot size(G)$, and*

$$\beta_3 \cdot \overline{\mathrm{SAT}}(G) \le \overline{\mathrm{SAT}}(G') \le \overline{\mathrm{SAT}}(G)$$

**Proof:** We describe the construction in two steps:

- (Robustization:) Let $\mathrm{e} : \Sigma \to \{0,1\}^{\ell}$ be any error-correcting-code with linear rate and relative distance $\rho > 0$ , so $\ell = O(\log |\Sigma|)$. Replace each variable $v \in V$ by a set of $\ell$ Boolean variables denoted $[v]$. These are supposed to represent the encoding via $\mathrm{e}$ of $v$'s assignment. Replace each constraint $c \in \mathcal{C}$ by a constraint $\tilde{c}$ over variables $[v] \cup [w]$. $\tilde{c}$ is satisfied iff the assignment for $[v] \cup [w]$ is the legal encoding via $\mathrm{e}$ of an assignment for $v$ and $w$ that would have satisfied $c$.
- (Composition:) Run an assignment tester $\mathcal{P}$ on each $\tilde{c}$. This makes sense since $\tilde{c}$ is a Boolean constraint over Boolean variables $[v] \cup [w]$. Let $Y_c$ be the resulting set of auxiliary variables, and let $\Psi_c$ the resulting set of constraints. Define the new constraint system $G' = \langle (V', E'), \Sigma_0, \mathcal{C}' \rangle$ as follows:
$$V' = \bigcup_{v \in V} [v] \cup \bigcup_{c \in \mathcal{C}} Y_c \,, \qquad \mathcal{C}' = \cup_{c \in \mathcal{C}} \Psi_c$$
(where $\mathcal{C}'$ implicitly defines $E'$).

First, let us verify that $size(G') = M(|\Sigma|) \cdot size(G)$. The inputs fed into $\mathcal{P}$ are constraints $\tilde{c} : \{0,1\}^{O(\ell)} \to \{\mathsf{T}, \mathsf{F}\}$. There is a finite number of these, at most $2^{2^{O(\ell)}}$. Let $M$ denote the maximal size of the output of $\mathcal{P}$ over all such inputs. Clearly, $size(G') \le M \cdot size(G)$ and $M$ is a constant that depends only on $\Sigma$ and $\mathcal{P}$.

It remains to be seen that $\beta_3 \cdot \overline{\mathrm{SAT}}(G) \le \overline{\mathrm{SAT}}(G') \le \overline{\mathrm{SAT}}(G)$. The proof is simple and follows exactly the proof of the composition theorem in [DR04]. Let us sketch the first inequality (that corresponds to the soundness argument). Given an assignment $\sigma' : V' \to \Sigma_0$, we extract from it an assignment $\sigma : V \to \Sigma$ by letting for each $v \in V$ $\sigma(v)$ to be a value whose encoding via $\mathrm{e}$ is closest to $\sigma'([v])$. By definition, a fraction $\overline{\mathrm{SAT}}_{\sigma}(G) \ge \overline{\mathrm{SAT}}(G)$ of constraints reject $\sigma$. Let $c \in \mathcal{C}$ be a constraint over variables $u, v$ that rejects $\sigma$. We will show that a constant fraction of the constraints in $\Psi_c$ reject $\sigma'$. The

[4] Two assignments $a, a'$ are $\delta$-far if $\Pr_x[a(x) \neq a'(x)] \ge \delta$.

main observation is that the input to $\tilde{c}$ (i.e., the restriction of $\sigma'$ to $[u] \cup [v]$) is at least $\rho/4$-far from a satisfying input (where $\rho$ denotes the code-distance of $\mathrm{e}$). The reason is that a $\rho/2$ fraction of the bits in either $[u]$ or $[v]$ (or both) must be changed in order to change $\sigma'$ into an assignment that satisfies $\tilde{c}$. Set $\delta = \min(\delta_0, \rho/4)$. By the soundness property of $\mathcal{P}$, at least $\Omega(\delta)$ fraction of $\Psi_c$ reject. Altogether, $\overline{\text{SAT}}(G') \geq \Omega(\delta) \cdot \overline{\text{SAT}}(G) = \beta_3 \cdot \overline{\text{SAT}}(G)$ for some $\beta_3 > 0$.

■

# 4 Main Theorem

Based on the constraint graph operations described in the previous section, we are now ready to prove our main theorem.

**Theorem 4.1 (Main)** *For any $\Sigma$, $|\Sigma| = O(1)$, there exists constants $C > 0$ and $0 < \alpha < 1$, such that given a constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ one can construct, in polynomial time, a constraint graph $G' = \langle (V', E'), \Sigma_0, \mathcal{C}' \rangle$ such that*

- *$size(G') \leq C \cdot size(G)$ and $|\Sigma_0| = O(1)$.*
- *(Completeness:) If $\text{SAT}(G) = 1$ then $\text{SAT}(G') = 1$*
- *(Soundness:) $\overline{\text{SAT}}(G') \geq \min(2 \cdot \overline{\text{SAT}}(G), \alpha)$.*

**Proof:** We construct $G'$ from $G$ by

$$G' = (\, prep(G)\, )^t \circ \mathcal{P}$$

for an appropriately selected constant $t \in \mathbb{N}$. Let us break this into three steps:

1. (Preprocessing step:) Let $H_1 = prep(G)$ be the result of applying Lemma 3.1 to $G$.

   So there exists some global constants $\lambda < d$ and $\beta_1 > 0$ such that $H_1$ is $d$-regular, has the same alphabet as $G$, $\lambda(H_1) \leq \lambda < d$, and $\beta_1 \cdot \overline{\text{SAT}}(G) \leq \overline{\text{SAT}}(H_1) \leq \overline{\text{SAT}}(G)$.

2. (Amplification step:) Let $H_2 = (H_1)^t$, for a large enough constant $t > 1$ to be specified below.

   According to Lemma 3.4, there exists some constant $\beta_2 = \beta(\lambda, d, |\Sigma|) > 0$ for which $\overline{\text{SAT}}(H_2) \geq \beta_2 \sqrt{t} \cdot \min(\overline{\text{SAT}}(H_1), \frac{1}{t})$. However, the alphabet grows to $\Sigma^{d^{\lceil t/2 \rceil}}$.

3. (Composition step:) Let $G' = H_2 \circ \mathcal{P}$ be the result of applying Lemma 3.5 to $H_2$ relying on a 2-query assignment tester $\mathcal{P}$, as guaranteed in Lemma 6.2.

   This reduces the alphabet to $\Sigma_0$ while still $\beta_3 \cdot \overline{\text{SAT}}(H_2) \leq \overline{\text{SAT}}(G') \leq \overline{\text{SAT}}(H_2)$, for a constant $\beta_3 > 0$.

Let us verify the properties claimed above. The size of $G'$ is linear in the size of $G$ because each step incurs a linear blowup. Specifically, in step 2, since $deg(H_1) = d$ and $t = O(1)$, the number of edges in $H_2 = (H_1)^t$ is equal to the number of edges in $H_1$ times a constant factor of $d^{t-1}$. In step 3, the total size grows by a factor $M$ that depends on the alphabet size of $H_2$, which equals $|\Sigma^{d^{\lceil t/2 \rceil}}| = O(1)$, and on $\mathcal{P}$ which is fixed throughout the proof, so $M$ is constant.

Completeness is clearly maintained at each step. Choose now $t = \lceil (\frac{2}{\beta_1\beta_2\beta_3})^2 \rceil$, and let $\alpha = \beta_3\beta_2/\sqrt{t}$. Altogether,

$$\begin{aligned}
\overline{\text{SAT}}(G') &\geq \beta_3 \cdot \overline{\text{SAT}}(H_2) && \text{(step 3, Lemma 3.5)}\\
&\geq \beta_3 \cdot \beta_2\sqrt{t} \cdot \min(\overline{\text{SAT}}(H_1), \frac{1}{t}) && \text{(step 2, Lemma 3.4)}\\
&\geq \beta_3 \cdot \beta_2\sqrt{t} \cdot \min(\beta_1\overline{\text{SAT}}(G), \frac{1}{t}) && \text{(step 1, Lemma 3.1)}\\
&\geq \min(2 \cdot \overline{\text{SAT}}(G), \alpha)
\end{aligned}$$

■

As a corollary of the main theorem we get,

**Corollary 4.2 (PCP Theorem)** *Gap-3SAT is NP-hard. (Alternatively, $SAT \in PCP_{\frac{1}{2},1}[O(\log n), O(1)]$).*

**Proof:** We reduce from constraint graph satisfiability. The basic idea is to repeatedly apply the main theorem until after $\log n$ iterations the gap is a constant fraction.

According to Proposition 2.1 it is NP-hard to decide if for a given constraint graph $G_0$ with $|\Sigma| \leq 7$, $\text{SAT}(G_0) = 1$ or not. Fix such a $G_0$.

Let $G_i$ $(i \geq 1)$ be the outcome of applying the main theorem on $G_{i-1}$. Then for $i \geq 1$ $G_i$ is a constraint graph with alphabet $\Sigma_0$. Let $k = \log |E(G_0)| = O(\log n)$. Observe that the size of $G_i$ for $i \leq k = O(\log n)$ is bounded by $C^i \cdot size(G_0) = \text{poly}(n)$.

Completeness is easy: if $\text{SAT}(G_0) = 1$ then $\text{SAT}(G_i) = 1$ for all $i$. For soundness, assume $\text{SAT}(G_0) < 1$. If for some $i^* < k$, $\overline{\text{SAT}}(G_{i^*}) \geq \alpha/2$ then the main theorem implies that for all $i > i^*$ $\overline{\text{SAT}}(G_i) \geq \alpha$. For all other $i$ it follows by induction that

$$\overline{\text{SAT}}(G_i) \geq \min(2^i\, \overline{\text{SAT}}(G_0), \alpha)\,.$$

If $\overline{\text{SAT}}(G_0) > 0$ then $\overline{\text{SAT}}(G_0) \geq \frac{1}{|E(G_0)|}$, so surely $2^k\overline{\text{SAT}}(G_0) > \alpha$. Thus $\overline{\text{SAT}}(G_k) \geq \alpha$.

Finally, a local gadget reduction takes $G_k$ to 3SAT form (by converting each constraint into a constant number of 3CNF clauses), while maintaining the gap up to some constant. To get to soundness of $\frac{1}{2}$, in the $SAT \in PCP_{\frac{1}{2},1}[O(\log n), O(1)]$ version, one can apply simple (sequential) repetition.

■

## 5 Soundness Amplification Lemma

**Lemma 3.4** *Let $\lambda < d$, and $|\Sigma|$ be arbitrary constants. There exists a constant $\beta_2 = \beta_2(\lambda, d, |\Sigma|) > 0$, such that for every $t \in \mathbb{N}$ and for every $d$-regular constraint graph $G = \langle (V, E), \Sigma, \mathcal{C}\rangle$ with self-loops and $\lambda(G) \leq \lambda$,*

$$\overline{\text{SAT}}(G^t) \geq \beta_2\sqrt{t} \cdot \min\left(\overline{\text{SAT}}(G)\,, \frac{1}{t}\right)\,.$$

Throughout the proof all constants including $O(\cdot)$ and $\Omega(\cdot)$ notation are independent of $t$ but may depend on $d$, $\lambda$ and $|\Sigma|$. Let us assume for notation clarity that $t$ is even.

The idea of the proof is as follows. Let us refer to the edges of $G^t$ as *paths*, since they come from $t$-step paths in $G$, and let us refer to the edges of $G$ as edges. An assignment for $G^t$, is a mapping $\vec{\sigma} : V \to \Sigma^{d^{t/2}}$ where each vertex specifies $\Sigma$ values for itself as well as all of its neighbors at distance $\leq t/2$. Let us define a new assignment mapping $V$ into $\Sigma$, by assigning each vertex the most popular value among the values assigned to it by its neighbors. The probability that a random edge rejects this new assignment is, by definition, at least $\overline{\text{SAT}}(G)$. We will show that the probability that a random path rejects $\vec{\sigma}$ is $\Omega(\sqrt{t})$ times larger (we say that a path rejects $\vec{\sigma}$ if the constraint on it is not satisfied by $\vec{\sigma}$).

This is done by charging to each rejecting edge, all rejecting paths that pass through it. The main point is that each rejecting edge is potentially responsible for $t \cdot d^{t-1}$ rejecting paths, while the total number of edges in $G^t$ is only a factor $d^{t-1}$ larger than that in $G$.

We first show (Lemma 5.1) that paths in which one of the 'middle' edges is rejecting, have a constant probability of rejecting themselves. Middle edges are those that are traversed by the path at step $i \in \left\{t/2 - \sqrt{t}, \ldots, t/2 + \sqrt{t}\right\}$. Hence we end up charging $\Theta(\sqrt{t}) \cdot d^{t-1}$ rejecting paths to each rejecting edge. We then show (Lemma 5.3) that almost all paths are charged no more than $O(1)$ times. This is where the expansion property of $G$ is used, as we prove that a random path is expected to pass through any (small) fixed set of edges a number of times that is a constant independent of $t$.

**Proof:** Denote by $\mathbf{E} = E(G^t)$ the edge set of $G^t$. An edge $\mathbf{e} = (e_1, \ldots, e_t) \in \mathbf{E}$ is a path of length $t$ in $G$. Since $G$ is $d$-regular, $G^t$ is $d^t$-regular, and $|\mathbf{E}| = d^{t-1}\,|E|$.

Let $\vec{\sigma} : V \to \Sigma^{d^{t/2}}$ be a 'best' assignment for $G^t$, $\text{SAT}(G^t) = \text{SAT}_{\vec{\sigma}}(G^t)$. For each $v$, $\vec{\sigma}(v)$ assigns values for $v$ and every vertex $w$ within distance $\leq t/2$ of $v$. We denote $\vec{\sigma}(v)_w \in \Sigma$ the restriction of $\vec{\sigma}(v)$ to $w$. This can be thought of as the opinion of $v$ about $w$. Define an assignment $\sigma : V \to \Sigma$ as follows. For every $j = 1, \ldots, t$ let $X_{v,j}$ be a random variable that assumes a value $a$ with probability that a $j$-step random walk from $v$ ends at a vertex $w$ for which $\vec{\sigma}(w)_v = a$. Define $\sigma(v) = a$ for a value $a$ which maximizes $\Pr[X_{v,t/2} = a]$.

Let $F$ be a subset of edges that reject $\sigma$, so that $\frac{|F|}{|E|} = \min(\overline{\text{SAT}}_\sigma(G), \frac{1}{t})$. Let

$$\Gamma_{i,e} = \{\mathbf{e} \in \mathbf{E} \mid \mathbf{e}_i = e\}\,, \quad \Gamma_i = \bigcup_{e \in F} \Gamma_{i,e}, \quad \Gamma = \bigcup_i \Gamma_i\,.$$

Also denote $\Gamma_i^* = \{\mathbf{e} \in \Gamma_i \mid \mathbf{e} \text{ rejects } \vec{\sigma}\}$ and $\Gamma^* = \cup \Gamma_i^*$. The set $\Gamma^*$ contains all rejecting paths that pass through $F$, but there can be other rejecting paths as well, so $\frac{|\Gamma^*|}{|\mathbf{E}|} \leq \overline{\text{SAT}}_{\vec{\sigma}}(G^t) = \overline{\text{SAT}}(G^t)$. We will show that for some $\beta > 0$, $\beta\sqrt{t} \cdot \frac{|F|}{|E|} \leq \frac{|\Gamma^*|}{|\mathbf{E}|}$ so

$$\begin{aligned}
\beta\sqrt{t} \cdot \min(\overline{\text{SAT}}(G), \frac{1}{t}) &\leq \beta\sqrt{t} \cdot \min(\overline{\text{SAT}}_\sigma(G), \frac{1}{t}) \\
&= \beta\sqrt{t} \cdot \frac{|F|}{|E|} \\
&\leq \frac{|\Gamma^*|}{|\mathbf{E}|} \\
&\leq \overline{\text{SAT}}_{\vec{\sigma}}(G^t) = \overline{\text{SAT}}(G^t)\,.
\end{aligned}$$

where the middle equality follows from the definition of $F$. We next show that a random path whose $i$th step equals some fixed $e \in F$ has constant probability of rejecting,

**Lemma 5.1** *Let $I = \left\{\frac{t}{2} - \sqrt{t} < i \leq \frac{t}{2} + \sqrt{t}\right\} \subset \mathbb{N}$. There exists some $\alpha > 0$ that depends only on $|\Sigma|$ and $d$, such that for every $e \in F$, and every $i \in I$*

$$\Pr_{\mathbf{e} \in \Gamma_{i,e}}\left[\mathbf{e} \in \Gamma_{i,e}^*\right] > \alpha\,.$$

**Proof:** Fix $e = (u, v) \in F$, and let $a = \sigma(u)$ and $b = \sigma(v)$. Since $e \in F$, the constraint on $e$ rejects the values $a$ and $b$. We show that with constant probability, a random path in $\Gamma_{i,e}$ starts at a vertex $u_0$ for which $\vec{\sigma}(u_0)_u = a$, and ends at a vertex $u_t$ for which $\vec{\sigma}(u_t)_v = b$. This implies, by definition, that the path rejects $\vec{\sigma}$. The reason is that if the path is randomly chosen in $\Gamma_{i,e}$, then the endpoints $u_0$ and $u_t$ are sufficiently random, and will see the plurality opinion often enough.

Fix $i \in I$. The idea is that every path $\mathbf{e} \in \Gamma_{i,e}$ can be written uniquely as $\mathbf{e} = \mathbf{p}_1 e \mathbf{p}_2$ where $\mathbf{p}_1, \mathbf{p}_2$ are paths of lengths $i - 1$ and $t - i$:

$$\mathbf{p}_1 = ((u_1, u_2), (u_2, u_3), \ldots, (u_{i-1}, u)) \quad \text{and} \quad \mathbf{p}_2 = ((v, v_1), (v_1, v_2), \ldots, (v_{t-i-1}, v_{t-i}))$$

Moreover, this path clearly rejects if $\vec{\sigma}(u_1)_u = a$ and $\vec{\sigma}(v_{t-i})_v = b$. These events are $X_{u,i-1} = a$ and $X_{v,t-i} = b$ respectively. Moreover, since the choice of $\mathbf{p}_1$ is independent of the choice of $\mathbf{p}_2$, we have $\Pr_{\Gamma_{i,e}}\left[\Gamma^*_{i,e}\right] \geq \Pr[X_{u,i-1} = a] \cdot \Pr[X_{v,t-i} = b]$.

Observe that by definition $\Pr[X_{u,t/2} = a] \geq \frac{1}{|\Sigma|}$ and $\Pr[X_{u,t/2} = b] \geq \frac{1}{|\Sigma|}$, since $a, b$ are the most popular values for $u, v$ respectively. Had it been possible that $i - 1 = t/2$ and $t - i = t/2$, the lemma would follow immediately from the definition of $\sigma$, taking $\alpha = \frac{1}{|\Sigma|^2}$. We will prove that for all $\ell$

$$\text{If} \quad |\ell - t/2| \leq \sqrt{t} \quad \text{then} \quad \Pr[X_{u,\ell} = a] > \frac{\tau}{2} \cdot \Pr[X_{u,t/2} = a] \tag{1}$$

for some $\tau > 0$ to be determined, and a symmetric argument will hold for $\Pr[X_{v,\ell} = b]$. The intuition for (1) is that the self-loops of $G$ make the distribution of vertices reached by a random $t/2$-step walk from $u$ roughly the same as distribution on vertices reached by an $\ell$-step walk from $u$, for $\ell \in I$.

Mark one self-loop on each vertex, and observe that any length-$\ell$ path from $u$ in $G$ can be equivalently described by (i) specifying in which steps the marked edges were traversed, and then (ii) specifying the remaining steps conditioned on choosing only non-marked edges. Let $X'_{u,k}$ be a random variable that assumes a value $\sigma$ with probability that a $k$-step random walk *conditioned on walking only on non-marked edges* reaches a vertex $w$ for which $\vec{\sigma}(w)_u = \sigma$. In other words, for a binomial variable $B_{\ell,p}$ with $\Pr[B_{\ell,p} = k] = \binom{\ell}{k} p^k (1-p)^{\ell-k}$ and $p = 1 - 1/d$,

$$\Pr[X_{u,\ell} = a] = \sum_{k=0}^{\ell} \Pr[B_{\ell,p} = k] \Pr[X'_{u,k} = a]\,. \tag{2}$$

The point is that if $|\ell_1 - \ell_2|$ is small, then the distributions $B_{\ell_1,p}$ and $B_{\ell_2,p}$ are similar, as formalized in the following lemma:

**Lemma 5.2** *For every $p \in [0, 1]$ and $c > 0$ there exists some $0 < \tau < 1$ such that if $n - \sqrt{n} \leq m < n$, then*

$$\forall k, |k - pn| \leq c\sqrt{n}, \qquad \tau \leq \frac{\Pr[B_{n,p} = k]}{\Pr[B_{m,p} = k]} \leq \frac{1}{\tau}$$

The proof is a straightforward computation, and can be found in Appendix A. The lemma implies that the distributions of $X_{u,\ell_1}$ and $X_{u,\ell_1}$ are similar. Indeed let us choose $c$ so that $\Pr[B_{\frac{t}{2},p} \notin I] \leq \frac{1}{2|\Sigma|}$ for the set $I = \left\{k \;\middle|\; |k - p\ell| \leq c\sqrt{t}\right\}$; and let $\tau$ be the appropriate constant from the lemma. Clearly $c$ can be chosen independently of $t$ since $k \notin I$ implies $|k - pt/2| \geq |k - p\ell| - |p\ell - pt/2| > (c-1)\sqrt{t}$. We now have

$$\begin{aligned}\Pr[X_{u,\ell}=a] &\geq \sum_{k\in I}\Pr[B_{\ell,p}=k]\Pr[X'_{u,k}=a] \\ &\geq \tau\cdot\sum_{k\in I}\Pr[B_{t/2,p}=k]\Pr[X'_{u,k}=a] \\ &\geq \tau\cdot\left(\Pr[X_{u,t/2}=a]-\frac{1}{2\,|\Sigma|}\right)\geq\frac{\tau}{2}\cdot\Pr[X_{u,t/2}=a]\end{aligned}$$

where the last inequality holds since $\Pr[X_{u,t/2}=a]\geq\frac{1}{|\Sigma|}$. So (1) is established, and the proof of Lemma 5.1 is complete with $\alpha=\left(\frac{\tau}{2}\cdot\frac{1}{|\Sigma|}\right)^2$. ■

It is easy to verify that $|\Gamma_i|=|F|\cdot|\Gamma_{i,e}|=|F|\,d^{t-1}$ since $\Gamma_i=\cup_{e\in F}\Gamma_{i,e}$ and this is a disjoint union. By Lemma 5.1,

$$\forall i\in I\qquad |\Gamma^*_i|=\sum_{e\in F}|\Gamma^*_{i,e}|\geq|F|\,\alpha\,|\Gamma_{i,e}|=\alpha\,|\Gamma_i|$$

Summing over all $i\in I$,

$$\sum_{i\in I}|\Gamma^*_i|\geq\alpha\sum_{i\in I}|\Gamma_i|\geq\alpha\,|I|\,|F|\,d^{t-1}. \tag{3}$$

If we divide the right hand side by $|\mathbf{E}|$, we get $\overline{\text{SAT}}(G)$ times $\alpha\,|I|=\Omega(\sqrt{t})$. Intuitively, the left hand side should more-or-less equal the number of rejecting paths in $G^t$, which lower bounds $\overline{\text{SAT}}(G^t)$. If that were so, we would get the desired lower bound. However, a path $\mathbf{e}$ will be over-counted in the left-hand side if it belongs to $\Gamma^*_i$ for more than one $i\in I$, or in other words, if it steps in $F$ more than once. The second part of the proof uses the expansion of $G$ to bound this event.

For any path $\mathbf{e}$, let $N_F(\mathbf{e})$ be the number of steps $\mathbf{e}$ takes in $F$, $N_F(\mathbf{e})=|\{\mathbf{e}\cap F\}|$. For clarity we omit $F$ from the notation and write $N(\mathbf{e})$. Fix $M>0$ to be specified later, and let $B_i$ be the set of paths that are overcounted in (3),

$$1\leq i\leq t,\quad B_i=\{\mathbf{e}\in\Gamma_i\,|\,N(\mathbf{e})\geq M\}\ . \tag{4}$$

Then

$$\sum_{i\in I}|\Gamma^*_i\setminus B_i|\leq\sum_{\mathbf{e}\in\Gamma^*\setminus(\cup_iB_i)}N(\mathbf{e})\leq M\,|\Gamma^*\setminus(\cup_iB_i)|\leq M\,|\Gamma^*| \tag{5}$$

We will show that $B_i$ is a small fraction of $\Gamma_i$,

**Lemma 5.3** *There exists $M>0$ such that for all $1\leq i\leq t$, and for $B_i$ defined in (4), $\Pr_{\Gamma_i}[B_i]\leq\alpha/2$.*

This will conclude the proof because

$$\sum_{i\in I}|\Gamma^*_i\setminus B_i|=\sum_{i\in I}(|\Gamma^*_i|-|B_i|)\geq|I|\,|F|\,d^{t-1}(\alpha-\alpha/2)\geq|F|\,d^{t-1}\cdot\alpha\sqrt{t} \tag{6}$$

where the first inequality follows from Equation (3) and Lemma 5.3. Combining Equations (5) and (6),

$$|\Gamma^*|\geq\frac{\alpha}{M}\cdot\sqrt{t}\,|F|\,d^{t-1}$$

which proves Lemma 3.4 with $\beta=\alpha/M$. It remains to prove Lemma 5.3. We will first prove that for all $j$, the expected value of $N(\mathbf{e})$ over $\mathbf{e}\in\Gamma_j$ is constant.

**Lemma 5.4** *For every $1 \le j \le t$, and every $F \subseteq E$, $\mathbb{E}_{\Gamma_j}[N] \le 2t\frac{|F|}{|E|} + O(1)$.*

**Proof:** Assume first $j = 1$. Define indicator variables $F_i(\mathbf{e})$ to equal $1$ if $\mathbf{e}_i \in F$ and $0$ otherwise. Clearly $N(\mathbf{e}) = \sum_{i=1}^{t} F_i(\mathbf{e})$, so by linearity of expectation,

$$\mathbb{E}_{\Gamma_1}[N] = \mathbb{E}_{\mathbf{e}\in\Gamma_1}\left[\sum_{i=1}^{t} F_i(\mathbf{e})\right] = \sum_{i=1}^{t} \Pr_{\mathbf{e}\in\Gamma_1}[F_i(\mathbf{e}) = 1] \,.$$

We now apply Proposition 2.4 to deduce that

$$\Pr_{\mathbf{e}\in\Gamma_1}[F_i(\mathbf{e}) = 1] \le \frac{|F|}{|E|} + \lambda^{i-1}$$

So

$$\mathbb{E}_{\Gamma_1}[N] \le \sum_{i=1}^{t}\left(\frac{|F|}{|E|} + \lambda^{i-1}\right) = t\frac{|F|}{|E|} + O(1)\,.$$

For $j > 1$ the estimates on $F_i$ are slightly different, but this at most doubles the expectation: $\mathbb{E}_{\Gamma_j}[F_i]$ is the probability over random paths whose $j$th step is a random edge in $F$, that the $i$th step is in $F$. If $i \ge j$ then we can ignore the part of the path before the $j$th step, and $\mathbb{E}_{\Gamma_j}[F_i] = \mathbb{E}_{\Gamma_1}[F_{i-j+1}]$. If $i < j$ think of the reverse path: choose a random $e \in F$ and then take $j - i$ random steps from it. So $\mathbb{E}_{\Gamma_j}[F_i] = \mathbb{E}_{\Gamma_1}[F_{j-i+1}]$. Altogether the expectation at most doubles. ■

Now it is easy to derive Lemma 5.3,

**Proof:** Let $N_0 = \max_j(\mathbb{E}_{\Gamma_j}[N])$. Since $|F|\,/\,|E| \le 1/t$, Lemma 5.4 implies $N_0 = O(1)$. Markov's inequality yields

$$\Pr_{\Gamma_j}[B_j] = \Pr_{\mathbf{e}\in\Gamma_j}\left[N(\mathbf{e}) \ge \frac{M}{N_0}\cdot N_0\right] \le N_0/M\,.$$

■

Choosing $M = \frac{2N_0}{\alpha}$ concludes the proof of Lemma 5.3, and thus also the proof of Lemma 3.4, with $\beta = \frac{\alpha}{M} = \frac{\alpha^2}{2N_0}$. ■

## 6 An Explicit 2-Query Assignment Tester

In this section we outline a construction of an assignment tester, as needed in Section 3.3. Let $\psi$ be a Boolean constraint over Boolean variables $x_1, \ldots, x_s$. We describe an algorithm $\mathcal{P}$ whose input is $\psi$ and whose output will be a system of constraints satisfying the requirements of Definition 3.1.

Let $L = \{f : \{0,1\}^s \to \{0,1\}\}$ be the set of all functions on $s$ bits, and define the encoding (via the Long-Code) of a string $a = (a_1, \ldots, a_s) \in \{0,1\}^s$ to be a table

$$A_a : L \to \{0,1\} \qquad \text{such that} \qquad \forall f,\; A_a(f) = f(a)\,.$$

The table-entries will be the variables. Recall that two tables $A, A' : L \to \{0,1\}$ are $\delta$-far from one another if $\Pr_f[A(f) \ne A'(f)] \ge \delta$.

**Theorem 6.1** *There exists a* Long-Code Test *such that for any $\psi : \{0,1\}^s \to \{0,1\}$,*

- *The test tosses some random coins, based on which it makes $3$ queries to a table $A : L \to \{0,1\}$.*
- *The test has perfect completeness: If $a \in \{0,1\}^s$ such that $\psi(a) = \mathsf{T}$, then the table $A_a$ satisfies the test with probability $1$.*

- *For every $0 < \delta < 1/4$, if a table $A : L \to \{0,1\}$ is $\delta$-far from $A_a$ for all $a$ for which $\psi(a) = \mathsf{T}$, then the test rejects with probability $\geq \Omega(\delta)$.*

For the sake of completeness, we include a proof of this theorem in Appendix B. In order to complete the construction we take two (rather standard) steps,

1. Add the variables $X = \{x_1, \ldots, x_s\}$. An assignment now is made of two tables $\sigma : X \to \{0,1\}$ and $A : L \to \{0,1\}$. Place two types of constraints as follows: The first type will be 3-variable constraints over variables indexed by $L$, as specified by the test in Theorem 6.1 (one constraint per outcome of random coin tosses). The second type is comparison constraints, defined as follows. For each choice of $i \in \{1, \ldots, s\}$ and $f \in L$ place a constraint that is satisfied iff $\sigma(x_i) = A(f) \oplus A(f + e_i)$. Assume wlog that there is an equal number of first-type constraints and second-type constraints[5].

2. Let $\psi_1, \ldots, \psi_N$ be the set of constraints defined in the previous step (so $N = 2^{O(2^s)}$). Introduce a new variable $z_i$ per $\psi_i$, and let $Z = \{z_i\}_i$. These variables will input values in $\{0,1\}^3$, and let $B : Z \to \{0,1\}^3$ be an assignment for these variables. The final system of constraints will be the following: there will be a constraint for every possible choice of $z_i \in Z$ and a variable $y$ of the three accessed by $\psi_i$ (so $y \in X \cup L$). This constraint will check that $B(z_i)$ satisfies $\psi_i$, and that it is consistent with the assignment for $y$.

**Lemma 6.2** *The constructed constraint system is a two-query assignment tester, with $\delta_0 = 1/4$ and $\Sigma_0 = \{0,1\}^3$.*

**Proof:** (sketch) Clearly the new constraints each access two variables from $X \cup L \cup Z$ (the variables $L \cup Z$ are the 'auxiliary' variables). Perfect completeness is evident. For soundness, assume that $\sigma : X \to \{0,1\}$ is an assignment which is $\delta$-far from every satisfying assignment for $\psi$. Let us first show that for every $A : L \to \{0,1\}$, the tables $\sigma, A$ cause at least $\Omega(\delta)$ of the constraints constructed at the end of step 1 to reject. First assume that $A : L \to \{0,1\}$ is $\delta$ far from a legal long-code encoding. Then by Theorem 6.1 at least $\Omega(\delta)$ of the long-code tests reject, and this is an $\Omega(\delta)$ fraction of all of the constraints. If $A$ is not $\delta$ far from a legal long-code encoding then it is $\delta$-close to the long-code encoding of some $\sigma' : X \to \{0,1\}$ which satisfies $\psi$. By assumption on $\sigma$, $\Pr_i[\sigma(x_i) \neq \sigma'(x_i)) > \delta$, so we claim $\Omega(\delta)$ fraction of the comparison constraints must reject. Indeed, for each $i$ we have

$$\Pr_{f \in L}\left[A(f) \oplus A(f + e_i) = f(\sigma') \oplus (f \oplus e_i)(\sigma')\right] \geq 1 - 2\delta \tag{7}$$

and if $\sigma(x_i) \neq \sigma'(x_i)$, then $f(\sigma') \oplus (f \oplus e_i)(\sigma') = \sigma'(x_i) \neq \sigma(x_i)$. Hence every $f$ that satisfies the equality in (7) causes the corresponding comparison constraint to reject. Altogether at least $(1 - 2\delta)\delta \geq \delta/2$ fraction of the constraints reject, in this case too.

Now consider the final system, generated in step 2. Let $B : Z \to \{0,1\}^3$. We have established that for every table $A$, the assignments $\sigma, A$ for $X \cup L$ must cause at least $\Omega(\delta)$ of the constraints to reject. So the associated $Z$ variables must be assigned a value inconsistent with $\sigma, A$, which will be detected with probability $1/3$. Thus at least $\frac{\Omega(\delta)}{3} = \Omega(\delta)$ fraction of the constraints reject. ■

[5] This can be achieved by placing multiple copies of the constraints with appropriate multiplicity for each type.

# 7 Short PCPs and Locally Testable Codes

## 7.1 Short PCPs

**Theorem 7.1** $SAT \in PCP_{\frac{1}{2},1}[\log_2(n \cdot \mathrm{poly}\log n), O(1)]$.

We prove this theorem by relying on a recent result of Ben-Sasson and Sudan,

**Theorem 7.2 ([BS05, Theorem 1])** $SAT \in PCP_{\frac{1}{2},1}[\log_2(n \cdot \mathrm{poly}\log n), \mathrm{poly}\log n]$.

This result can be manipulated into the following form,

**Lemma 7.3** *There exist constants $c_1, c_2 > 0$ and a polynomial-time reduction that transforms any SAT instance $\varphi$ of size $n$ into a constraint graph $G = \langle (V,E), \Sigma, \mathcal{C} \rangle$ such that*

- *$size(G) \le n(\log n)^{c_1}$ and $|\Sigma| = O(1)$.*
- *If $\varphi$ is satisfiable, then $\mathrm{SAT}(G) = 1$.*
- *If $\varphi$ is not satisfiable, then $\mathrm{SAT}(G) \le 1 - \frac{1}{(\log n)^{c_2}}$.*

Before proving the lemma, let us see how it implies the theorem,

**Proof:**( of Theorem 7.1 ) We apply the main theorem (Theorem 4.1) iteratively $k = c_2 \cdot \log\log n$ times. This results in a constraint-graph $G'$ for which $\overline{\mathrm{SAT}}(G') \ge \min(2^k \cdot \overline{\mathrm{SAT}}(G)\,,\, \alpha) = \alpha$, and such that $size(G') = C^{c_2 \log\log n} \cdot n \cdot (\log n)^{c_1} = n \cdot (\log n)^{c_1 + c_2 \log C} = n \cdot \mathrm{poly}\log n$.

To get an error-probability of $\frac{1}{2}$ one can apply standard techniques for 'clever' amplification through expander neighborhoods. ■

**Proof:**( of Lemma 7.3 ) Basically, the idea is to replace each constraint in the constraint system from Theorem 7.2 with a new constraint system, using composition.

Theorem 7.2 yields some constants $a_1, a_2 > 0$ and a reduction from SAT to a system of at most $m = n \cdot (\log n)^{a_1}$ constraints, each over at most $(\log n)^{a_2}$ variables such that satisfiable inputs go to satisfiable systems, and unsatisfiable inputs result in systems for which any assignment satisfies at most $\frac{1}{2}$ of the constraints.

**Two-variable Constraints** Let us introduce one new variable per constraint, over alphabet $\Sigma = \{0,1\}^{(\log n)^{a_2}}$. The number of new variables is $m = n \cdot (\log n)^{a_1}$. Introduce $(\log n)^{a_2}$ new tests per new variable: Each test will query the new variable and exactly one of the original variables queried by the corresponding test. The test will check that the value for the new variable satisfies the original test, and that it is consistent with the second variable. Call this system $\Psi$ and observe that $|\Psi| = n \cdot (\log n)^{a_1 + a_2}$.

What is $\overline{\mathrm{SAT}}(\Psi)$? Given an assignment for the original variables it must cause at least $m/2$ (original) tests to reject. Each new variable that corresponds to a rejecting test must participate in at least one new rejecting constraint. Indeed, even if it is assigned a value that differs from this assignment, it will be inconsistent with at least one original variable. Altogether, at least $\frac{m/2}{m \cdot (\log n)^{a_2}} \ge (\log n)^{-(a_2+1)}$ fraction of the constraints in $\Psi$ must reject.

**Composition** We next apply composition to reduce the alphabet size from $\log|\Sigma| = \mathrm{poly}\log n$ to $O(1)$. This is exactly as done in Lemma 3.5 except that we use a different assignment tester algorithm $\mathcal{P}$ (or equivalently: a PCP of Proximity). Here are the parameters we need of $\mathcal{P}$: the error probability $\varepsilon$ and the output alphabet size are arbitrary constants; the number of queries is $q = 2$; and the proximity parameter is any value $0 < \delta < \frac{1}{4}$. Most importantly we only require that the size of the output is *polynomial* (and not quasi-linear) in the input size. Existence of such an algorithm $\mathcal{P}$ is an implicit consequence of the proof of the PCP Theorem of [AS98, ALM⁺98], and was explicitly described in [BGH⁺04, DR04].

Here is a brief summary of the construction of Lemma 3.5: We encode each variable via a linear-rate, linear-distance error-correcting-code, treating the 'small' variable in each constraint as if its value lies in the large alphabet. We then run $\mathcal{P}$ on each constraint and let the new system $\Psi'$ be the union of the output constraint systems.

The soundness analysis shows that $\overline{\mathrm{SAT}}(\Psi') \geq \overline{\mathrm{SAT}}(\Psi) \cdot (1-\varepsilon) = \Omega((\log n)^{-(a_2+1)}) = \frac{1}{\mathrm{poly}\log n}$ where the middle equality holds since $\varepsilon$ is a constant. Since the input size for $\mathcal{P}$ was the size of one constraint in $\Psi$, i.e., $\mathrm{poly}\log n$, it follows that the size of the constraint system output by $\mathcal{P}$ is also $\mathrm{poly}\log n$. This means that $|\Psi'| = |\Psi| \cdot \mathrm{poly}\log n = n \cdot \mathrm{poly}\log n$ ■

### 7.2 Short Locally Testable Codes

A similar construction to Theorem 7.1 can be used to obtain quasi-linear locally-testable codes (with block-length $n \cdot \mathrm{poly}\log n$) that are testable with a constant number of queries. This calls for more attention to the difference between a PCP reduction (which is what we have constructed in Theorem 4.1) and a stronger *assignment-tester* reduction. In the terminology of [BGH⁺04, BS05] this is the difference between PCPs and *PCPs of Proximity*. The latter object, as it turns out, can be easily made into a locally-testable code with similar block-length, see [BGH⁺04, Construction 4.3].

hIn order to get locally testable codes, we enhance every step of our construction with 'proximity'. The main observation is that our proofs always follows through a "local-decoding" argument: an assignment for the new constraint graph is translated into an assignment for the original constraint graph in a local way, and so the 'proximity' is easy to maintain. We postpone details to the full version.

## Acknowledgements

I am thankful to Omer Reingold and Luca Trevisan for many discussions about combinatorial analyses of graph powering, which were the direct trigger for the amplification lemma. I would also like to thank David Arnon, Miki Ben-Or, Ehud Friedgut, and Alex Samorodnitsky for helpful comments.

## References

[ALM⁺98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. *J. ACM*, 45(3):501–555, 1998. 1, 18

[AS98] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998. 1, 18

[BGH⁺04] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. In *Proc. 36th ACM Symp. on Theory of Computing*, 2004. 2, 3, 8, 18

[BGS98] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal on Computing*, 27(3):804–915, June 1998. 21

[BS05] Eli Ben-Sasson and Madhu Sudan. Robust PCPs of proximity, shorter PCPs and applications to coding. In *Proc. 37th ACM Symp. on Theory of Computing*, 2005. 3, 17, 18

[BSVW03] Eli Ben-Sasson, Madhu Sudan, Salil P. Vadhan, and Avi Wigderson. Randomness-efficient low degree tests and short PCPs via epsilon-biased sets. In *Proc. 35th ACM Symp. on Theory of Computing*, pages 612–621, 2003. 3

[DR04] Irit Dinur and Omer Reingold. Assignment testers: Towards combinatorial proofs of the PCP theorem. In *Proceedings of the 45th Symposium on Foundations of Computer Science (FOCS)*, 2004. 2, 3, 8, 9, 18

[FGL$^+$96] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. *Journal of the ACM*, 43(2):268–292, 1996. 1

[FKN02] E. Friedgut, G. Kalai, and A. Naor. Boolean functions whose fourier transform is concentrated on the first two levels. *Adv. in Applied Math.*, 29:427–437, 2002. 22

[GS97] Goldreich and Safra. A combinatorial consistency lemma with application to proving the PCP theorem. In *RANDOM: International Workshop on Randomization and Approximation Techniques in Computer Science*. LNCS, 1997. 3

[GS02] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. In *Proc. 43rd IEEE Symp. on Foundations of Computer Science*, pages 13–22, 2002. 3

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of ACM*, 48:798–859, 2001. 20, 22

[HS01] Prahladh Harsha and Madhu Sudan. Small PCPs with low query complexity. In *STACS*, pages 327–338, 2001. 3

[LW03] N. Linial and A. Wigderson. Expander graphs and their applications. Lecture notes of a course: http://www.math.ias.edu/ boaz/ExpanderCourse/, 2003. 4

[PS94] A. Polishchuk and D. Spielman. Nearly linear size holographic proofs. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 194–203, 1994. 3

[PY91] C. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991. 6

[Rei05] Omer Reingold. Undirected st-connectivity in log-space. In *Proc. 37th ACM Symp. on Theory of Computing*, 2005. 2

[RVW] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. *Annals of Mathematics*. 2, 4

## A A Lemma about similar binomial distributions

For $n \in \mathbb{N}$ and $p \in [0,1]$ let $B_{n,p}$ denote a binomially distributed random variable, i.e., $\Pr[B_{n,p} = k] = \binom{n}{k} p^k (1-p)^{n-k}$. The following lemma asserts that if $n, m$ are close, then the distributions of $B_{n,p}$ and $B_{m,p}$ are close.

**Lemma** 5.2 *For every* $p \in [0,1]$ *and* $c > 0$ *there exists some* $0 < \tau < 1$ *such that if* $n - \sqrt{n} \le m < n$, *then*

$$\forall k \in \mathbb{N}, |k - pn| \le c\sqrt{n}, \qquad \tau \le \frac{\Pr[B_{n,p} = k]}{\Pr[B_{m,p} = k]} \le \frac{1}{\tau}\,.$$

**Proof:** Write $n = m + r$ for some $0 \le r \le \sqrt{n}$. We will use the identity $\binom{m+1}{k} = \frac{m+1}{m+1-k}\binom{m}{k}$,

$$\begin{aligned}\Pr[B_{n,p} = k] &= \binom{m+r}{k} p^k (1-p)^{m+r-k} \\ &= \frac{m+1}{m+1-k} \cdot \frac{m+2}{m+2-k} \cdots \frac{m+r}{m+r-k}\binom{m}{k} \cdot p^k (1-p)^{m-k} (1-p)^r \\ &= X \cdot p^k (1-p)^{m-k} \binom{m}{k} = X \cdot \Pr[B_{m,p} = k]\end{aligned}$$

where $X = (1-p)^r \frac{m+1}{m+1-k} \cdot \frac{m+2}{m+2-k} \cdots \frac{m+r}{m+r-k}$ is bounded as follows. For all $a \le r \le \sqrt{n}$,

$$\frac{m+a}{m+a-k} \ge \frac{m}{(1-p)m + (c+1)\sqrt{n}} \ge \frac{1}{1-p}\left(1 - \frac{c+1}{(1-p)\sqrt{n}}\right)$$

where the first inequality holds since $m - k \le m - pn + c\sqrt{n} \le (1-p)m + c\sqrt{n}$. Also,

$$\frac{m+a}{m+a-k} \le \frac{m+\sqrt{n}}{(1-p)m - c\sqrt{n}} \le \frac{1}{1-p}\left(1 + \frac{4c}{(1-p)\sqrt{n}}\right)$$

The product of $r$ such terms cancels the $(1-p)^r$ and leaves a factor at least $\tau = e^{-\frac{4c+1}{1-p}}$, and at most $1/\tau$. ■

## B The Long Code Test

We prove Theorem 6.1. This is basically reworking a test of Håstad [Hås01], into our easier setting:

**Standard Definitions.** We identify $L = \{f : [n] \to \{-1,1\}\}$ with the Boolean hypercube $\{1,-1\}^n$, and use letters $f, g$ for points in the hypercube. We use letters $A, B$ or $\chi$ to denote functions whose domain is the hypercube[6]. For $\alpha \subset [n]$, define

$$\chi_\alpha : \{-1,1\}^n \to \{-1,1\}, \qquad \chi_\alpha(f) \triangleq \prod_{i \in \alpha} f(i)\,.$$

The characters $\{\chi_\alpha\}_{\alpha \subseteq [n]}$ form an orthonormal basis for the space of functions $\{A : \{-1,1\}^n \to \mathbb{R}\}$, where inner product is defined by $\langle A, B\rangle = \mathbb{E}_f\,[A(f)B(f)] = 2^{-n}\sum_f A(f)B(f)$. It follows that any function $A : \{-1,1\}^n \to \{-1,1\}$ can be written as $A = \sum_\alpha \hat{A}_\alpha \chi_\alpha$, where $\hat{A}_\alpha = \langle A, \chi_\alpha\rangle$. We also have Parseval's identity, $\sum_\alpha |\hat{A}_\alpha|^2 = \langle A, A\rangle = 1$.

[6] We consider here functions whose domain is an arbitrary set of size $n$, and wlog we take the set $[n]$. In the application this set is usually some $\{0,1\}^s$ but we can safely ignore this structure, and forget that $n = 2^s$.

**The Test.** Let $\psi : [n] \to \{-1,1\}$ be some predicate, and fix some $\tau > 0$. Let $A : \{-1,1\}^n \to \{-1,1\}$. A function $A : \{1,-1\}^n \to \{1,-1\}$ is the legal encoding of the value $a \in [n]$ iff $A(f) = f(a)$ for all $f \in L$. The following procedure tests whether $A$ is close to a legal encoding of some value $a \in [n]$ that satisfies $\psi$.

1. Select $f, g \in L$ at random
2. Fix $\tau = 0.01$. Set $h = g\mu$ where $\mu \in L$ is selected by doing the following independently for every $y \in [n]$. If $f(y) = 1$ set $\mu(y) = -1$. If $f(y) = -1$ set
$$\mu(y) = \begin{cases} 1 & \text{w. prob. } 1-\tau \\ \\ -1 & \text{w. prob. } \tau \end{cases}.$$
3. Accept unless $A(g) = A(f) = A(h) = 1$.

**Folding.** As usual, we fold $A$ over true and over $\psi$, as done in [BGS98]. This means that whenever the test needs to read $A[f]$, it reads $A[f \wedge \psi]$ instead. In addition, we fold over true which means for every pair $f, -f$ we let $A$ specify only one, and access the other through the identity $A[f] = -A[-f]$. In other words, we assume wlog that $A(f) = A(f \wedge \psi)$ and $A(f) = -A(-f)$ for all $f$.

It is well-known that $\hat{A}_\alpha = 0$ whenever (i) $|\alpha|$ is even, or (ii) $\exists i \in \alpha$ for which $\psi(i) = 1$ (recall that 1 corresponds to false). The reason is that we can partition $\{1,-1\}^n$ into pairs $f, f'$ such that

$$\hat{A}_\alpha = 2^{-n} \sum_f A(f)\chi_\alpha(f) = 2^{-n} \cdot \frac{1}{2} \sum_f (A(f)\chi_\alpha(f) + A(f')\chi_\alpha(f')) = 2^{-n-1} \sum_f 0 = 0\,.$$

In (i) let $f' = -f$, so $\chi_\alpha(f) = \chi_\alpha(f')$ but $A(f) = -A(f')$. In (ii) let $f' = f + e_i$ where $i$ is an index for which $\psi(i) = 1$; so $\chi_\alpha(f) = -\chi_\alpha(f')$ but $A(f) = A(f')$.

**Correctness.** It is easy to check completeness: We fix some $a \in [n]$ and assign for all $f$, $A(f) = f(a)$. Clearly if $A(f) = f(a) = -1$ then the test accepts. Also, if $A(f) = f(a) = 1$ then $A(h) = h(a) = -g(a) = -A(g) \neq A(g)$, and again the test accepts.

For soundness, arithmetize the acceptance probability as follows

$$\Pr[\text{Test accepts}] = \mathbb{E}_{f,g,h}\left[1 - \frac{(1+A(f))(1+A(g))(1+A(h))}{8}\right] =$$

and note that since the pairs $(f,g)$ and $(f,h)$ are pairs of random independent functions, and since $A$ is folded, this equals,

$$= \frac{7}{8} - \frac{1}{8}\mathbb{E}_{g,h}\left[A(g)A(h)\right] - \frac{1}{8}\mathbb{E}_{f,g,h}\left[A(f)A(g)A(h)\right].$$

The first expectation can be expanded as

$$\mathbb{E}_{g,h}\left[\sum_{\alpha,\beta\subseteq[n]} \hat{A}_\alpha \hat{A}_\beta \chi_\alpha(g)\chi_\beta(h)\right] = \sum_{\alpha\subseteq[n]} \hat{A}_\alpha^2 (-\tau)^{|\alpha|}$$

which is bounded by $\tau$ in absolute value, since $\hat{A}_\phi = 0$. For the acceptance probability to be above $1-\varepsilon$, the second expectation (whose value let us name $W$) must be $\leq -1+\tau+8\varepsilon$. We write it as

$$\begin{aligned}
-1+\tau+8\varepsilon \geq W &= \mathbb{E}_{g,f,\mu}\left[\sum_{\alpha,\beta,\gamma\subseteq[n]} \hat{A}_\alpha \hat{A}_\beta \hat{A}_\gamma \chi_\alpha(g)\chi_\beta(g\mu)\chi_\gamma(f)\right] = \\
&= \sum_{\alpha,\gamma\subseteq[n]} \hat{A}_\gamma \hat{A}_\alpha^2 \mathbb{E}_{f,\mu}\left[\chi_\alpha(\mu)\chi_\gamma(f)\right] \\
&= \sum_{\gamma\subseteq\alpha\subseteq[n]} \hat{A}_\gamma \hat{A}_\alpha^2 (-1+\tau)^{|\gamma|}(-\tau)^{|\alpha\setminus\gamma|}\,.
\end{aligned}$$

We now bound the absolute value of this sum, following [Hås01]. First we claim that

$$\sum_{\gamma\subseteq\alpha}((1-\tau)^{|\gamma|}(\tau)^{|\alpha\setminus\gamma|})^2 \leq (1-\tau)^{|\alpha|}\,.$$

The left hand side is the probability that tossing $2|\alpha|$ independent $\tau$-biased coins results in a pattern $\gamma\gamma$ where $\gamma \in \{0,1\}^{|\alpha|}$. This probability is $(\tau^2+(1-\tau)^2)^{|\alpha|} \leq (1-\tau)^{|\alpha|}$ since $\tau < 1-\tau$. By Cauchy-Schwartz,

$$\sum_{\gamma\subseteq\alpha} |\hat{A}_\gamma|(1-\tau)^{|\gamma|}(\tau)^{|\alpha\setminus\gamma|} \leq \sqrt{\sum_{\gamma\subseteq\alpha}|\hat{A}_\gamma|^2}\cdot\sqrt{\sum_{\gamma\subseteq\alpha}((1-\tau)^{|\gamma|}(\tau)^{|\alpha\setminus\gamma|})^2} \leq (1-\tau)^{|\alpha|/2}$$

so, splitting the sum into $|\alpha| = 1$ and $|\alpha| > 1$,

$$|W| \leq \sum_{|\alpha|=1} |\hat{A}_\alpha^2|(1-\tau) + \sum_{|\alpha|>1} |\hat{A}_\alpha|^2(1-\tau)^{|\alpha|/2}\,.$$

Denoting by $\rho = \sum_{|\alpha|>1}|\hat{A}_\alpha|^2$, we have $|W| \leq (1-\rho)(1-\tau)+\rho(1-\tau)^{3/2}$, since $\hat{A}_\alpha = 0$ for $|\alpha|$ even. Thus

$$1-\tau-8\varepsilon \leq |W| \leq (1-\tau)((1-\rho)+\rho\sqrt{1-\tau}) \quad\Rightarrow\quad \rho \leq \frac{8\varepsilon}{(1-\tau)(1-\sqrt{1-\tau})}\,.$$

Since $\tau$ is fixed, we can choose $\varepsilon = \Theta(\delta)$ small enough so that this entire expression is $\Theta(\delta)$.

At this point we use the following result,

**Theorem B.1 ([FKN02])** *Let $\rho > 0$ and let $A : \{1,-1\}^n \to \{1,-1\}$ be a Boolean function for which $\sum_{\alpha,|\alpha|>1}|\hat{A}_\alpha|^2 < \rho$. Then either $|\hat{A}_\phi|^2 = 1-O(\rho)$ or $|\hat{A}_{\{i\}}|^2 = 1-O(\rho)$ for some $i \in [n]$.*

Thus, by folding, there must be some $i \in [n]$ for which $\psi(i) = -1$ and $\mathrm{dist}(A,\chi_{\{i\}}) \leq O(\rho)$. (Note that Theorem B.1 allows also for $\mathrm{dist}(A,-\chi_{\{i\}}) = O(\rho)$ but this would cause the test to have failed with probability $\approx 1/4$, contradicting our assumption.)

We have proven that unless the table $A$ is $\delta$-close to some $\chi_{\{i\}}$ for a value of $i$ that satisfies $\psi$, at least $\varepsilon = \Omega(\delta)$ of the tests must reject. ■