Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 46 (2005)

# The PCP Theorem by Gap Amplification

Irit Dinur*

September 25, 2005

**Abstract**

We describe a new proof of the PCP theorem that is based on a combinatorial amplification lemma. The *unsat value* of a set of constraints $\mathcal{C} = \{c_1, \ldots, c_n\}$, denoted $\text{UNSAT}(\mathcal{C})$, is the smallest fraction of unsatisfied constraints, ranging over all possible assignments for the underlying variables.

We prove a new combinatorial amplification lemma that doubles the unsat-value of a constraint-system, with only a linear blowup in the size of the system. Iterative application of this lemma yields a 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 unsat-value 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 *unsat-value* of $\mathcal{C}$, denoted $\text{UNSAT}(\mathcal{C})$, is the smallest fraction of unsatisfied constraints, over all possible assignments for $V$. Clearly $\mathcal{C}$ is satisfiable if and only if $\text{UNSAT}(\mathcal{C}) = 0$. Also, if $\mathcal{C}$ is not satisfiable then $\text{UNSAT}(\mathcal{C}) \geq 1/n$.

**Background** The PCP Theorem is equivalent to stating that gap-3SAT is NP-hard, namely: for some $\alpha > 0$, given a set $\mathcal{C}$ of constraints such that each is a conjunction of three literals, it is NP-hard to distinguish between $\text{UNSAT}(\mathcal{C}) = 0$ and $\text{UNSAT}(\mathcal{C}) > \alpha$. 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$^+$96, ALM$^+$98] connection between this formulation and the gap-3SAT formulation stated above came as a surprise, and together with the proof of the PCP Theorem by [AS98, ALM$^+$98], 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.



ISSN 1433-8092

**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. Consider the NP-hard problem of deciding if a given graph is 3-colorable or not. This is a system of inequality constraints, where each constraint is over two variables, and the variables take values in the set $\{1, 2, 3\}$. Given such a system $\mathcal{C}$, it is NP-hard to distinguish between the cases (i) $\text{UNSAT}(\mathcal{C}) = 0$ and (ii) $\text{UNSAT}(\mathcal{C}) \geq 1/n$, where $n$ is the number of constraints. We repeatedly apply the amplification lemma to $\mathcal{C}$, doubling the unsat value at each iteration. The outcome $\mathcal{C}'$ is a constraint system for which in the first case still $\text{UNSAT}(\mathcal{C}') = 0$, and in the second case $\text{UNSAT}(\mathcal{C}') \geq \alpha$ for some constant $\alpha > 0$. This proves that gap constraint satisfaction is NP-hard, which is (or is equivalent to) the PCP Theorem.

What makes the unsat value double? Any 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 unsat value 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 unsat value 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 unsat value 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 unsat value, 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 $\log |G|$ times:

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

**Related Work** 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

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

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].

This construction is inspired by the zig-zag construction of expander graphs due to [RVW02] and by Reingold's remarkable proof for $SL = L$ [Rei05]. Although there is no direct technical connection between these works and our construction, our proof has the same overall structure, consisting of a logarithmic number of iterations, where each iteration makes a small improvement in the interesting parameter (be it the unsat value in our case, or the spectral gap in Reingold's case).

The steady increase of the unsat value is inherently different from the original proof of the PCP Theorem. There, a constant unsat value (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. Composition is essential in both proofs.

**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, $n \cdot \mathrm{poly}\log n$, but the number of queries it requires is $\mathrm{poly}\log n$. 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 $n \cdot \mathrm{poly}\log n$, such that if the input was a 'no' instance, then $\textsc{unsat}(\mathcal{C}) \geq \frac{1}{\mathrm{poly}\log n}$. Then, by applying our amplification lemma $O(\log\log n)$ times, we raise the unsat value to a constant, while increasing the size of the system by only another polylogarithmic factor. Namely, we show that $SAT \in PCP_{\frac{1}{2},1}[\log_2(n \cdot \mathrm{poly}\log n),\ O(1)]$.

We further extend our main amplification step to work for assignment-tester reductions (alternatively called PCPs of Proximity). This carries over to extend our constructions of PCPs to constructions of assignment-testers / PCPs of Proximity. By obtaining "short" assignment-testers (with comparable parameters to those of the short PCPs described above) one immediately gets short locally-testable codes as well.

**Organization** Section 2 contains some preliminaries, including a formal definition of constraint graphs, and some basic facts about expander graphs and probability. 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 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 construct PCPs and locally-testable codes whose length is linear up to a poly-logarithmic factor. In Section 8 we describe how to extend our main amplification step for assignment-testers. We include a short discussion about our amplification and parallel-repetition in Section 9.

# 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', defi ned 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 $c(e) \subseteq \Sigma^2$, and $\mathcal{C} = \{c(e)\}_{e \in E}$. A constraint $c(e)$ is said to be satisfied by $(a, b)$ iff $(a, b) \in c(e)$.*

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

$$\text{UNSAT}_\sigma(G) = \Pr_{(u,v) \in E} [(\sigma(u), \sigma(v)) \notin c(e)] \quad \text{and} \quad \text{UNSAT}(G) = \min_\sigma \text{UNSAT}_\sigma(G) .$$

We call $\text{UNSAT}(G)$ the **unsat-value** of $G$, or just the unsat 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 $|\Sigma| = 3$, it is NP-hard to decide if $\text{UNSAT}(G) = 0$.*

**Proof:** We reduce from graph 3-colorability. Given a graph $G$, let the alphabet be $\Sigma = \{1, 2, 3\}$ for the three colors, and equip the edges with inequality constraints. Clearly, $G$ is 3-colorable if and only if $\text{UNSAT}(G) = 0$. ∎

We sometimes use the same letter $G$ to denote the constraint graph and the underlying graph.

## 2.2 Expander Graphs

Expander graphs play an important role in many results in theoretical computer science. In this section we will state some well-known properties of expander graphs. For an excellent exposition to this subject, we refer the reader to [LW03].

**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** *of $G$ 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 $h(X_n) \geq h_0$.* ∎

**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 [RVW02]. 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 adjacency matrix of a graph $G = (V, E)$ is a $|V| \times |V|$ matrix $A$ such that $A_{ij} = 1$ iff $(i, j) \in E$ and $A_{ij} = 0$ otherwise. The second eigenvalue of a graph $G$ is the second largest eigenvalue of its adjacency matrix. The following important relation between the edge expansion and the second eigenvalue is well-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*

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

■

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 $B \subseteq V$ be the support of $K$. Let $A$ be the normalized $n \times n$ adjacency matrix of $G$, i.e., $A_{ij}$ equals $k/d$ where $k$ 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, $\Pr_K[v] \leq \frac{d}{2|F|}$. 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 inside $F$,

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

Let $\mathbf{1}$ be the all 1 vector. Write $x = x^{\perp} + x^{||}$ where $x^{||} \triangleq \frac{1}{n}\mathbf{1}$, is an eigenvector of $A$ with eigenvalue 1, and $x^{\perp} \triangleq x - x^{||}$. 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 $\|x\| = \sqrt{\sum_v x_v^2}$. Clearly,

$$\|A^i x^{\perp}\| \leq |\tilde{\lambda}|^i \|x^{\perp}\| \leq |\tilde{\lambda}|^i \|x\| .$$

Observe that $\|x\|^2 \leq (\sum_v |x_v|) \cdot (\max_v |x_v|) \leq 1 \cdot (\max_v |x_v|) \leq \frac{d}{2|F|}$. By Cauchy-Schwartz

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

Combining the above we get the claim,

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

■

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

### 2.3 Probability

The following easy fact is a Chebychev-style inequality. It is useful for showing that for a non-negative random variable $X$, $Pr[X > 0] \approx \mathbb{E}[X]$ whenever $\mathbb{E}[X] \approx \mathbb{E}[X^2]$.

**Fact 2.5** *For any non-negative random variable $X$, $\Pr[X > 0] \geq \frac{\mathbb{E}^2[X]}{\mathbb{E}[X^2]}$.*

**Proof:** Since $X$ is non-negative, both $\mathbb{E}[X^2] = \mathbb{E}[X^2|X > 0] \cdot (Pr[X > 0])$ and $\mathbb{E}[X|X > 0] = \mathbb{E}[X] \cdot Pr[X > 0]$. Thus

$$\frac{\mathbb{E}^2[X]}{\mathbb{E}[X^2]} = \frac{(\mathbb{E}[X|X > 0] \cdot \Pr[X > 0])^2}{\mathbb{E}[X^2|X > 0] \cdot \Pr[X > 0]} \leq \Pr[X > 0]$$

where the inequality follows because $\mathbb{E}[X^2|X > 0] \geq \mathbb{E}^2[X|X > 0]$ (to see this, observe that for any random variable $X'$, $Var[X'] = \mathbb{E}[X'^2] - \mathbb{E}^2[X'] \geq 0$, and we plug in $X'$ to be the random variable $[X|X > 0]$). ■

### 2.4 Error Correcting Codes

An *error-correcting code* is a collection of strings $C \subseteq \Sigma^n$, where $\Sigma$ is some finite alphabet. $n$ is called the block-length of the code, and $\log_{|\Sigma|} |C|$ is the rate of the code. The distance of the code is $\min_{x \neq y \in C} \operatorname{dist}(x, y)$ where $\operatorname{dist}(\cdot, \cdot)$ refers to Hamming distance.

A one-to-one mapping $\mathrm{e} : D \to \Sigma^n$ is also sometimes called an error-correcting code. Its rate and distance are defined to be the respective rate and distance of its image $\mathrm{e}(D)$.

It is well-known that there exist families of codes $\{C_n \subset \{0,1\}^n\}_{n \in \mathbb{N}}$ for which both the distance and the rate are $\Theta(n)$, and for which there is a polynomial-sized circuit that checks $x \stackrel{?}{\in} C_n$, see e.g. [SS96].

## 3 Operations on Constraint Graphs

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

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

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

### 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 $size(G') = O(size(G))$.*
- $\beta_1 \cdot \text{UNSAT}(G) \leq \text{UNSAT}(G') \leq \text{UNSAT}(G)$.

Note that the third item implies that completeness is maintained, i.e., if $\text{UNSAT}(G) = 0$ then $\text{UNSAT}(G') = 0$. 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*

$$c \cdot \text{UNSAT}(G) \leq \text{UNSAT}(G') \leq \text{UNSAT}(G)$$

*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 $\text{UNSAT}(G)$ 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 $d = d_0 + 1$.

The graph $G'$ will have, for each vertex $v$ of $G$, a copy of $X_{d_v}$ where $d_v$ is the degree of $v$ in $G$. Denote the vertices of this copy of $X_{d_v}$ by $[v]$ and so the vertices of $G'$ are

$$[V] \stackrel{\triangle}{=} \cup_v [v]\,.$$

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 $\text{UNSAT}(G')$. The (completeness) upper bound $\text{UNSAT}(G') \leq \text{UNSAT}(G)$ is easy: An assignment $\sigma : V \to \Sigma$ can be extended to an assignment $\sigma' : [V] \to \Sigma$ by

$$\forall v \in V \; x \in [v], \quad \sigma'(x) \stackrel{\triangle}{=} \sigma(v).$$

The assignment $\sigma'$ causes the same number of edges to reject as does $\sigma$, 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, $\text{UNSAT}_{\sigma'}(G') = \text{UNSAT}(G')$. Define $\sigma : V \to \Sigma$ according to plurality of $\sigma'$, i.e., let $\sigma(v)$ be the most popular value among $(\sigma'(x))_{x \in [v]}$:

$$\forall v \in V, \quad \sigma(v) \stackrel{\triangle}{=} \max \arg_{a \in \Sigma} \left\{ \Pr_{x \in [v]} \left[ \sigma'(x) = a \right] \right\}. \tag{1}$$

Let $F \subseteq E$ be the edges of $G$ that reject $\sigma$, and let $\mathbf{F} \subseteq \mathbf{E}$ be the edges of $G'$ that reject $\sigma'$. Let $S \subseteq [V]$ be the set of vertices of $G'$ whose value disagrees with the plurality,

$$S = \bigcup_{v \in V} \left\{ x \in [v] \,|\, \sigma'(x) \neq \sigma(v) \right\}.$$

The external edge corresponding to an $e \in F$ either rejects $\sigma'$ (i.e. is in $\mathbf{F}$), or has at least one endpoint in $S$. Hence, for $\alpha \stackrel{\triangle}{=} \frac{|F|}{|E|} = \text{UNSAT}_\sigma(G)$,

$$|\mathbf{F}| + |S| \geq |F| = \alpha \cdot |E| \,. \tag{2}$$

There are two cases,

- If $|\mathbf{F}| \geq \frac{\alpha}{2} |E|$ we are done since $\frac{\alpha}{2} |E| = \frac{\alpha}{2d} |\mathbf{E}|$ and so $\text{UNSAT}(G') \geq \text{UNSAT}(G)/2d$ (recall that $d$ is a constant independent of the degree of $G$).
- Otherwise, $|\mathbf{F}| < \frac{\alpha}{2} |E|$, so by (2), $|S| \geq \frac{\alpha}{2} |E|$. 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 $|S_a| \leq |[v]|/2$, so by the edge expansion of the appropriate expander $X_{d_v}$, $E(S_a, \bar{S_a}) \geq h_0 \cdot |S_a|$. All of the edges leaving $S_a$ carry equality constraints that reject $\sigma'$. So there are at least $h_0 \sum_v |S \cap [v]| = h_0 |S| \geq \frac{\alpha h_0}{2} |E|$ edges that reject $\sigma'$. Since $|E| = |\mathbf{E}|/d$, we get $\text{UNSAT}(G') \geq \frac{h_0}{2d} \text{UNSAT}(G)$.

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 $\lambda(G') \leq d + d_0 + 1 - \frac{{h_0}^2}{d+d_0+1} < deg(G')$,*
- *$size(G') = O(size(G))$, and*
- *$\frac{d}{d+d_0+1} \cdot \text{UNSAT}(G) \leq \text{UNSAT}(G') \leq \text{UNSAT}(G)$.*

**Proof:** The idea is to add to $G$ self-loops and edges of an expander and put trivial 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 $h(X) \geq h_0$ (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 trivial 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),

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

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

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 $\beta_1 = c \cdot \frac{d}{d+d_0+1}$. ■

We conclude with a corollary of the above proofs that will be useful in Section 7.

**Corollary 3.4** *Let $\beta_1 > 0$ be the constant from Lemma 3.1. Fix a constraint graph $G$, and let $G' = prep(G)$. Let $V$ be the vertices of $G$ and let $[V] = \cup_{v \in V} [v]$ be the vertices of $G'$. For any assignment $\sigma' : [V] \to \Sigma$, let $\sigma : V \to \Sigma$ be defined according to plurality of $\sigma'$, as in Equation (1). Then, $\text{UNSAT}_{\sigma'}(prep(G)) \geq \text{UNSAT}_\sigma(G) \cdot \beta_1$.*

**Proof:** Let $G_1$ be the graph obtained from $G$ after Lemma 3.2, and let $G'$ be the graph obtained from $G_1$ after Lemma 3.3. Clearly from the proof of Lemma 3.3, for every $\sigma'$, $\text{UNSAT}_{\sigma'}(G') \geq \text{UNSAT}_{\sigma'}(G_1) \cdot c'$. More interestingly, looking into the proof of Lemma 3.2, we see that it actually proves $\text{UNSAT}_{\sigma'}(G_1) \geq \text{UNSAT}_{\sigma}(G) \cdot \frac{d}{d+d_0+1}$ (where $\sigma$ is defined according to plurality of $\sigma'$ as in Equation (1)). Combining the two inequalities,

$$\text{UNSAT}_{\sigma'}(G') \geq \text{UNSAT}_{\sigma'}(G_1) \cdot c' \geq \text{UNSAT}_{\sigma}(G) \cdot \frac{d}{d+d_0+1} \cdot c' = \text{UNSAT}_{\sigma}(G) \cdot \beta_1$$

■

### 3.2 Powering

This operation is a new operation on constraint systems, and it is the one that amplifies the unsat-value. Let $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ be a constraint graph , and let $t \in \mathbb{N}$. We define $G^t = \left\langle (V, \mathbf{E}), \Sigma^{d^{\lceil t/2 \rceil}}, \mathcal{C}^t \right\rangle$ 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 $\text{UNSAT}(G) = 0$ then clearly $\text{UNSAT}(G^t) = 0$. More interestingly, we prove that $\text{UNSAT}(G^t) \geq \Theta(\sqrt{t}) \cdot \text{UNSAT}(G)$, essentially.

**Lemma 3.5 (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 $\lambda(G) \leq \lambda$,*

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

So, as long as $\text{UNSAT}(G) \leq \frac{1}{t}$, this means $\text{UNSAT}(G^t) \geq \Theta(\sqrt{t}) \cdot \text{UNSAT}(G)$. 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 unsat-value. We rely on *composition* which is an essential component in the construction of PCPs. 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) SAT to gap constraint satisfaction. Such a reduction is a polynomial-time

algorithm that inputs a SAT formula on $n$ Boolean variables, and generates a new system of constraints $\mathcal{C}$ with the following gap property: Satisfiable formulae translate to systems $\mathcal{C}$ for which $\text{UNSAT}(\mathcal{C}) = 0$, and unsatisfiable formulae translate to systems $\mathcal{C}$ for which $\text{UNSAT}(\mathcal{C}) > \alpha$, for some $\alpha > 0$.

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

How would this work with constraint graphs? Assume we have a "PCP"-reduction $\mathcal{P}$ as above, and let $G$ be a constraint graph. We can put each constraint of $G$ in SAT form, and then feed it to $\mathcal{P}$. The output would be a constant-size constraint graph with alphabet $\Sigma_0$. The final constraint graph would be the union of the constant-size constraint-graphs output by running $\mathcal{P}$ over each of $G$'s constraints (these constant-size constraint graphs will have vertices in common, so the union will not be a disjoint union). Thus we have achieved our goal of reducing the size of the alphabet from $\Sigma$ to $\Sigma_0$. The only parameter that depends on $|\Sigma|$ is the size $M(|\Sigma|)$ of each constant-size constraint graph output by $\mathcal{P}$. The main point is that as long as $|\Sigma| = O(1)$, $\mathcal{P}$ can be allowed to be as inefficient as needed and still $M(|\Sigma|)$ is independent of $n$. Consequently, this procedure incurs only a linear overhead (with $M(|\Sigma|)$ factoring into the constant).

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. Essentially, using an assignment-tester reduction $\mathcal{P}$ will force the different constant-size constraint graphs to have common vertices, and that will ensure consistency. For an exposition as to why assignment-testers 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. For a Boolean circuit $\Phi$ over $n$ variables, denote by $\text{SAT}(\Phi) \subseteq \{0,1\}^n$ the set of assignments that satisfy $\Phi$.

**Definition 3.1 (Assignment Tester)** *An* Assignment Tester *with alphabet $\Sigma_0$ and rejection probability $\epsilon > 0$ is a polynomial-time transformation $\mathcal{P}$ whose input is a circuit $\Phi$ over Boolean variables $X$, and whose output is a constraint graph $G = \langle (V, E), \Sigma_0, \mathcal{C} \rangle$ such that[4] $V \supset X$, and such that the following hold. Let $V' = V \setminus X$, and let $a : X \to \{0,1\}$ be an assignment.*

- *(Completeness) If $a \in \text{SAT}(\Phi)$, there exists $b : V' \to \Sigma_0$ such that $\text{UNSAT}_{a \cup b}(G) = 0$.*
- *(Soundness) If $a \notin \text{SAT}(\Phi)$ then for all $b : V' \to \Sigma_0$, $\text{UNSAT}_{a \cup b}(G) \geq \epsilon \cdot \text{dist}(a, \text{SAT}(\Phi))$.*

Notice that no restriction is imposed on the running time of $\mathcal{P}$ or on $size(G)$. In particular, we ignored the size of the circuit $\Phi$, which we allow to be even exponential in $|X|$. We describe an explicit construction of such an algorithm in Section 6 (see Lemma 6.2). As mentioned earlier, such a reduction (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,

[3]The implicit assumption here is that such (inefficient) reductions are significantly easier to come by, indeed, see e.g. Section 6.

[4]In a constraint graph, the set $V$ plays a double role of both variables and vertices. By $V \supset X$ it is meant that some of the vertices of $V$ are identified with the $X$ variables.

**Lemma 3.6 (Composition)** *Assume the existence of an assignment tester $\mathcal{P}$, with constant rejection probability $\varepsilon > 0$, and alphabet $\Sigma_0$, $|\Sigma_0| = O(1)$. There exists $\beta_3 > 0$ that depends only on $\mathcal{P}$, such that any constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ can be transformed into a constraint graph $G' = \langle (V', E'), \Sigma_0, \mathcal{C}' \rangle$, denoted $G \circ \mathcal{P}$, such that $size(G') = M(|\Sigma|) \cdot size(G)$, and*

$$\beta_3 \cdot \text{UNSAT}(G) \leq \text{UNSAT}(G') \leq \text{UNSAT}(G)$$

**Proof:** We describe the construction in two steps: robustization and composition.

- (Robustization:) First, in order to run $\mathcal{P}$ on each of the constraints of $G$, the constraints must be cast in the form of an input to $\mathcal{P}$. This basically amounts converting each constraint of $G$, which is defined over two non-Boolean variables, into a function over $O(\log |\Sigma|)$ Boolean variables. A naïve way to do this is by having $\lceil \log |\Sigma| \rceil$ Boolean variables encode the binary representation of each $\Sigma$-variable. However, for subtle issues that are discussed at length in [DR04] (and which will show up in the proof below), it is necessary to encode the values in $\Sigma$ via an error-correcting code.

  So let $\mathrm{e} : \Sigma \to \{0,1\}^\ell$ be any encoding with linear rate and relative distance $\rho > 0$. In other words, $\ell = O(\log |\Sigma|)$, and for every $\sigma_1 \neq \sigma_2 \in \Sigma$, the strings $\mathrm{e}(\sigma_1)$ and $\mathrm{e}(\sigma_2)$ differ on at least $\rho\ell$ bits. 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 $c \in \mathcal{C}$ over variables $v, w$ by a constraint $\tilde{c}$ over $2\ell$ Boolean 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 $G_c = \langle (V_c, E_c), \Sigma_0, \mathcal{C}_c \rangle$ denote the resulting constraint graph, and recall that $[v] \cup [w] \subset V_c$. Assume, wlog, that $E_c$ has the same cardinality for each $c$, and define the new constraint graph $G' = \langle (V', E'), \Sigma_0, \mathcal{C} \rangle$, where

$$V' = \bigcup_{c \in G} V_c, \qquad E' = \bigcup_{c \in G} E_c, \qquad \mathcal{C}' = \bigcup_{c \in G} \mathcal{C}_c \,.$$

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

It remains to be seen that $\beta_3 \cdot \text{UNSAT}(G) \leq \text{UNSAT}(G') \leq \text{UNSAT}(G)$. The proof is straightforward and follows exactly the proof of the composition theorem in [DR04]. Let us sketch the first inequality (that corresponds to the soundness argument). We need to prove that every assignment for $G'$ violates at least $\beta_3 \cdot \text{UNSAT}(G)$ fraction of $G'$'s constraints. So let $\sigma' : V' \to \Sigma_0$ be an assignment for $G'$. We first extract from it an assignment $\sigma : V \to \Sigma$ for $G$ 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 $\text{UNSAT}_\sigma(G) \geq \text{UNSAT}(G)$ 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 of the graph $G_c$ reject $\sigma'$. Since $|E'| = \sum_{c \in \mathcal{C}} |E_c|$, and we assumed that $|E_c|$ is the same for all $c \in \mathcal{C}$, this will prove the required inequality. The 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 $\mathrm{e}$), i.e., $\mathrm{dist}(\sigma'|_{[u]\cup[v]}, \mathrm{SAT}(\tilde{c})) \geq \rho/4$. 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 $\tilde{c}$. By the soundness property of $\mathcal{P}$, at least $\varepsilon \cdot \rho/4 = \Omega(1)$ fraction of $G_c$'s constraints reject. Altogether, $\text{UNSAT}(G') \geq \frac{\varepsilon\rho}{4} \cdot \text{UNSAT}(G) = \beta_3 \cdot \text{UNSAT}(G)$ setting $\beta_3 = \varepsilon\rho/4 > 0$. ∎

# 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 $\text{UNSAT}(G) = 0$ then $\text{UNSAT}(G') = 0$*
- *(Soundness:) $\text{UNSAT}(G') \geq \min(2 \cdot \text{UNSAT}(G), \alpha)$.*

**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 $\beta_1 \cdot \text{UNSAT}(G) \leq \text{UNSAT}(H_1) \leq \text{UNSAT}(G)$.

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

   According to Lemma 3.5, there exists some constant $\beta_2 = \beta(\lambda, d, |\Sigma|) > 0$ for which $\text{UNSAT}(H_2) \geq \beta_2\sqrt{t} \cdot \min(\text{UNSAT}(H_1), \frac{1}{t})$. 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 3.6 to $H_2$ relying on an assignment tester $\mathcal{P}$, as guaranteed in Lemma 6.2.

   This reduces the alphabet to $\Sigma_0$ while still $\beta_3 \cdot \text{UNSAT}(H_2) \leq \text{UNSAT}(G') \leq \text{UNSAT}(H_2)$, 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,

$$\begin{aligned}
\text{UNSAT}(G') &\geq \beta_3 \cdot \text{UNSAT}(H_2) && \text{(step 3, Lemma 3.6)} \\
&\geq \beta_3 \cdot \beta_2\sqrt{t} \cdot \min(\text{UNSAT}(H_1), \frac{1}{t}) && \text{(step 2, Lemma 3.5)} \\
&\geq \beta_3 \cdot \beta_2\sqrt{t} \cdot \min(\beta_1 \text{UNSAT}(G), \frac{1}{t}) && \text{(step 1, Lemma 3.1)} \\
&\geq \min(2 \cdot \text{UNSAT}(G), \alpha)
\end{aligned}$$

■

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. According to Proposition 2.1 it is NP-hard to decide if for a given constraint graph $G$ with $|\Sigma| = 3$, $\text{UNSAT}(G) = 0$ or not. So let $G$ be an instance of constraint-graph satisfiability with $|\Sigma| = 3$. The basic idea is to repeatedly apply the main theorem until the unsat-value becomes a constant fraction.

Let $G_0 = G$ and 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 $E_0$ be the edge-set of $G_0$, and let $k = \log |E_0| = O(\log n)$. Observe that the size of $G_i$ for $i \leq k = O(\log n)$ is at most $C^i \cdot size(G_0) = \text{poly}(n)$.

Completeness is easy: if $\text{UNSAT}(G_0) = 0$ then $\text{UNSAT}(G_i) = 0$ for all $i$. For soundness, assume $\text{UNSAT}(G_0) > 0$. If for some $i^* < k$, $\text{UNSAT}(G_{i^*}) \geq \alpha/2$ then the main theorem implies that for all $i > i^*$ $\text{UNSAT}(G_i) \geq \alpha$. For all other $i$ it follows by induction that

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

If $\text{UNSAT}(G_0) > 0$ then $\text{UNSAT}(G_0) \geq \frac{1}{|E_0|}$, so surely $2^k \text{UNSAT}(G_0) > \alpha$. Thus $\text{UNSAT}(G_k) \geq \alpha$.

This proves that gap constraint satisfaction is NP-hard, for two-variable constraints and alphabet size $|\Sigma_0|$. If one is interested specifically in gap-3SAT, a local gadget reduction takes $G_k$ to 3SAT form (by converting each constraint into a constant number of 3CNF clauses), while maintaining the unsat-value 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 $u = 1/\log(\frac{1}{1-\alpha}) = O(1)$. I.e., create new constraints that are ANDs of all possible $u$-tuples of the old constraints. This, of course, increases the number of queries per constraint to $2u$.

■

# 5 Soundness Amplification Lemma

**Lemma 3.5** *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 $\lambda(G) \leq \lambda$,*

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

In other words, as long as $\text{UNSAT}(G) \leq 1/t$, we have $\text{UNSAT}(G^t) > \Omega(\sqrt{t}) \cdot \text{UNSAT}(G)$.

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

Why does $G^t$ have a larger unsat-value than $G$? An assignment $\vec{\sigma} : V \to \Sigma^{d^{t/2}}$ assigns each vertex a vector of $d^{t/2}$ values from $\Sigma$ that supposedly represent its opinions about all of its neighbors at distance $t/2$. Intuitively, since each vertex gets more information, and since it is compared against vertices that are further away, there is more chance to detect inconsistencies.

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. Given an assignment for $G^t$, $\vec{\sigma} : V \to \Sigma^{d^{t/2}}$ we extract from it a new assignment $\sigma : V \to \Sigma$, by assigning each vertex $v$ the most popular value among the "opinions" (under $\vec{\sigma}$) of $v$'s neighbors. We then relate the fraction of edges falsified by this "popular-opinion" assignment $\sigma$ to the fraction of paths falsified by $\vec{\sigma}$. The probability that a random edge rejects this new assignment is, by definition, at least $\text{UNSAT}(G)$. The idea is that a random path

passes through some rejecting edge with even higher probability. Moreover, we will show that if a path does pass through a rejecting edge, it itself rejects with constant probability.

**Proof:** Let $\vec{\sigma} : V \to \Sigma^{d^{t/2}}$ be a 'best' assignment for $G^t$, $\text{UNSAT}(G^t) = \text{UNSAT}_{\vec{\sigma}}(G^t)$. For each $v$, $\vec{\sigma}(v)$ assigns a vector of $d^{t/2}$ values in $\Sigma$, interpreted as values for every vertex $w$ within distance $t/2$ 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. Let $X_v$ be a random variable that assumes a value $a$ with probability that a $t/2$-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 $\Pr[X_v = a]$:

$$\forall v \in V, \quad \sigma(v) \stackrel{\triangle}{=} \max \arg_{a \in \Sigma} \{\Pr[X_v = a]\} . \tag{3}$$

As mentioned above, the assignment $\sigma$ can be interpreted as being the "popular opinion" about $v$ among $v$'s neighbors.

Let $F$ be a subset of edges[5] that reject $\sigma$, so that $\frac{|F|}{|E|} = \min(\text{UNSAT}_\sigma(G), \frac{1}{t})$. From now on $\vec{\sigma}, \sigma, F$ will be fixed for the rest of the proof.

Denote by $\mathbf{E} = E(G^t)$ the edge set of $G^t$. There is a one-to-one correspondence between edges $\mathbf{e} \in \mathbf{E}$ and paths of length $t$ in $G$. With some abuse of notation we write $\mathbf{e} = (v_0, v_1, \ldots, v_t)$ where $(v_{i-1}, v_i) \in E$ for all $1 \le i \le t$.

**Definition 5.1** *A path* $\mathbf{e} = (v_0, \ldots, v_t) \in \mathbf{E}$ *is* hit *by its* $i$*-th edge if*

1. $(v_{i-1}, v_i) \in F$*, and*
2. *Both* $\vec{\sigma}(v_0)_{v_{i-1}} = \sigma(v_{i-1})$ *and* $\vec{\sigma}(v_t)_{v_i} = \sigma(v_i)$.

Let $I = \left\{ \frac{t}{2} - \sqrt{t} < i \le \frac{t}{2} + \sqrt{t} \right\} \subset \mathbb{N}$ be the set of "middle" indices. For each path $\mathbf{e}$, we define $N(\mathbf{e})$ to be the number of times $\mathbf{e}$ is hit in its middle portion:

$$N(\mathbf{e}) = |\{ i \in I \mid i \text{ hits } \mathbf{e}\}| .$$

$N(\mathbf{e})$ is an integer between 0 and $2\sqrt{t}$. Clearly, $N(\mathbf{e}) > 0$ implies that $\mathbf{e}$ rejects under $\vec{\sigma}$ (because having $\mathbf{e}$ hit by, say, the $i$-th edge, means $\sigma(v_{i-1})$ is inconsistent with $\sigma(v_i)$, and this inconsistency carries over to the constraint on $\vec{\sigma}(v_0)$ and $\vec{\sigma}(v_t)$). Thus,

$$\Pr_{\mathbf{e}}[N(\mathbf{e}) > 0] \le \Pr_{\mathbf{e}}[\mathbf{e} \text{ rejects } \vec{\sigma}] = \text{UNSAT}(G^t) .$$

We will prove

$$\Omega(\sqrt{t}) \cdot \frac{|F|}{|E|} \le \Pr_{\mathbf{e}}[N(\mathbf{e}) > 0] . \tag{4}$$

Since by definition

$$\min(\text{UNSAT}(G), \frac{1}{t}) \le \min(\text{UNSAT}_\sigma(G), \frac{1}{t}) = \frac{|F|}{|E|} ,$$

combining the above three equations we get

$$\Omega(\sqrt{t}) \cdot \min(\text{UNSAT}(G), \frac{1}{t}) \le \Omega(\sqrt{t}) \cdot \frac{|F|}{|E|} \le \Pr_{\mathbf{e}}[N(\mathbf{e}) > 0] \le \text{UNSAT}(G^t) \tag{5}$$

which gives the lemma.

We will prove (4) by estimating the first and second moments of the random variable $N$,

[5] $F$ is simply the set of all edges that reject $\sigma$, as long as this set is not too large.

**Lemma 5.1**

$$\mathbb{E}_{\mathbf{e}}[N(\mathbf{e})] \geq \Omega(\sqrt{t}) \cdot \frac{|F|}{|E|}$$

**Lemma 5.2**

$$\mathbb{E}_{\mathbf{e}}[(N(\mathbf{e}))^2] \leq O(\sqrt{t}) \cdot \frac{|F|}{|E|}$$

Equation (4) follows by relying on Fact 2.5, hence $\Pr[N > 0] \geq \mathbb{E}^2[N]/\mathbb{E}[N^2] = \Omega(\sqrt{t}) \cdot \frac{|F|}{|E|}$. ■

## 5.1 Proof of Lemma 5.1

Define an indicator variable $N_i$ by setting $N_i(\mathbf{e}) = 1$ iff the path $\mathbf{e}$ is hit by its $i$-th edge, as in definition 5.1. Recall $I = \left\{\frac{t}{2} - \sqrt{t} < j \leq \frac{t}{2} + \sqrt{t}\right\}$. Clearly, $N = \sum_{i \in I} N_i$. In order to estimate $\mathbb{E}[N]$ we will estimate $\mathbb{E}[N_i]$, and use linearity of expectation.

Fix $i \in I$. In order to estimate $\mathbb{E}[N_i]$ we choose a random $\mathbf{e} \in \mathbf{E}$ according to the following procedure:

1. Choose a random $e = (u, v) \in E$
2. Choose a random path of length $i - 1$ starting from $u$, denote it by $(u = v_{i-1}, v_{i-2}, \ldots, v_1, v_0)$.
3. Choose a random path of length $t - i$ starting from $v$, denote it by $(v = v_i, v_{i+1}, \ldots, v_t)$.
4. Output the path $\mathbf{e} = (v_0, \ldots, v_t)$

Since $G$ is $d$-regular, the stationary distribution is uniform, so this procedure outputs a uniformly random $\mathbf{e} \in \mathbf{E}$. According to Definition 5.1, $\mathbf{e}$ is hit by its $i$-th edge iff $(u, v) \in F$ and $\vec{\sigma}(v_0)_u = \sigma(u)$ and $\vec{\sigma}(v_t)_v = \sigma(v)$.

Clearly, the probability that step 1 results in an edge $(u, v) \in F$ equals exactly $\frac{|F|}{|E|}$. Observe also that the choice of $v_0$ in step 2 *only depends on* $u$, and the choice of $v_t$ in step 3 *only depends on* $v$. Therefore

$$\Pr[N_i > 0] = \frac{|F|}{|E|} \cdot p_u \cdot p_v \tag{6}$$

where $p_u = \Pr_{v_0}[\vec{\sigma}(v_0)_u = \sigma(u)]$ and $p_v = \Pr_{v_t}[\vec{\sigma}(v_t)_v = \sigma(v)]$. It remains to analyze $p_u$ and $p_v$. Let us focus on $p_u$ as the case of $p_v$ is symmetric.

Define a random variable $X_{u,\ell}$ as follows. $X_{u,\ell}$ takes a value $a \in \Sigma$ with probability that a random $\ell$-step walk from $u$ ends in a vertex $w$ for which $\vec{\sigma}(w)_u = a$. In these terms $p_u = \Pr[X_{u,i-1} = \sigma(u)]$, (and $p_v = \Pr[X_{v,t-i} = \sigma(v)]$). Recall that by definition $\sigma(u)$ equals a value $a \in \Sigma$ that maximizes $\Pr[X_{u,t/2} = a]$. In particular, $\Pr[X_{u,t/2} = \sigma(u)] \geq \frac{1}{|\Sigma|}$. For $i - 1 = t/2$ it follows immediately that $p_u \geq 1/|\Sigma|$.

We will prove that for all $\ell$

$$\text{If} \quad |\ell - t/2| \leq \sqrt{t} \quad \text{then} \quad \Pr[X_{u,\ell} = a] > \frac{\tau}{2} \cdot \Pr[X_{u,t/2} = a] \tag{7}$$

for some $\tau > 0$ to be determined. The intuition for (7) 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 $a$ with probability that a $k$-step random walk *conditioned on walking only on non-marked edges* reaches a vertex $w$ for which $\vec{\sigma}(w)_u = a$. 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 $p = 1 - 1/d$,

$$\Pr[X_{u,\ell} = a] = \sum_{k=0}^{\ell} \Pr[B_{\ell,p} = k] \Pr[X'_{u,k} = a]\,. \tag{8}$$

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 5.3** *For every $p \in [0,1]$ and $c > 0$ there exists some $0 < \tau < 1$ such that if $\ell_1 - \sqrt{\ell_1} \le \ell_2 < \ell_1$, then*

$$\forall k, |k - p\ell_1| \le c\sqrt{\ell_1}, \qquad \tau \le \frac{\Pr[B_{\ell_1,p} = k]}{\Pr[B_{\ell_2,p} = k]} \le \frac{1}{\tau}$$

The proof is a straightforward computation, and can be found in Appendix A. We apply the lemma with $\ell_1 = t/2$ and $\ell_2 = i-1$ and choose $c$ so that $\Pr[B_{\frac{t}{2},p} \notin I] \le \frac{1}{2|\Sigma|}$ for the set $I = \left\{k \;\middle|\; |k - p\ell| \le c\sqrt{t}\right\}$; and let $\tau$ be the appropriate constant from the lemma. Clearly $c$ can be chosen independently of $t$ since $k \notin I$ implies $|k - pt/2| \ge |k - p\ell| - |p\ell - pt/2| > (c-1)\sqrt{t}$. We now have

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

where the last inequality holds since $\Pr[X_{u,t/2} = a] \ge \frac{1}{|\Sigma|}$. So (7) is established, and so $p_u, p_v > \frac{\tau}{2|\Sigma|}$ because both $i-1, t-i$ are at most $\sqrt{t}$ away from $t/2$. Plugging this into Equation (6), we get $\mathbb{E}[N_i] \ge \frac{|F|}{|E|} \cdot \Omega(1)$, and this completes the proof of Lemma 5.1. ∎

## 5.2 Proof of Lemma 5.2

For a path $\mathbf{e}$, let $\mathbf{e}_i$ denote its $i$-th edge. In order to upper bound $\mathbb{E}_{\mathbf{e}}[N^2]$ (all expectations are taken over uniform choice of $\mathbf{e}$) we defi ne a random variable $Z(\mathbf{e}) = |\{\, i \in I \mid \mathbf{e}_i \in F\}|$ that counts how many times $\mathbf{e}$ intersects $F$ in the middle portion (recall $I = \left\{\frac{t}{2} - \sqrt{t} < j \le \frac{t}{2} + \sqrt{t}\right\}$). Clearly, $N(\mathbf{e}) \le Z(\mathbf{e})$ for all $\mathbf{e}$, so $\mathbb{E}[N^2] \le \mathbb{E}[Z^2]$.

Let $Z_i(\mathbf{e})$ be an indicator random variable that is 1 iff $\mathbf{e}_i \in F$. So $Z(\mathbf{e}) = \sum_{i \in I} Z_i(\mathbf{e})$, and by linearity of expectation,

$$\mathbb{E}[Z^2] = \sum_{i,j \in I} \mathbb{E}_{\mathbf{e}}[Z_i(\mathbf{e}) Z_j(\mathbf{e})] = \sum_{i \in I} \mathbb{E}[Z_i] + 2 \sum_{\substack{i<j \\ i,\,j\,\in\,I}} \mathbb{E}[Z_i Z_j] = |I| \frac{|F|}{|E|} + 2 \sum_{\substack{i<j \\ i,\,j\,\in\,I}} \mathbb{E}[Z_i Z_j] \tag{9}$$

As it turns out, $E[Z^2]$ is not much larger than $\frac{|I||F|}{|E|} \approx \sqrt{t}\frac{|F|}{|E|}$. The intuitive reason is that since the graph $G$ is an expander, correlations between the $i$-th and the $j$-th steps of a random walk cannot last long, so $\sum \mathbb{E}[Z_i Z_j]$ is small.

**Proposition 5.4** *Fix $i, j \in I,\ i < j$, and $F \subseteq E$. Then,*

$$\mathbb{E}[Z_i Z_j] \leq \frac{|F|}{|E|}\left(\frac{|F|}{|E|} + \lambda^{j-i}\right).$$

Let us first see that combining the proposition with (9) completes the lemma. Indeed, since $|I| = 2\sqrt{t}$ and since $\frac{|F|}{|E|} \leq \frac{1}{t}$,

$$\sum_{\substack{i<j \\ i,j\in I}} \mathbb{E}[Z_i Z_j] \leq \frac{|F|}{|E|} \sum_{\substack{i<j \\ i,j\in I}} \left(\frac{|F|}{|E|} + \lambda^{j-i}\right) < |I|^2 \left(\frac{|F|}{|E|}\right)^2 + |I|\,\frac{|F|}{|E|} \sum_{i=1}^{2\sqrt{t}} \lambda^i = O(\sqrt{t}) \cdot \frac{|F|}{|E|}$$

where the 'O' notation depends only on $\lambda$. Let us now prove the Proposition.

**Proof:** Observe that $Z_i Z_j \in \{0,1\}$, and $\Pr[Z_j > 0] = \frac{|F|}{|E|}$. Thus,

$$\mathbb{E}[Z_i Z_j] = \Pr[Z_i Z_j > 0] = \Pr[Z_i > 0]\Pr[Z_j > 0 \mid Z_i > 0] = \frac{|F|}{|E|} \cdot \Pr[Z_j > 0 \mid Z_i > 0].$$

Assume first $i = 1$. By Proposition 2.4,

$$\Pr_{\mathbf{e}}[Z_j(\mathbf{e}) > 0 \mid Z_1(\mathbf{e}) > 0] \leq \frac{|F|}{|E|} + \lambda^{j-1}$$

where $\lambda < 1$ is the normalized second eigenvalue of the graph $G$. If $i > 1$, we don't care where the random path $\mathbf{e}$ visited during its first $i-1$ steps, so we can ignore those steps. In other words the last $t - i + 1$ steps of a random walk of length $t$ are a random walk of length $t - i + 1$. This is formalized by writing

$$\Pr_{|\mathbf{e}|=t}[Z_j(\mathbf{e}) > 0 \mid Z_i(\mathbf{e}) > 0] = \Pr_{|\mathbf{e}'|=t-i+1}[Z_{j-i+1}(\mathbf{e}') > 0 \mid Z_1(\mathbf{e}') > 0].$$

Now by applying Proposition 2.4 on paths of length $t-i+1$, the right hand side cannot exceed $\frac{|F|}{|E|}+\lambda^{j-i}$. ■

This completes the proof of the amplification lemma ■

Finally, we state an immediate corollary of this proof which will be useful for Section 7.

**Corollary 5.5** *For every $\vec{\sigma} : V \to \Sigma^{d^{t/2}}$ let $\sigma : V \to \Sigma$ be defined according to "popular opinion" (as in Equation (3) above). Then,*

$$\text{UNSAT}_{\vec{\sigma}}(G^t) \geq \beta_2\sqrt{t} \cdot \min\left(\text{UNSAT}_{\sigma}(G),\ \frac{1}{t}\right).$$

**Proof:** This follows directly from Equation (4) (and from the definition of $F$ and $N(\cdot)$). ■

# 6 An Explicit 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 constraint graph 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

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

Recall that two tables $A, A' : L \to \{0,1\}$ are $\delta$-far from one another if $\Pr_f[A(f) \neq A'(f)] \geq \delta$.

**Theorem 6.1** *There exists a* Long-Code Test $T$ *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* $\delta \in [0,1]$*, if a table* $A : L \to \{0,1\}$ *is at least* $\delta$*-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. Let $X = \{x_1, \ldots, x_s\}$ be a set of $s$ Boolean variables. Also, let there be an auxiliary (Boolean) variable per each $f \in L$. With slight abuse of notation we identify $L$ with this set of variables, and interpret an assignment for these variables as a table $A : L \to \{0,1\}$.

   Define a new test $T'$ as follows. With probability $1/2$ run the Long-Code test $T$ (as specified in Theorem 6.1), and with probability $1/2$ choose a random $x_i \in X$ and a random $f \in L$, and test that $\sigma(x_i) = A(f) \oplus A(f + e_i)$.

2. Introduce a new variable $z_r$ per outcome $r$ of the coin tosses of $T'$. These variables will input values in $\{0,1\}^3$, supposedly specifying the correct values of all three variables queried by $T'$ on coin tosses $r$. The final system of constraints will be the following: there will be a constraint for every possible choice of $z_r \in Z$ and a variable $y$ of the three accessed by $T'$ on coin toss $r$ (so $y \in X \cup L$). This constraint will check that the assignment for $z_r$ would have satisfied $T'$, and that it is consistent with the assignment for $y$.

The constraint graph $G$ will have vertices $X \cup L \cup Z$, constraints (and edges) as specified above, and alphabet $\Sigma_0 = \{0,1\}^3$.

**Lemma 6.2** *The reduction taking* $\psi$ *to* $G$ *is an assignment tester, with* $\Sigma_0 = \{0,1\}^3$ *and constant rejection probability.*

**Proof:** (sketch) Perfect completeness is evident. For soundness, assume that $\sigma : X \to \{0,1\}$ is an assignment such that $\mathrm{dist}(\sigma, \mathrm{SAT}(\psi)) = \delta$, for some $\delta > 0$. Let us first show that for every $A : L \to \{0,1\}$, the tables $\sigma, A$ cause $T'$ to reject with probability at least $\Omega(\delta)$. First assume that $A : L \to \{0,1\}$ is $\delta/2$ far from a legal long-code encoding. Then by Theorem 6.1 $T$ rejects with probability at least $\Omega(\delta)$, so $T'$ rejects with probability at least half of that, which is also $\Omega(\delta)$. Otherwise, $A$ is $\delta/2$-close to the long-code encoding of some $\sigma' : X \to \{0,1\}$ which satisfies $\psi$. By assumption on $\sigma$ and by the triangle inequality, $\Pr_i[\sigma(x_i) \neq \sigma'(x_i)) > \delta/2$. Now recall that with probability $1/2$, $T'$ chooses a random $i$ and a random $f$ and checks that $A(f) \oplus A(f + e_i) = \sigma(x_i)$. Since $A$ is close to the long-code encoding of $\sigma'$, for all $i$:

$$\begin{aligned} \Pr_{f \in L}\left[A(f) \oplus A(f + e_i) = \sigma'(x_i)\right] &\geq \Pr_{f \in L}\left[A(f) = f(\sigma') \quad \textit{and} \quad A(f + e_i) = (f \oplus e_i)(\sigma')\right] \\ &\geq 1 - 2 \cdot \delta/2 = 1 - \delta \end{aligned}$$

The check fails whenever $i, f$ are such that $\sigma'(x_i) \neq \sigma(x_i)$ and yet $A(f) \oplus A(f + e_i) = \sigma'(x_i)$. Altogether this occurs with probability at least $(1 - \delta)\delta/2 \geq \delta/4$, and $T'$ runs this test with probability $1/2$, so it rejects again with probability $\Omega(\delta)$.

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 $T'$ to reject with probability at least $\Omega(\delta)$. So the associated $Z$ variables must be assigned a value inconsistent with $\sigma, A$, and each inconsistency will be detected with probability $\geq 1/3$. Thus at least $\frac{\Omega(\delta)}{3} = \Omega(\delta)$ fraction of the constraints reject. ■

# 7 Short PCPs and Locally Testable Codes

In this section we describe how to construct extremely-short Probabilistically Checkable Proofs and Locally-Testable Codes (LTCs). Our starting point is the construction of Ben-Sasson and Sudan [BS05]. The case of short PCPs follows rather directly from our main theorem (Theorem 4.1) and is described first, in Subsection 7.2. The case of short LTCs is analogous, and is obtained similarly from a variant of the main theorem. This variant is an adaptation of our reduction between constraint graphs into a special kind of reduction called an assignment tester or a PCP of Proximity. We feel that this adaptation may be of independent interest, and it is described fully in Section 8. Assuming this adaptation, we describe our short LTCs in Subsection 7.3. Let us first begin with some definitions and notations.

## 7.1 Definitions and Notation

Given a system of constraints $\Phi$, we denote its *unsat-value* by $\text{UNSAT}(\Phi)$: the minimum over all possible assignments for $\Phi$'s variables, of the fraction of unsatisfied constraints. This is a natural extension of the unsat-value of a constraint graph.

**Definition 7.1 ($PCP_{s,c}[\log \ell, q]$)** *We define the class of languages $PCP_{s,c}[\log_2(\ell(n)), q(n)]$, with parameters $s(n), c(n)$ and $\ell(n)$ and $q(n)$ as follows. A language $L$ is in this class iff there is a reduction taking an instance $x$ to a system of constraints $\Phi(x)$ such that, for $n = |x|$,*

- *$|\Phi(x)| \leq \ell(n)$; and each constraint $\varphi \in \Phi(x)$ accesses at most $q(n)$ variables.*
- *If $x \in L$ then $1 - \text{UNSAT}(\Phi(x)) \geq c(n)$*
- *If $x \notin L$ then $1 - \text{UNSAT}(\Phi(x)) \leq s(n)$*

**Definition 7.2 (Locally Testable Codes)** *A code $C \subset \Sigma^n$ is $(q, \delta, \varepsilon)$-locally testable if there is an oracle algorithm $A$ of query complexity $q$ such that*

- *For every $x \in C$, $Pr[A^x \ accepts] = 1$.*
- *For every string $y \in \Sigma^n$ such that $\text{dist}(y, C) \geq \delta$, $Pr[A^y \ rejects] \geq \varepsilon$.*

## 7.2 Short PCPs

Our main theorem in this section is,

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

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

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

From this result, we derive $SAT \in PCP_{1-\frac{1}{\text{poly} \log n},1}[\log_2(n \cdot \text{poly} \log n), O(1)]$. More precisely,

**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*

- *$size(G) \leq n(\log n)^{c_1}$ and $|\Sigma| = O(1)$.*
- *If $\varphi$ is satisfiable, then $\text{UNSAT}(G) = 0$.*

- *If $\varphi$ is not satisfiable, then* $\text{UNSAT}(G) \geq \frac{1}{(\log n)^{c_2}}$.

Before proving the lemma, let us see how it implies Theorem 7.1,

**Proof:**( of Theorem 7.1 ) Given a SAT instance of size $n$, we rely on Lemma 7.3 to reduce it to a constraint graph $G$ whose size we denote by $m = n \cdot (\log n)^{c_1}$. Then, we apply the main theorem (Theorem 4.1) iteratively $k = c_2 \cdot \log\log m < 2c_2 \log\log n$ times. This results in a constraint-graph $G'$ for which $\text{UNSAT}(G') \geq \min(2^k \cdot \text{UNSAT}(G)\,,\, \alpha) = \alpha$, and such that $size(G') = C^{c_2 \log\log m} \cdot m \leq n \cdot (\log n)^{c_1 + 2c_2 \log C} = n \cdot \text{poly}\log n$.

To get an error-probability of $\frac{1}{2}$ one can apply standard techniques for effi cient amplifi cation through expander neighborhoods. ■

**Proof:**( of Lemma 7.3 ) Theorem 7.2 yields some constants $a_1, a_2 > 0$ and a reduction from SAT to a system $\Psi_0$ of at most $m = n \cdot (\log n)^{a_1}$ constraints, each over at most $(\log n)^{a_2}$ variables such that satisfi able inputs go to satisfi able systems, and unsatisfi able inputs result in systems for which any assignment satisfi es at most $\frac{1}{2}$ of the constraints. Our goal is to reduce the number of queries per constraint. Basically, this is done by introducing new variables over a large alphabet, which enables few queries in a naive way (which causes the rejection probability to deteriorate). Then, the alphabet size is reduced through composition.

**Two-variable Constraints** For each constraint in $\Psi_0$, let us introduce one new (big) variable. This variable will take values over alphabet $\Sigma = \{0,1\}^{(\log n)^{a_2}}$ that supposedly represent values to all of the original (small) variables queried in that constraint. The number of big variables is $m = n \cdot (\log n)^{a_1}$. Introduce $(\log n)^{a_2}$ new constraints per big variable: Each constraint will query the big variable and exactly one of the small variables queried by the corresponding constraint. The constraint will check that the value for the big variable satisfi es the original constraint, and that it is consistent with the second (small) variable. Call this system $\Psi$ and observe that $|\Psi| = n \cdot (\log n)^{a_1 + a_2}$.

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

**Composition** We next apply composition to reduce the alphabet size from $\log|\Sigma| = \text{poly}\log n$ to $O(1)$. This is exactly as done in Lemma 3.6 except that we are somewhat more restricted in our choice of the assignment tester algorithm $\mathcal{P}$ (or equivalently: a PCP of Proximity), in that the output size of $\mathcal{P}$ must be polynomial in the input size. Observe that we only require that the size of the output is *polynomial* (and not quasi-linear) in the input size, so there is no circularity in our argument. 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 3.6: 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 $\text{UNSAT}(\Psi') \geq \text{UNSAT}(\Psi) \cdot \varepsilon = \Omega((\log n)^{-(a_2+1)}) = \frac{1}{\text{poly}\log n}$ where the middle equality holds since the rejection probability $\varepsilon$ is a constant. Since the input size for $\mathcal{P}$ was the size of one constraint in $\Psi$, i.e., $\text{poly}\log n$, it follows that the size of the constraint system output by $\mathcal{P}$ is also $\text{poly}\log n$. This means that $|\Psi'| = |\Psi| \cdot \text{poly}\log n = n \cdot \text{poly}\log n$ ■

## 7.3 Short Locally Testable Codes

A similar construction to that of Theorem 7.1 can be used to obtain locally-testable codes with inverse poly-logarithmic rate (i.e., mapping $k$ bits to $k \cdot \text{poly} \log k$ bits), that are testable with a constant number of queries.

The way we go about it is by relying on a variant of the main theorem (Theorem 4.1). Recall that the main theorem is a reduction from $G$ to $G' = (prep(G)^t) \circ \mathcal{P}$. We will need a stronger kind of reduction, that is an assignment tester (also called a PCP of Proximity), as defined in Definition 3.1.

In the next section we will prove that the main amplification step (as in Theorem 4.1) can also work for assignment-testers. Formally,

**Theorem 8.1** *There exists $t \in \mathbb{N}$ such that given an assignment-tester with constant-size alphabet $\Sigma$ and rejection probability $\epsilon$, one can construct an assignment-tester with the same alphabet and rejection probability at least $\min(2\epsilon, 1/t)$, such that the output size of the new reduction is bounded by at most by a constant factor times the output size of the given reduction.*

Just as our main theorem (Theorem 4.1) could be combined with the construction of [BS05] yielding a short PCP, Theorem 8.1 can be combined with the construction of [BS05] to yield short PCPs of Proximity / assignment-tester reductions.

**Corollary 7.4** *There exists an assignment-tester with constant size alphabet, and constant rejection probability $\epsilon > 0$, such that inputs of size $n$ are transformed to outputs of size at most $n \cdot \text{poly} \log n$.*

**Proof:** As in the proof of Theorem 7.1, we begin with a lemma that follows from the construction of [BS05],

**Lemma 7.5** *There exist a polynomial-time assignment-tester with constant alphabet size and rejection probability $\epsilon \geq \frac{1}{(\log n)^{O(1)}}$, such that inputs of size $n$ are transformed to outputs of size at most $n \cdot \text{poly} \log n$.*

The difference between this lemma and Lemma 7.3 is that here we require the reduction to be an assignment-tester. This can be derived from the construction of [BS05], in a similar way to the proof of Lemma 7.3.

Let $\mathcal{A}_0$ be the assignment-tester from Lemma 7.5. Let $\mathcal{A}_i$ be the result of applying Theorem 8.1 on $\mathcal{A}_{i-1}$. For $i = O(\log \log n)$, the reduction $\mathcal{A}_i$ will have the required parameters. ∎

Finally, we claim that Corollary 7.4 directly implies the existence of locally testable codes of rate $1/\text{poly} \log n$.

**Corollary 7.6** *For every $\delta > 0$ there exists an $\varepsilon = \Omega(\delta) > 0$, and an infinite family of codes $\{C_N\}_N$ with rate $1/\text{poly} \log N$, such that $C_N$ is $(2, \delta, \varepsilon)$-locally-testable.*

**Proof:** We apply the construction of [BGH$^+$04, Construction 4.3] to the assignment-tester from Corollary 7.4. We give a brief sketch. We construct $C_N$ as follows. Fix $n \in \mathbb{N}$ and let $C'_n \subset \Sigma^n$ be an error correcting code with rate and distance $\Theta(n)$. Let $\Phi$ be a circuit over variables $X = \{x_1, \ldots, x_n\}$ that accepts iff the assignment for $X$ is a codeword in $C'_n$. We can assume that $|\Phi| = O(n)$ (using, e.g., expander codes [SS96]). Run the reduction of Corollary 7.4 on $\Phi$, and let $G$ be the output constraint graph, $size(G) = n \cdot \text{poly} \log n$. Let $Y = V \setminus X$ be the new variables added by the reduction, and denote $m = |Y|$, $m \leq n \cdot \text{poly} \log n$. Let $\ell = \frac{2m}{\delta n}$, $N = n\ell + m$, and define a new code

$$C_N = \left\{ a^\ell b \in \Sigma^{n\ell+m} \;\middle|\; a \in C'_n, b \in \Sigma^m \text{ and } \text{UNSAT}_\sigma(G) = 0 \text{ where } \sigma|_X = a \text{ and } \sigma|_Y = b \right\} \subset \Sigma^N .$$

where $a^\ell b$ denotes the concatenation of $\ell$ copies of $a$ with $b$. Clearly, the rate of $C_N$ is $1/\text{poly}\log N$. We claim that $C_N$ is $(2, \delta, \varepsilon)$-locally-testable. Here is the testing algorithm for a given word $w \in \Sigma^{n\ell+m}$. Denote the $i$-th bit of $w$ by $w_i$.

1. Flip a random coin.
2. If heads, choose a random $i \in [n]$ and a random $j \in \{1, 2, \ldots, \ell - 1\}$, and accept iff $w_i = w_{i+j\cdot\ell}$
3. If tails, choose a random constraint in $G$. View $w[1, \ldots, n]$ as an assignment for $X$ and $w[n\ell + 1, \ldots, n\ell + m]$ as an assignment for $Y$. Accept iff the constraint is satisfied by this assignment.

Clearly, every $w \in C_N$ passes the test with probability 1. If $\text{dist}(w', C_N) > \delta$, then for any codeword $\sigma = a^\ell b \in C_N$, since $m \leq n\ell \cdot \frac{\delta}{2}$, the strings $w'$ and $\sigma$ must differ on $\delta n\ell/2$ of their first $n\ell$ bits. The reader may verify that the test rejects with probability at least $\Omega(\delta)$. ■

**Remark 7.1 (Constant Relative Distance)** *The codes above also have a constant relative distance. This follows almost immediately from the distance of $C'_n$, except for the following caveat. A problem would arise if for some assignment $a$ for $X$ that satisfies $\Phi$ there were two assignments $b_1, b_2$ for $Y$ such that both $\text{UNSAT}_{a\cup b_1}(G) = 0$ and $\text{UNSAT}_{a\cup b_2}(G) = 0$. This would imply that $a^\ell b_1, a^\ell b_2 \in C_N$, and their distance can be quite small. However, this can be ruled out if every assignment $a$ has only one unique assignment $b$ such that $\text{UNSAT}_{a\cup b}(G) = 0$. This can be ensured here, and therefore we conclude that the above does yield codes with constant relative distance.*

# 8 Adapting the Main Theorem for Assignment-Testers

In this section we show how to adapt the main amplification step (Theorem 4.1), that was described as a reduction between constraint graphs, to work within the more demanding framework of an assignment-tester. This gives an extension of our main theorem (and Corollary 4.2), to assignment-testers / PCPs of proximity.

**Theorem 8.1** *There exists $t \in \mathbb{N}$ such that given an assignment-tester with constant-size alphabet $\Sigma$ and rejection probability $\epsilon$, one can construct an assignment-tester with the same alphabet and rejection probability at least $\min(2\epsilon, 1/t)$, such that the output size of the new reduction is bounded by at most by a constant factor times the output size of the given reduction.*

Suppose we have a reduction taking $\Phi$ to $G$. We construct from $G$ a new graph $G'$ and prove that the reduction taking $\Phi$ to $G$ and then to $G'$ has the desired properties.

Let $H = (prep(G))^t$ be the result of running the preprocessing step (Lemma 3.1) and then raising the resulting constraint graph to the power $t$. What are the variables of $H$? Going from $G$ to $prep(G)$ each variable $v \in V$ is split into many copies, and we denote the set of copies of $v$ by $[v]$. Next ,going from $prep(G)$ to $H = (prep(G))^t$, the variables of $H$ are identical to those of $prep(G)$, while taking values from a larger alphabet. So denoting the variables of $H$ by $V_H$, we have $V_H = \cup_{v\in V}[v]$. Syntactically, $V_H$ is disjoint from $V$, although the values for $V_H$ are supposed to "encode" values for $V$. Indeed, an assignment $\sigma : V \to \Sigma$ can be mapped to an assignment $\sigma_2 : V_H \to \Sigma^{d^{t/2}}$ that "encodes" it, by the following two steps.

1. First define a mapping $\sigma \mapsto \sigma_1$, where the assignment $\sigma_1 : V_H \to \Sigma$ for $prep(G)$ is defined by assigning all copies of $v$ the same value as $\sigma(v)$:

$$\forall v \in V \; w \in [v], \quad \sigma_1(w) \stackrel{\triangle}{=} \sigma(v). \tag{10}$$

Let us name this mapping $m_1$. Observe also that given any assignment for $prep(G)$, $\sigma' : V_H \to \Sigma$, it can be "decoded" into an assignment for $G$ according to maximum likelihood as follows. Simply set $\sigma = m_1^{-1}(\sigma')$ to be an assignment $\sigma : V \to \Sigma$ for which $m_1(\sigma)$ is closest[6] in Hamming distance to $\sigma'$.

2. Next, define a mapping $\sigma_1 \mapsto \sigma_2$, where the assignment $\sigma_2 : V_H \to \Sigma^{d^t/2}$ for $H$ is defined by assigning each vertex $w$ a vector consisting of the $\sigma_1$-values of all of its neighbors at distance $t/2$

$$\forall w \in V_H, \quad \sigma_2(w)_v \stackrel{\triangle}{=} \sigma_1(v) \text{ for all } v \text{ at distance } t/2 \text{ from } w \text{ in } G\,. \tag{11}$$

Let us name this mapping $m_2$, and again, given any assignment $\sigma' : V_H \to \Sigma^{d^t/2}$ for $(prep(G))^t$ it can be "decoded" into an assignment for $prep(G)$ as follows. Simply set $\sigma = m_2^{-1}(\sigma')$ to be an assignment $\sigma : V_H \to \Sigma$ for which $m_2(\sigma)$ is closest in Hamming distance to $\sigma'$.

Going back to our reduction, we recall that in order for our reduction to be an assignment-tester, our output constraint graph must have the variables $X$ of $\Phi$ contained in its set of variables. Then, we must also verify that the completeness and soundness conditions (that refer to $X$) hold.

**The Graph $H'$** We next transform $H$ to $H'$ so as to include $X$ among the variables of $H'$. The vertices of $H'$ will be $V_H \cup X$. The constraints of $H'$ will include all of the constraints of $H$, and also additional constraints that will check that the assignment for $V_H$ is a correct encoding, according to the mapping $m_2 \circ m_1$ which maps $\sigma$ to $\sigma_2$ (via $\sigma_1$), of the assignment for $X$.

We describe the constraints between $X$ and $V_H$ by the following randomized procedure. Let $A : V_H \to \Sigma^{d^t/2}$ and let $a : X \to \{0,1\}$.

1. Select $x \in_R X$.
2. Select $z \in_R [x]$ (recall that $[x]$ is the set of vertices in $prep(G)$ that are copies of $x$).
3. Take a $t/2$-step random walk in $prep(G)$ starting from $z$, and let $w$ be the endpoint of the walk. Accept if and only if $A(w)_z = a(x)$.

For every possible random choice of the test, we will place (an edge and) a constraint between $w$ and $x$, that accepts iff the test accepts. We will reweigh the constraints (by duplication) so that the weight of the comparison constraints defined by the random procedure is half of the total weight of the edges. This completes the description of $H'$. Observe that the size of $H'$ is at most a constant times the size of $G$, because $prep(G)$ is $d$-regular for $d = O(1)$, so every vertex $w \in V_H$ participates in exactly $d^{t/2} = O(1)$ new comparison constraints. The next lemma states that the reduction from $\Phi$ to $H'$ is an assignment-tester with large alphabet, and rejection probability $\Theta(\sqrt{t}) \cdot \epsilon$.

**Lemma 8.2** *Assume $\varepsilon < 1/t$, and fix $a : X \to \{0,1\}$.*

- *If $a \in \mathrm{SAT}(\Phi)$, there exists $b : V_H \to \Sigma^{d^t/2}$ such that $\mathrm{UNSAT}_{a \cup b}(H') = 0$.*
- *If $\delta = \mathrm{dist}(a, \mathrm{SAT}(\Phi)) > 0$, then for every $b : V_H \to \Sigma^{d^t/2}$, $\mathrm{UNSAT}_{a \cup b}(H') > \delta \cdot \min(\frac{1}{16}, (\beta_1 \beta_2 \sqrt{t}/2)\varepsilon)$.*

We prove this lemma shortly below. First, note that the constraint graph $H'$ is almost what we need, except that it is defined over the alphabet $\Sigma^{d^t/2}$, rather than over $\Sigma$. Let us now proceed to construct the final graph $G$.

[6] Breaking ties arbitrarily.

**The Graph $G'$** To reduce the alphabet of $H'$, we use composition. I.e., we assume that we have at our disposal an assignment-tester $\mathcal{P}$ such that its rejection probability is some constant $\varepsilon_0 > 0$, and its alphabet is $\Sigma$. We make no requirements about the length of the output of $\mathcal{P}$, because we will only run it on constant size inputs. For example, we can use the construction given in Section 6, whose rejection-probability is a constant (and this parameter is implicit in the Definition 3.1).

Now, the Composition Theorem of assignment-testers, [DR04, Theorem 3.7], states that given any two such reductions, their composition is well defined (it is essentially described in the proof of Lemma 3.6 herein) and is itself an assignment-tester, with the following parameters:

- The *alphabet size* is that of the inner reduction $\mathcal{P}$, thus the constraints in $G'$ are over alphabet $\Sigma$, as desired.
- The *output size* is the product of the output sizes of the two reductions. In our case, this means that the output size of the reduction $\Phi \Rightarrow H'$ is multiplied by a *constant* factor that is the maximum size of the output of $\mathcal{P}$ when run on a constraint of $H'$.
- The *rejection probability* is the product of the rejection probabilities of the two reductions. Thus, it is a constant multiple ($\varepsilon_0$) of the rejection probability of the reduction $\Phi \Rightarrow H'$. Since this value was $\min(\frac{1}{16}, (\beta_1\beta_2\sqrt{t}/2)\varepsilon)$, by choosing $t$ large enough, even after multiplying by $\varepsilon_0$ it is still larger than $2\varepsilon$ for all small enough (but constant) $\varepsilon$.

This completes the description of the transformation taking $\Phi$ to $G'$. It remains to prove Lemma 8.2.

**Proof:** (of Lemma 8.2) In this proof, there are four constraint graphs that we keep in mind

$$G \quad \Rightarrow \quad prep(G) \quad \Rightarrow \quad H = (prep(G))^t \quad \Rightarrow \quad H' .$$

Recall that we encode assignments for $G$ via $m_1$, obtaining assignments for $prep(G)$. These are encoded via $m_2$, giving assignments for $H$. We can also go in the opposite direction where an assignment for $H$ can be decoded into an assignment for $prep(G)$ via $m_2^{-1}$, and similarly an assignment for $prep(G)$ can be decoded via $m_1^{-1}$ into as assignment for $G$.

- Suppose $a \in \mathrm{SAT}(\Phi)$. Then, by assumption on the reduction from $\Phi$ to $G$, there is an assignment $b : V \to \Sigma$ such that $\sigma = a \cup b$ satisfies all constraints in $G$. The assignment $\sigma$ is mapped, via $m_1$ to an assignment $\sigma_1$ for $prep(G)$, and $\sigma_1$ in turn is mapped via $m_2$ into an assignment for $H$: $\sigma_2 : V_H \to \Sigma^{d^{t/2}}$. By the completeness of the preprocessing and the powering, $\sigma_2$ will satisfy all constraints in $H$. It is easy to verify that $\sigma_2$ will also satisfy (together with $a$) all of the new comparison constraints, so $\mathrm{UNSAT}_{a\cup\sigma_2}(H') = 0$
- Assume now $\mathrm{dist}(a, \mathrm{SAT}(\Phi)) = \delta > 0$. Fix some assignment $b : V_H \to \Sigma^{d^{t/2}}$. We will show that the assignment $a \cup b$ violates many of the constraints. The idea is to first "decode" $b$ (through maximum likelihood decoding of the encoding $m_2 \circ m_1$) thereby getting an assignment $b_0 : V \to \Sigma$. Then, we show that either $b_0$ is close to the assignment $a$, in which case it is far from $\mathrm{SAT}(\Phi)$, so by amplification $b$ must violate many of the constraints in $H$. Otherwise, if $b_0$ is far from $a$, then many (a constant fraction!) of the comparison constraints will fail.

  So let $b_1 = m_2^{-1}(b)$ be an assignment for the vertices of $prep(G)$, and let $b_0 = m_1^{-1}(b_1)$ be an assignment for the vertices of $G$, where notation $m_1^{-1}, m_2^{-1}$ refers to maximum-likelihood decoding. There are two cases.

  - If $\mathrm{dist}(b_0|_X, a) \leq \delta/2$, then $\mathrm{dist}(b_0|_X, \mathrm{SAT}(\Phi)) > \delta/2$ by the triangle inequality. Since the reduction from $\Phi$ to $G$ is an assignment-tester with rejection probability $\varepsilon$, this means

that no matter what $b_0|_{(V \setminus X)}$ is, $\text{UNSAT}_{b_0}(G) > \varepsilon\delta/2$. Now we claim that $b_1$ must also be violating a similar fraction of the constraints of $prep(G)$:

$$\text{UNSAT}_{b_1}(prep(G)) > \varepsilon\delta/2 \cdot \beta_1. \tag{12}$$

Indeed, recall Corollary 3.4 that asserts that for every $G$ and for every assignment $\sigma'$ for $prep(G)$, the fraction of constraints of $prep(G)$ violated by $\sigma'$ is proportional to the fraction of constraints of $G$ violated by $m_1^{-1}(\sigma')$. Plugging in $b_1$ for $\sigma'$, and since $b_0 = m_1^{-1}(b_1)$, this implies (12).

Next, we claim that $b$ must be violating an even larger fraction of $H = (prep(G))^t$ than $\text{UNSAT}_{b_1}(prep(G))$:

$$\text{UNSAT}_b((prep(G))^t) > \text{UNSAT}_{b_1}(prep(G)) \cdot \beta_2\sqrt{t}\,. \tag{13}$$

Indeed, recall Corollary 5.5 that states that for every $G$ and every assignment $\vec{\sigma}$ for $G^t$, the fraction of constraints of $G^t$ violated by $\vec{\sigma}$ is larger than the fraction of constraints of $G$ violated by the "popular opinion" assignment, by factor $\Omega(\sqrt{t})$. Observe that indeed $m_2^{-1}(\vec{\sigma})$ is the "popular opinion" assignment. Plugging in $b$ for $\vec{\sigma}$, and since $b_1 = m_2^{-1}(b)$, (and since $\varepsilon < 1/t$) this implies (13). Combining (12) and (13),

$$\text{UNSAT}_b(H) > \varepsilon\delta/2 \cdot \beta_1 \cdot \beta_2\sqrt{t}\,.$$

Since the constraints of $H$ are half of the constraints of $H'$, we have

$$\text{UNSAT}_{a\cup b}(H') \geq \frac{1}{2}\text{UNSAT}_b(H) \geq \varepsilon\delta/4 \cdot \beta_1 \cdot \beta_2\sqrt{t}$$

– If $\text{dist}(b_0|_X, a) > \delta/2$, then we will show that $\delta/8$ fraction of the comparison constraints reject. Indeed, with probability $\delta/2$ step 1 in the randomized test selects a variable $x \in X$ for which $b_0(x) \neq a(x)$. Conditioned on that, consider the probability that step 2 selects a $z \in [x]$ such that $b_1(z) \neq a(x)$. Since $b_0(x)$ is, by definition, a most popular value among values assigned by $b_1$ to the copies of $x$; and since by conditioning, $a(x) \neq b_0(x)$, this probability is at least $1/2$. Conditioned on both previous events occurring, step 3 selects a vertex $w$ for which $b(w)_z \neq a(x)$, with probability at least $1/2$ (for similar reasoning). Altogether, with probability at least $\frac{\delta}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \delta/8$ the test rejects. This means that at least $\delta/16$ of the total number of tests reject, i.e., $\text{UNSAT}_{a\cup b}(H') \geq \delta/16$.

We have proven that for $\delta = \text{dist}(a, \text{SAT}(\Phi))$, and for every assignment $b$, the rejection probability $\text{UNSAT}_{a\cup b}(H')$ is either at least $\delta \cdot \frac{1}{16}$ or at least $\delta \cdot (\beta_1\beta_2\sqrt{t}/2 \cdot \varepsilon)$.

This completes the proof. ∎

Theorem 8.1 also gives an immediate combinatorial construction of assignment-testers or PCPPs in the same way that the main theorem (Theorem 4.1) was used to derive the PCP Theorem (Corollary 4.2).

**Corollary 8.3** *There is an assignment-tester, with constant alphabet, constant rejection probability, and polynomial output length.*

The proof follows by observing that the identity reduction is an assignment-tester with rejection probability $1/n$, and that $O(\log n)$ iterations of Theorem 8.1 bring it down to a constant, while causing only a polynomial increase in the output size.

## 9 Amplification and Parallel-Repetition

The celebrated parallel repetition theorem of Raz [Raz98] gives different method of amplification. The theorem asserts that given a system of constraints $\mathcal{C}$ with $\text{UNSAT}(\mathcal{C}) = \alpha$, the $t$-parallel-repetition system, $\mathcal{C}^t_{||}$, will have $\text{UNSAT}(\mathcal{C}^t_{||}) \geq 1 - (1-\alpha)^{\Theta(t)}$.

For small values of $\alpha$ this equals $\Theta(t) \cdot \alpha$ and is comparable with our construction (our amplification lemma only asserts $\text{UNSAT}(\mathcal{C}^t) > \Theta(\sqrt{t}) \cdot \alpha$, but there is a modification[7] of our construction, due to Jaikumar Radhakrishnan [Rad05], that replaces $\sqrt{t}$ by $t$).

For larger (say, constant) values of $\alpha$, the parallel repetition brings the unsat-value closer and closer to 1, a feature that is very useful in inapproximability reductions. On the other hand, our amplification stops to make any progress for constant $\alpha > 0$, as is demonstrated in an example of Bogdanov [Bog05].

In terms of the size of the system, our construction incurs a linear blowup, while in the parallel repetition case the system size grows exponentially in $t$. The linear-blowup feature is crucial for our iterative proof of the PCP theorem.

One may view our amplification as a derandomization of the parallel-repetition theorem. By derandomization, it is meant that some carefully chosen subset of the original system is being considered. We recall that Feige and Kilian proved that no generic derandomization of the parallel-repetition theorem is possible [FK95]. Their result focuses on a range of parameters that does not apply to our setting. This raises questions about the limits of such constructions in a wider range of parameters.

## Acknowledgements

I am thankful to Omer Reingold and Luca Trevisan for many discussions, especially ones about combinatorial analyses of graph powering, which were the direct trigger for the amplification lemma. I would like to thank Jaikumar Radhakrishnan for very helpful comments on an earlier version of this manuscript. I would also like to thank David Arnon, Miki Ben-Or, Ehud Friedgut, Oded Goldreich, 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.

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

[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.

[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.

[7] Amplification by factor $\Theta(t)$ is achieved by the following modified definition of $\dot{G}$: the vertices stay the same, and the alphabet is $\Sigma^{d^t}$ as before. The edges of $G^t$ are weighted, and described by the following random process: choose a vertex $v$ at random and choose a second vertex by taking a random walk form $v$ that stops after each step with probability $1/t$. The constraints are defined as before.

[Bog05] Anrej Bogdanov. Gap amplification fails below 1/2. Comment on ECCC TR05-046, can be found at `http://eccc.uni-trier.de/eccc-reports/2005/TR05-046/commt01.pdf`, 2005.

[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.

[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.

[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.

[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.

[FK95] Uri Feige and Joe Kilian. Impossibility results for recycling random bits in two-prover proof systems. In *Proc. 27th ACM Symp. on Theory of Computing*, pages 457–468, 1995.

[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.

[GS97] O. Goldreich and S. 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.

[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.

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

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

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

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

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

[Rad05] Jaikumar Radhakrishnan. Private communication. 2005.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, June 1998.

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

[RVW02] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. *Annals of Mathematics*, 155(1):157–187, 2002.

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Trans. Inform. Theory*, 42(6, part 1):1710–1722, 1996. Codes and complexity.

# 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 5.3** *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*

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

**Proof:** Write $n = m + r$ for some $0 \leq r \leq \sqrt{n}$. 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 $a \leq r \leq \sqrt{n}$,

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

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

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

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[8]. For $\alpha \subset [n]$, define

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

[8] 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 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, $\sum_\alpha |\hat{A}_\alpha|^2 = \langle A, A\rangle = 1$.

**The Test.** Let $\psi : [n] \to \{-1,1\}$ be some predicate, and fix $\tau = \frac{1}{100}$. 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. 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,
$$= \frac{7}{8} - \frac{1}{8}\mathbb{E}_{g,h}[A(g)A(h)] - \frac{1}{8}\mathbb{E}_{f,g,h}[A(f)A(g)A(h)]\,.$$
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 $\hat{A}_\phi = 0$. Let us denote the acceptance probability by $1-\varepsilon$. This implies that the second expectation (whose value let us name $W$) must be at most $-1+\tau+8\varepsilon$. 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 $\tau = \frac{1}{100}$ is fixed, we have $\rho = O(\varepsilon)$.

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 $\approx 1/4$ which is certainly $\Omega(\delta)$.)

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. ∎



ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'