On Parallel Repetition of PCPs

Alessandro Chiesa
EPFL, Lausanne, Switzerland

Ziyi Guan
EPFL, Lausanne, Switzerland

Burcu Yıldız
EPFL, Lausanne, Switzerland

Abstract

Parallel repetition refers to a set of valuable techniques used to reduce soundness error of probabilistic proofs while saving on certain efficiency measures. Parallel repetition has been studied for interactive proofs (IPs) and multi-prover interactive proofs (MIPs). In this paper we initiate the study of parallel repetition for probabilistically checkable proofs (PCPs).

We show that, perhaps surprisingly, parallel repetition of a PCP can *increase soundness error*, in fact bringing the soundness error to one as the number of repetitions tends to infinity. This "failure" of parallel repetition is common: we find that it occurs for a wide class of natural PCPs for NP-complete languages. We explain this unexpected phenomenon by providing a characterization result: the parallel repetition of a PCP brings the soundness error to zero if and only if a certain "MIP projection" of the PCP has soundness error strictly less than one. We show that our characterization is tight via a suitable example. Moreover, for those cases where parallel repetition of a PCP does bring the soundness error to zero, the aforementioned connection to MIPs offers preliminary results on the rate of decay of the soundness error.

Finally, we propose a simple variant of parallel repetition, called consistent parallel repetition (CPR), which has the same randomness complexity and query complexity as the plain variant of parallel repetition. We show that CPR brings the soundness error to zero for *every* PCP (with non-trivial soundness error). In fact, we show that CPR decreases the soundness error at an exponential rate in the repetition parameter.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases probabilistically checkable proofs, parallel repetition

Digital Object Identifier 10.4230/LIPIcs.ITCS.2024.34

Related Version *Full Version*: `https://eprint.iacr.org/2023/1714`

Funding The authors are partially supported by the Ethereum Foundation.

Acknowledgements The authors thank Ngoc Khanh Nguyen, Guy Weissenberg, Eylon Yogev, and Mingnan Zhao for valuable feedback and comments on earlier drafts of this paper. The authors thank anonymous reviewers of ITCS for valuable comments and suggestions that have improved this paper.

1 Introduction

Probabilistic proofs play a fundamental role in theoretical computer science, and are invaluable tools in cryptography, facilitating applications such as delegation of computation, zero knowledge proofs, and more. Probabilistic proofs comprise notions such as interactive proofs (IPs), multi-prover interactive proofs (MIPs), probabilistically checkable proofs (PCPs), and others. A central goal in this area is constructing probabilistic proofs with small soundness error (the maximum probability that any prover convinces the verifier to accept an instance that is not in the language).

Parallel repetition. Parallel repetition is a class of ideas aimed at reducing soundness error without increasing key efficiency measures such as round complexity or query complexity. Parallel repetition has been defined and studied for IPs and MIPs, as we review in more detail in Section 1.2.

- Parallel repetition for IPs is straightforward: the t-wise parallel repetition of an IP with soundness error β is a new IP, *with the same round complexity*, whose soundness error is β^t.
- Parallel repetition for MIPs is less understood. The t-wise parallel repetition of a 1-round MIP with soundness error $\beta < 1$ is a new 1-round MIP, *with the same number of provers*, whose soundness error β_t tends to 0 as t tends to infinity. In special cases (e.g., 2 provers), we know that β_t decays exponentially in t (with certain dependencies on the MIP). The rate of decay in the general case is a major open problem.

Parallel repetition underlies many results in hardness of approximation, which rely on custom-made PCP constructions in which one of the steps is to apply Raz's Theorem on parallel repetition for MIPs [15].[1]

Parallel repetition for PCPs. In this paper we study *parallel repetition for PCPs*.

A PCP for a language L is a proof system where a verifier $\mathbf{V}$, given as input an instance $\mathbb{x}$ and given oracle access to a string $\pi\colon [\mathsf{l}] \to \Sigma$, probabilistically queries a few locations of π and then decides whether to accept or reject. The soundness error β of the PCP verifier $\mathbf{V}$ is a function that, given any instance $\mathbb{x} \notin L$, outputs (an upper bound on) the maximum acceptance probability of $\mathbf{V}(\mathbb{x})$ across all strings $\pi\colon [\mathsf{l}] \to \Sigma$.

For $t \in \mathbb{N}$, the *t-wise parallel repetition* of the PCP verifier $\mathbf{V}$ is a PCP verifier $\mathbf{V}_t$ that receives as input an instance $\mathbb{x}$ and oracle access to a string $\Pi\colon [\mathsf{l}]^t \to \Sigma^t$, and works as follows.

> $\mathbf{V}_t^{\Pi}(\mathbb{x})$: For every $i \in [t]$, sample fresh randomness ρ_i for $\mathbf{V}$ and deduce the queries $(q_{i,j})_{j\in[\mathsf{q}]}$ that $\mathbf{V}(\mathbb{x};\rho_i)$ makes. For every $j \in [\mathsf{q}]$, query Π at location $(q_{i,j})_{i\in[t]} \in [\mathsf{l}]^t$ to obtain an answer $(a_{i,j})_{i\in[t]} \in \Sigma^t$. For every $i \in [t]$, check that $\mathbf{V}(\mathbb{x};\rho_i)$ accepts given query answers $(a_{i,j})_{j\in[\mathsf{q}]}$.

The above definition is folklore (e.g., see [3]). The basic question that we study in this paper is:

> If $\mathbf{V}$ has soundness error β then what is the soundness error β_t of $\mathbf{V}_t$?

Surprisingly, the effect of parallel repetition for PCPs on soundness error has not been studied so far. It may be natural to guess that parallel repetition for PCPs works similarly as for MIPs: the soundness error tends to zero as the number of repetitions tends to infinity and, in some cases (say, 2-query PCPs), one can show that the rate of decay is exponential (with the rate depending in some way on the PCP).

In this paper we initiate a systematic study of parallel repetition for PCPs, and show that the above natural guess is incorrect: parallel repetition for PCPs fails to work in many cases and, in contrast, a variant of parallel repetition that we introduce always works. Overall, our work contributes an initial set of results on a basic question about probabilistic proofs, which we believe merits further study due to its fundamental nature.

---

[1] Roughly, a common recipe is to transform the PCP obtained from the PCP Theorem into an MIP, then apply parallel repetition for MIPs, then transform the resulting MIP back into a PCP (with certain special properties), and then perform further optimizations/customizations to establish the desired hardness of approximation result.

1.1 Our results

Our first result shows that, in the general case, parallel repetition of a PCP does not work as expected: there is a 2-query PCP (with non-trivial soundness error) for which parallel repetition *increases* soundness error.

▶ **Theorem 1** (informal). *There exists a* 2*-query PCP for an* NP*-complete language* L *with soundness error* $\beta < 1$ *such that the soundness error* β_t *of its* t*-wise parallel repetition tends to* 1*:*

$$\text{for every } \mathbb{x} \notin L,\ \lim_{t\to\infty} \beta_t(\mathbb{x}) = 1 \ .$$

In fact, for infinitely many $\mathbb{x} \notin L$, $\beta_{t+1}(\mathbb{x}) > \beta_t(\mathbb{x})$ *for every* $t \in \mathbb{N}$ *(where* $\beta_1(\mathbb{x}) := \beta(\mathbb{x})$*).*

Counter to intuition, Theorem 1 refutes the sensible conjecture that $\beta(\mathbb{x})^t \leq \beta_t(\mathbb{x}) \leq \beta(\mathbb{x})$. This is in sharp contrast to the case of MIPs, where this basic relationship does hold for parallel repetition of MIPs.

Moreover, the PCP underlying Theorem 1 is not contrived: it is the "canonical" PCP for graph 3-coloring where the PCP string is the 3-color assignment for the given graph and the PCP verifier checks the colors of the two vertices of a random edge of the graph. Hence Theorem 1 tells us that, for *every* graph that is not 3-colorable, applying parallel repetition to this canonical PCP leads to soundness error 1 in the limit! This includes graphs that are far from being 3-colorable and, in particular, also those graphs generated by the PCP Theorem (for which the soundness error β of the canonical PCP is a constant bounded away from 1).

The failure of parallel repetition for PCPs is rather common. We show that it occurs for a wide class of PCPs for *constraint satisfaction problems* (CSPs).[2] We associate to any given CSP a corresponding "canonical" PCP: the PCP string is the assignment to the variables of the CSP, and the PCP verifier samples a random constraint of the CSP and checks if it is satisfied (by reading from the PCP string the variables involved in that constraint). We prove that parallel repetition fails for the canonical PCP of any *symmetric CSP* (informally, a CSP where every constraint "looks the same"); the class of symmetric CSPs includes well-known NP-complete problems such as graph 3-coloring (as above), independent set, clique, and others.

▶ **Lemma 2** (informal). *Let* PCP *be the canonical PCP for a symmetric CSP. If the CSP is not satisfiable (for every assignment to the variables there is at least one constraint that is not satisfied by the assignment), then, letting* β_t *be the soundness error for the* t*-wise repetition of* PCP *(and* $\beta_1 := \beta$*), it holds that*

$$\forall\, t \in \mathbb{N},\ \beta_{t+1} \geq \beta_t \quad \text{and} \quad \beta > 0 \implies \lim_{t\to\infty} \beta_t > 0 \ .$$

(If $\beta = 0$ *then it is straightforward to see that* $\beta_t = 0$ *for every* $t \in \mathbb{N}$*.)*

Lemma 2 does not extend to non-symmetric CSPs. One can construct a non-symmetric instance of 3Sat (an example of a CSP) whose canonical PCP satisfies $\beta > 0$ and $\lim_{t\to\infty} \beta_t = 0$.

[2] A *constraint satisfaction problem* (CSP) is a list of constraints over a list of variables. Each constraint is a predicate over some of the variables. The CSP is satisfiable if there exists an assignment of the variables that satisfies all constraints simultaneously.

Since parallel repetition for PCPs does not always work, next we ask: *when does it work*? We identify a criterion that characterizes when parallel repetition reduces soundness error of a PCP to zero (in the limit). Briefly, we associate with each PCP a corresponding MIP, which we call its *MIP projection*.

▶ **Definition 3** (informal). *The* **MIP projection** *of a PCP verifier* $\mathbf{V}$ *is an MIP verifier* $\mathcal{V}$ *that works as follows: sample randomness* ρ *for* $\mathbf{V}$ *and deduce the queries* $(q_j)_{j\in[\mathsf{q}]}$ *that* $\mathbf{V}(\mathbb{x};\rho)$ *makes; then, for every* $j \in [\mathsf{q}]$*, send* q_j *to the* j*-th prover to obtain an answer* b_j*; finally, check that* $\mathbf{V}(\mathbb{x};\rho)$ *accepts given the answers* $(b_j)_{j\in[\mathsf{q}]}$*.*

The number of provers in the MIP projection of a PCP equals the number of queries of the PCP, and the MIP projection has *no consistency checks*. This is unlike the well-known q-query PCP to 2-prover MIP transformation, where all queries are sent to one prover and one of the queries at random is sent to the other prover, for consistency. The soundness error of the MIP projection is at least the soundness error of the PCP, and it can be strictly larger. Our result is that the soundness error of the MIP projection tells us precisely when parallel repetition of a PCP works: parallel repetition of a PCP drives soundness error to zero if and only if the MIP projection has non-trivial soundness error.

▶ **Theorem 4** (informal). *Consider a PCP for a language* L*, and let* β_t *be the soundness error of the* t*-wise parallel repetition of the PCP. Letting* β_{MIP} *be the soundness error of the MIP projection of the PCP,*

$$\text{for every } \mathbb{x} \notin L,\ \lim_{t\to\infty} \beta_t(\mathbb{x}) = 0 \iff \beta_{\mathsf{MIP}}(\mathbb{x}) < 1 \ .$$

Theorem 4 helps us interpret Theorem 1: the PCP in Theorem 1 is such that its MIP projection has soundness error 1, and therefore parallel repetition does not drive the soundness error to 0. In fact, for that example, the limit achieved in Theorem 1 is 1 (not just some constant greater than 0). Theorem 4 also explains the aforementioned 3Sat example: the canonical PCP for that 3Sat instance has an MIP projection with soundness error less than 1, so parallel repetition of that PCP works just fine.

More generally, our characterization is essentially tight in the following sense: if $\beta_{\mathsf{MIP}}(\mathbb{x}) = 1$ then our analysis shows that $\lim_{t\to\infty} \beta_t(\mathbb{x}) \in [1/2^{\mathsf{vr}}, 1]$, where vr is the randomness complexity of the given (non-repeated) PCP verifier; and we show that there exists a PCP for which (on infinitely many instances not in the language) parallel repetition leads, in the limit, to soundness error $1/2^{\mathsf{vr}}$.

Rate of decay for parallel repetition. The above results (Theorem 1, Lemma 2, Theorem 4) consider the *limiting behavior* of the soundness error of the parallel repetition of a PCP. Next we study the *rate of decay*: when parallel repetition drives the soundness error of a PCP to zero in the limit, at what rate does soundness error decrease (as the number of repetitions increases)?

Our proof of Theorem 4 (which we outline in Section 2.2) tells us that the rate of decay of parallel repetition for a PCP is upper bounded by the rate of decay of parallel repetition for the corresponding MIP projection. In particular, we can use known results on the rate of decay of parallel repetition for MIPs ([9, 15, 7, 8, 12, 14, 16, 1, 17]), if applicable to the MIP projection.

We additionally prove that, in general, we cannot hope for a rate of decay that is better than for the MIP projection of the PCP: if a PCP is an "evaluation of an MIP", parallel repetition of the PCP decreases soundness error at the same rate as parallel repetition of its MIP projection. Understanding the rate of decay of parallel repetition for PCPs in general (when parallel repetition does work), remains an open problem.

▶ **Definition 5** (informal). *Let* $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[\mathsf{k}]}, \mathcal{V})$ *be a (one-round)* k*-prover MIP. The* **PCP evaluation** *of the MIP verifier* $\mathcal{V}$ *is a PCP verifier* $\mathbf{V}$ *that works as follows: sample randomness* ρ *for* $\mathcal{V}$ *and deduce the messages* $(a_j)_{j\in[\mathsf{k}]}$ *that* $\mathcal{V}(\mathbb{x};\rho)$ *sends to the MIP provers; then, for every* $j \in [\mathsf{k}]$*, query the PCP string at* (j, a_j) *to obtain answers* b_j*; finally, check that* $\mathcal{V}(\mathbb{x}, \rho, (b_j)_{j\in[\mathsf{k}]}) = 1$.

▶ **Lemma 6** (informal). *Consider the PCP evaluation of an MIP for a language* L *with soundness error less than* 1*. Let* β_t *be the soundness error of the* t*-wise parallel repetition of the PCP. Let* $\beta_{\mathsf{MIP},t}$ *be the soundness error of the* t*-wise parallel repetition of the MIP. Then*

$$\text{for every } \mathbb{x} \notin L \text{ and } t \in \mathbb{N},\ \beta_t(\mathbb{x}) = \beta_{\mathsf{MIP},t}(\mathbb{x}) < 1 \ .$$

A variant that always works. Finally, we identify a natural variant of parallel repetition for PCPs that we can prove always works, in the sense that the soundness error tends to zero for every PCP (with non-trivial soundness error). The definition below adds a natural consistency test across repetitions within the repeated verifier, which incurs no additional randomness or queries.

▶ **Definition 7** (informal). *The* **consistent parallel repetition** *of a PCP verifier* $\mathbf{V}$ *is a new PCP verifier* $\hat{\mathbf{V}}_t$ *that works the same as a parallel repetition verifier and, in addition, it checks that any duplicate queries across different repetitions are answered consistently.*

▶ **Theorem 8** (informal). *Consider a PCP for a language* L *with soundness error* $\beta < 1$*, and let* $\hat{\beta}_t$ *be the soundness error for the* t*-wise consistent parallel repetition of the PCP. Then*

$$\text{for every } \mathbb{x} \notin L,\ \hat{\beta}_t(\mathbb{x}) \leq O_{\mathbb{x}}(1) \cdot \beta(\mathbb{x})^t \ ,$$

where $O_{\mathbb{x}}(1)$ *hides a constant determined by the PCP and* $\mathbb{x}$ *(and independent of* t*).*

In particular, Theorem 8 implies that

$$\text{for every } \mathbb{x} \notin L,\ \beta(\mathbb{x}) < 1 \implies \lim_{t\to\infty} \hat{\beta}_t(\mathbb{x}) = 0 \ .$$

Note that, in contrast to the case of parallel repetition of MIPs, the rate of decay in Theorem 8 achieves the desired exponential decay up to a leading multiplicative constant (not in the exponent).

Theorem 8 enables the application of (consistent) parallel repetition in new regimes. For example consider the case of a PCP with constant soundness error β and super-constant query complexity $\mathsf{q} = \omega(1)$. Converting such a PCP into a 2-prover MIP via a standard transformation leads to a large soundness error $1 - \frac{1-\beta}{\mathsf{q}} = 1 - o(1)$, which makes the use of parallel repetition for MIPs too expensive in this case (Footnote 1). Instead, our work shows that one can directly use parallel repetition for PCPs to reduce the soundness error β to an arbitrary constant while preserving the query complexity q.

Open questions. Our work motivates basic questions about parallel repetition for PCPs.

1. What is the necessary and sufficient condition for parallel repetition to increase the soundness error? Theorem 1 gives examples for which parallel repetition increases the soundness error. Perhaps one could prove a more fine-grained version of Theorem 4 to also characterize when parallel repetition increases the soundness error.

2. When parallel repetition works for a given PCP (equivalently, the PCP's MIP projection has non-trivial soundness error), what is the rate of decay? We know that the rate of decay is no worse than that for parallel repetition of the MIP projection, and sometimes it equals that. But perhaps one could say more (e.g., via a direct analysis of the rate of decay, without invoking facts about the rate of decay for MIPs).
3. Can the rate of decay for consistent parallel repetition in Theorem 8 be improved? The hidden constant achieved in our analysis is large, and we suspect that it can be improved. Or perhaps a different analysis may establish an alternative expression for the rate of decay that is better for smaller values of t.

1.2 Related work

Parallel repetition for PCPs is a folklore definition but it has not been studied.[3] Below we summarize prior work on parallel repetition for IPs and MIPs, and also explain how direct product testing considers a distinct question. Separately, parallel repetition is also studied in a cryptographic context (e.g. for interactive arguments); we do not discuss this line of work since our setting is information-theoretic.

Parallel repetition for IPs. An interactive proof (IP) is a protocol where a prover and a verifier exchange messages and after that the verifier outputs a decision bit denoting whether to accept or reject; both prover and verifier are probabilistic algorithms. The t-wise repetition of an IP is a new IP where the prover and verifier run, simultaneously and in lockstep, t independent executions of the given IP. One can show that if no prover can convince the verifier to accept with probability greater than β then no prover can convince the repeated verifier to accept with probability greater than β^t. The proof for this statement is delicate (e.g., see [10, Appendix C.1]), but otherwise parallel repetition for IPs is straightforward.

Parallel repetition for MIPs. A multi-prover interactive proof (MIP) is a protocol where *multiple* provers exchange messages with a single verifier and after that the verifier outputs a decision bit denoting whether to accept or reject; the provers are allowed to share randomness but otherwise are not allowed to communicate during the interaction. The t-wise repetition of an MIP is similarly defined to the case of an IP: it is a new MIP where the prover and verifier run, simultaneously and in lockstep, t independent executions of the given MIP (with the same provers). Parallel repetition for MIPs has been studied in a line of work leading to notable progress, but a comprehensive understanding remains a challenging open problem.

Briefly, parallel repetition of any MIP decreases the soundness error to zero as the number of repetitions tends to infinity (provided that the initial soundness error is strictly less than 1) [18]; however the analysis only shows a slow rate of decay. The rate of decay is known to *not* be β^t [9]; in fact, sometimes parallel repetition yields a soundness error that is exponentially larger than the "ideal" β^t [7, 8, 16]. That said, parallel repetition of 2-prover MIPs (with non-trivial soundness error) does decrease soundness error exponentially fast, at a rate that depends on certain aspects of the MIP [15]; the rate of decay is studied and optimized in a line of works [12, 14, 1, 17].

[3] Parallel repetition for PCPs is occasionally mentioned in the literature (e.g., see [3]) but only informally, and giving the impression that it behaves the same as parallel repetition for MIPs. Our results show that this is not the case.

Direct product tests. A *direct product test* is a proximity test for the set of functions that can be expressed as tensors of another function: for a given $t \in \mathbb{N}$, all functions $\Pi\colon [l]^t \to \Sigma^t$ for which there exists $\pi\colon [l] \to \Sigma$ such that $\Pi = ((\pi[q_1], \ldots, \pi[q_t]))_{(q_1,\ldots,q_t)\in[l]^t}$. Introduced in [11], direct product tests are useful in PCP constructions, and they have been studied in a line of works [5, 6, 2, 13, 4].

Parallel repetition and direct product tests are *different notions*, and in some applications direct product tests and parallel repetition are used in combination: either the function is far from a tensor, in which case the direct product test accepts with small probability; or the function is close to a tensor, in which case parallel repetition needs to "work" only for functions that are close to a tensor (a much weaker goal).

2 Techniques

We summarize the main ideas behind our results. Each subsection below outlines the ideas contained in the corresponding technical section.

2.1 Parallel repetition for PCPs does not always work

We comment on the proof of Theorem 1.

Consider the NP-complete language *graph 3-coloring*, which consists of graphs $G = (V, E)$ whose vertices can be labeled via colors in $\{0, 1, 2\}$ such that every edge in the graph has vertices labeled with different colors. Moreover, consider a simple PCP for this language:

- A PCP string $\pi\colon V \to \{0, 1, 2\}$ is a coloring of the vertices of the given graph G.
- The PCP verifier, given the graph G, samples a random edge $\{u, v\}$ of the graph G and checks that $\pi[u] \neq \pi[v]$ (by querying π at locations u and v). We assume that the edge is sampled so that $u < v$ according to, e.g., a lexicographic order on the vertices of the graph G.

If G is 3-colorable then setting π to any 3-coloring of G makes the PCP verifier accept with probability 1. If G is not 3-colorable then, for every PCP string π, the probability that the PCP verifier accepts π is at most $\frac{|E|-1}{|E|}$ (at least one edge is not satisfied in any coloring); in fact, the probability is at most $\mathsf{val}(G)$, the maximum fraction of valid edges across any coloring of G (which can be less than $\frac{|E|-1}{|E|}$).

The soundness error tends to 1. For every graph G that is not 3-colorable, the soundness error of the parallel repetition of the above PCP tends to 1.

Consider for example the 2-wise parallel repetition. We argue that there is a PCP string $\widetilde{\Pi}_2\colon V^2 \to \{0, 1, 2\}^2$ that convinces the 2-wise repeated PCP verifier to accept with probability at least $1 - \left(\frac{|E|-1}{|E|}\right)^2$.

For every query $(q_1, q_2) \in V^2$, set $\widetilde{\Pi}_2[(q_1, q_2)]$ to be $(0, 0)$ if q_1 or q_2 is the smallest non-isolated vertex in G (with respect to the lexicographic order of V) and $(1, 1)$ otherwise. (A smallest non-isolated vertex always exists because G is not 3-colorable.) Let $\mathbf{Q}_1 = (q_{1,1}, q_{1,2})$ and $\mathbf{Q}_2 = (q_{2,1}, q_{2,2})$ be the two queries of $\mathbf{V}_2$. By construction of $\mathbf{V}$, we know that $q_{1,1} < q_{2,1}$ and $q_{1,2} < q_{2,2}$. Hence at least one of $\widetilde{\Pi}_2[\mathbf{Q}_1]$ and $\widetilde{\Pi}_2[\mathbf{Q}_2]$ is $(1, 1)$. Thus $\mathbf{V}_2$ rejects if and only if both $\widetilde{\Pi}_2[\mathbf{Q}_1]$ and $\widetilde{\Pi}_2[\mathbf{Q}_2]$ are $(1, 1)$, which happens only when $\mathbf{V}_2$ queries an edge that is not adjacent to the smallest non-isolated vertex in both repetitions, which in turn happens with probability at most $\left(\frac{|E|-1}{|E|}\right)^2$.

In Figure 1 we give an example of $\widetilde{\Pi}_2$ for the 4-clique graph K_4, which is not 3-colorable. The smallest non-isolated vertex is v_1. Thus, in $\widetilde{\Pi}_2$ only query pairs that include v_1 have the answer $(0,0)$, and all other queries are answered with $(1,1)$; see Table 1. As long as one of $q_{1,1}$ and $q_{1,2}$ is v_1, $\mathbf{V}_2$ accepts because it receives the answers $(0,0)$ and $(1,1)$.

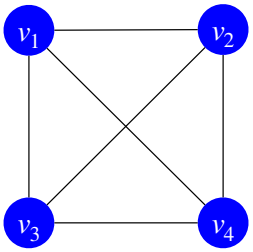

Figure 1 The 4-clique graph K_4.

Table 1 The PCP string $\widetilde{\Pi}_2$ for the 4-clique graph.

q_1 \ q_2	v_1	v_2	v_3	v_4
v_1	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
v_2	$(0,0)$	$(1,1)$	$(1,1)$	$(1,1)$
v_3	$(0,0)$	$(1,1)$	$(1,1)$	$(1,1)$
v_4	$(0,0)$	$(1,1)$	$(1,1)$	$(1,1)$

The above idea extends to the case of t-wise parallel repetition, where there is a PCP string $\widetilde{\Pi}_t \colon V^t \to \{0,1,2\}^t$ that convinces the t-wise repeated PCP verifier to accept with probability at least $1 - \left(\frac{|E|-1}{|E|}\right)^t$.

We conclude that, for every graph $\mathbb{x} = G$ that is not 3-colorable, $\lim_{t\to\infty} \beta_t(\mathbb{x}) = 1$.

The soundness error strictly increases. Next we outline how we show that $\beta_{t+1}(\mathbb{x}) > \beta_t(\mathbb{x})$ for infinitely many graphs $\mathbb{x} = G$ that are not 3-colorable.

First we explain why $\beta_2(K_4) > \beta_1(K_4) = \beta(K_4)$ for the 4-clique graph $K_4 = (V,E)$ (which is not 3-colorable). Let $n \coloneqq |V| = 4$ and $m \coloneqq |E| = 6$. Consider the coloring $\chi \coloneqq \{(v_1,0),(v_2,1),(v_3,2),(v_4,2)\}$ for K_4, shown in Figure 2. The coloring χ is a 3-coloring of $K_4' \coloneqq (V, E \setminus \{\{v_3,v_4\}\})$. Define $\widetilde{\Pi}_2$ to be $((\min\{\chi(u),\chi(v)\},\min\{\chi(u),\chi(v)\}))_{(u,v)\in V^2}$ (see Table 2). Let $\mathbf{Q}_1 = (q_{1,1},q_{1,2})$ and $\mathbf{Q}_2 = (q_{2,1},q_{2,2})$ be the two queries made by $\mathbf{V}_2$. By construction of $\mathbf{V}$, $q_{1,1} < q_{2,1}$ and $q_{1,2} < q_{2,2}$. Therefore, $\min\{\chi(q_{1,1}),\chi(q_{2,1})\} \le \min\{\chi(q_{2,1}),\chi(q_{2,2})\}$ by definition of χ. Note that $\mathbf{V}_2$ rejects only when $q_{1,1} = q_{1,2} = v_3$ and $q_{2,1} = q_{2,2} = v_4$ (which implies $\min\{\chi(q_{1,1}),\chi(q_{2,1})\} = \min\{\chi(q_{2,1}),\chi(q_{2,2})\}$). Therefore, we deduce that $\beta_2(K_4) \ge 1 - 1/m^2 = 35/36$. Since $\beta_1(K_4) \le 1 - 1/m = 5/6$, we conclude that $\beta_2(K_4) > \beta_1(K_4)$.

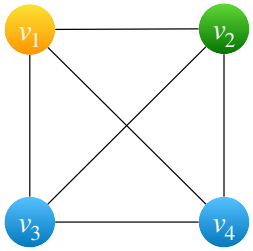

Figure 2 The 4-clique graph colored by χ.

Table 2 The PCP string $\widetilde{\Pi}_2$ for the 4-clique graph.

q_1 \ q_2	v_1	v_2	v_3	v_4
v_1	$(0,0)$	$(0,0)$	$(0,0)$	$(0,0)$
v_2	$(0,0)$	$(1,1)$	$(1,1)$	$(1,1)$
v_3	$(0,0)$	$(1,1)$	$(2,2)$	$(2,2)$
v_4	$(0,0)$	$(1,1)$	$(2,2)$	$(2,2)$

More generally, via similar ideas, for every $m \in \mathbb{N}$ with $m \ge 6$, we construct a graph G such that $\beta_t(G) = 1 - 1/m^t$ for every $t \in \mathbb{N}$, concluding the first part of Theorem 1. The graph G consists of a 4-clique and $m-6$ connected components of size 2; this amounts to $4 + 2\cdot(m-6) = 2m-8$ vertices and $6 + (m-6) = m$ edges (a 4-clique has 6 edges).

- We sketch why $\beta_t(G) \geq 1 - 1/m^t$. While the graph G is not 3-colorable (it contains a 4-clique), deleting one edge from the 4-clique makes the new graph G' 3-colorable. In particular, possibly after renaming the vertices, we obtain a 3-coloring χ such that:
  1. for every $u, v \in V$ such that $u < v$, $\chi(u) \leq \chi(v)$; and
  2. the size of the set $S := \{\{u, v\} \in (E \setminus \{v_{n-1}, v_n\}) : \chi(u) = \chi(v)\}$ is minimized.

  The rest of the argument is analogous to the case of the 4-clique graph.
- We argue that $\beta_t(G) < 1$ for every t, which implies that $\beta_t(G) \leq 1 - 1/m^t$ because the number of random choices for $\mathbf{V}_t$ is m^t (as each repetition of the verifier $\mathbf{V}$ samples one out of m edges). This follows from the (general) fact that, for every $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ and instance $\mathbb{x}$, $\beta_t(\mathbb{x}) = 1$ implies that $\beta(\mathbb{x}) = 1$.

  Indeed, suppose by way of contradiction that $\beta_t(\mathbb{x}) = 1$. Let $\widetilde{\Pi}_t$ be a PCP string for the t-wise repetition such that $\mathbf{V}_t^{\widetilde{\Pi}_t}(\mathbb{x})$ always accepts. Define the PCP string $\widetilde{\pi} := (\widetilde{\Pi}_t[i^t])_{i \in [\mathsf{l}]}$ for $\mathbf{V}$. For every randomness ρ for $\mathbf{V}$, $\mathbf{V}_t^{\widetilde{\Pi}_t}(\mathbb{x}; \rho^t)$ queries $(q_i)^t$ for every $i \in [\mathsf{q}]$ where $(q_1, \ldots, q_{\mathsf{q}})$ is the query list of $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho)$. Thus, for every randomness ρ for $\mathbf{V}$, $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho) = 1$ because $\mathbf{V}_t^{\widetilde{\Pi}_t}(\mathbb{x}; \rho^t) = 1$, which implies that $\beta(\mathbb{x}) = 1$.

2.2 When does parallel repetition for PCPs work?

We comment on the proof of Theorem 4, which characterizes the limiting behavior of parallel repetition of a PCP in terms of the soundness error of its MIP projection.

It is straightforward to show that the soundness error of the MIP projection of a PCP is at least the soundness error of the PCP. Hence, if the PCP has soundness error 1 then its MIP projection has soundness error 1. On the other hand, if the PCP has soundness error less than 1, then its MIP projection may or may not have soundness error less than 1. For example, the PCP for graph 3-coloring described above has soundness error less than 1 but its MIP projection has soundness error 1 (the first MIP prover always answers color 0 and the second MIP prover always answers color 1, regardless of the messages sent by the MIP verifier).

A key property is that performing an MIP projection and performing parallel repetition "commute": given a PCP, *the MIP projection of the parallel repetition of the PCP is the same as the parallel repetition of the MIP projection of the PCP.*

The MIP projection has soundness error < 1. Suppose that the MIP projection of the PCP has soundness error less than 1. Recall that parallel repetition of an MIP with soundness error less than 1 drives the soundness error to 0 [18]. Therefore, for every PCP whose MIP projection has soundness error less than 1, parallel repetition of the MIP projection drives soundness error to 0. Hence, the MIP projection of the parallel repetition of the PCP also has soundness error that tends to 0, which is an upper bound on the soundness error of the parallel repetition of the PCP.

The MIP projection has soundness error $= 1$. Conversely, for a given instance $\mathbb{x} \notin L$, suppose that the MIP projection has soundness error 1 (i.e., it has no soundness at all). Let $(\widetilde{\mathcal{P}}_i)_{i \in [\mathsf{q}]}$ be optimal malicious provers for the MIP projection. Let vr be the randomness complexity of the (non-repeated) PCP verifier $\mathbf{V}$. For every $t > 1$, we construct a PCP string $\widetilde{\Pi}_t$ that convinces the t-wise parallel repeated PCP verifier $\mathbf{V}_t$ to accept $\mathbb{x}$ with probability at least $1/2^{\mathsf{vr}}$ (a lower bound independent of t).

An initial guess would be to construct a PCP string $\widetilde{\Pi}_t$ for the repeated PCP verifier $\mathbf{V}_t$ as follows.

1. Initialize a repeated PCP string $\widetilde{\Pi}_t$ to be a string with an arbitrary symbol everywhere.
2. For every possible randomness choice $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t)$ of the repeated PCP verifier $\mathbf{V}_t$:
   a. Compute the query list $\mathbf{Q}$, where for every $i \in [\mathsf{q}]$, $\mathbf{Q}[i]$ is a t-tuple such that $\mathbf{Q}[i][j]$ is the i-th query made by the PCP verifier $\mathbf{V}$ given instance $\mathbb{x}$ and randomness ρ_j.
   b. For every $i \in [\mathsf{q}]$ and $j \in [t]$, compute the answer $\mathsf{ans}_{i,j} \coloneqq \widetilde{\mathcal{P}}_i(\mathbb{x}, \mathbf{Q}[i][j])$ to the query $\mathbf{Q}[i][j]$.
   c. For every $i \in [\mathsf{q}]$, set $\widetilde{\Pi}_t[\mathbf{Q}[i]] \coloneqq (\mathsf{ans}_{i,j})_{j \in t}$.

It is tempting to conclude that $\widetilde{\Pi}_t$ convinces the repeated PCP verifier with probability 1 because $(\widetilde{\mathcal{P}}_i)_{i \in [\mathsf{q}]}$ convince the MIP projection verifier with probability 1. However, this is not true as we now explain.

We say that two (not necessarily distinct) randomness choices $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$ are *incompatible* if there exist distinct $i, j \in [\mathsf{q}]$ such that $\mathbf{Q}_1[i] = \mathbf{Q}_2[j]$, where $\mathbf{Q}_1, \mathbf{Q}_2$ are the query lists of the repeated PCP verifier $\mathbf{V}_t$ when given instance $\mathbb{x}$ and randomness $\boldsymbol{\rho}_1$, $\boldsymbol{\rho}_2$, respectively.

Intuitively, constructing a PCP string $\widetilde{\Pi}_t$ by considering answers from the MIP provers across incompatible randomness choices may lead to distinct answers for the same location in $\widetilde{\Pi}_t$, which hinders arguing that the repeated verifier $\mathbf{V}_t$ would accept on both randomness choices.

Consider the following example. Let ρ_1, ρ_2, ρ_3 be three distinct randomness choices for a 2-query PCP verifier. Let $\mathrm{Q}_1 \coloneqq (1, 2)$, $\mathrm{Q}_2 \coloneqq (2, 3)$, $\mathrm{Q}_3 \coloneqq (4, 1)$ be the query lists corresponding to ρ_1, ρ_2, ρ_3. For the 2-wise parallel repeated PCP verifier, the randomness choice $\boldsymbol{\rho}_1 \coloneqq (\rho_1, \rho_2)$ has query list $\mathbf{Q}_1 \coloneqq ((1, 2), (2, 3))$ and the randomness choice $\boldsymbol{\rho}_2 \coloneqq (\rho_3, \rho_1)$ has query list $\mathbf{Q}_2 \coloneqq ((4, 1), (1, 2))$. Note that $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$ are incompatible because $\mathbf{Q}_1[1] = \mathbf{Q}_2[2]$. In particular, in the construction of $\widetilde{\Pi}_t$ outlined above, $\widetilde{\mathcal{P}}_1$ separately gives some answers to query 1 and query 2, and $\widetilde{\mathcal{P}}_2$ may separately give different answers to query 1 and query 2. Hence we do not have a single value that we can assign to entry $(1, 2)$ in $\widetilde{\Pi}_t$, so we cannot conclude that the 2-wise parallel repeated PCP verifier $\mathbf{V}_2$ accepts on $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$.

One subtlety we neglect in the above discussion is: what happens if $\mathbf{Q}_1[i] = \mathbf{Q}_2[i]$ for some $i \in [\mathsf{q}]$? For any query list $\mathbf{Q}$ of the repeated PCP verifier, the i-th query $\mathbf{Q}[i]$ is answered by the i-th MIP prover $\widetilde{\mathcal{P}}_i$ as in the construction above. Therefore, $\mathbf{Q}_1[i] = \mathbf{Q}_2[i]$ does not lead to clashing answers in the PCP string $\widetilde{\Pi}_t$.

To avoid incompatibility, we find a "large" set S of randomness for the repeated PCP verifier such that S does not contain incompatible randomness choices. Let S be the set of repeated verifier randomness where the randomness for the last repetition is fixed to be some arbitrary string $\rho^* \in \{0,1\}^{\mathsf{vr}}$. Consider any $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2 \in S$ and any distinct $i, j \in [\mathsf{q}]$. Let $\mathbf{Q}_1, \mathbf{Q}_2$ be the query lists of $\mathbf{V}_t$ corresponding to $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2$. Since the PCP verifier $\mathbf{V}$ does not make duplicate queries (within the same query list), we know that $\mathbf{Q}_1[i][t] \neq \mathbf{Q}_2[j][t]$, and therefore $\mathbf{Q}_1[i] \neq \mathbf{Q}_2[j]$. Hence the set S does not contain incompatible randomness choices.

We construct a PCP string $\widetilde{\Pi}_t$ similarly to the above procedure, by going over all randomness choices in the set S. Since the MIP projection has soundness error 1, $\widetilde{\Pi}_t$ convinces the repeated PCP verifier with probability at least $|S| / (2^{\mathsf{vr}})^t = 1/2^{\mathsf{vr}}$, as desired.

2.3 Rate of decay of parallel repetition for PCPs

We sketch the proof of Lemma 6: if a PCP is the evaluation of an MIP (Definition 5) then the rate of decay of parallel repetition for this PCP is the same as the rate of decay of parallel repetition for the MIP. Let MIP be an MIP for a language L and let PCP be its PCP evaluation. Below:

- β_{MIP} is the soundness error of MIP;
- $\beta_{\mathsf{MIP},t}$ is the soundness error of the t-wise parallel repetition of MIP;

- β is the soundness error of PCP; and
- β_t is the soundness error of the t-wise parallel repetition of PCP.

Throughout, we fix an instance $\mathbb{x} \notin L$.

$\beta_{\mathsf{MIP}}(\mathbb{x}) < 1$ implies $\beta_t(\mathbb{x}) < 1$. It is not hard to show that $\beta(\mathbb{x}) = \beta_{\mathsf{MIP}}(\mathbb{x})$ (any prover strategy for MIP can be converted to an equally effective prover strategy for PCP, and vice versa). Therefore, if $\beta_{\mathsf{MIP}}(\mathbb{x}) < 1$ then $\beta(\mathbb{x}) < 1$. In Section 2.1, we explained that $\beta_t(\mathbb{x}) = 1$ implies $\beta(\mathbb{x}) = 1$. Hence we conclude that $\beta(\mathbb{x}) < 1$ implies $\beta_t(\mathbb{x}) < 1$.

$\beta_t(\mathbb{x}) \leq \beta_{\mathsf{MIP},t}(\mathbb{x})$. In Section 2.2 we mentioned that performing an MIP projection and performing parallel repetition "commute". In other words,

$$\beta'_{\mathsf{MIP},t}(\mathbb{x}) = \beta''_{\mathsf{MIP},t}(\mathbb{x})$$

where $\beta'_{\mathsf{MIP},t}$ is the soundness error of the parallel repetition of the MIP projection of PCP and $\beta''_{\mathsf{MIP},t}$ is the soundness error of the MIP projection of the parallel repetition of PCP. Moreover, since the soundness error of the MIP projection of a PCP is at least the soundness error of the PCP,

$$\beta_t(\mathbb{x}) \leq \beta''_{\mathsf{MIP},t}(\mathbb{x}) \ .$$

On the other hand, performing an *MIP projection* and a *PCP evaluation* are "inverses" of each other: MIP is essentially the same proof system as the MIP projection of PCP, up to a minor syntactic difference in the verifier alphabet that does not affect the soundness error. By Definition 3 and Definition 5, if MIP is a k-prover MIP with verifier alphabet $\Sigma_{\mathcal{V}}$, then the MIP projection of PCP has verifier alphabet $[\mathsf{k}] \times \Sigma_{\mathcal{V}}$. This "almost equivalence" still holds after parallel repetition, which implies that

$$\beta_{\mathsf{MIP},t}(\mathbb{x}) = \beta'_{\mathsf{MIP},t}(\mathbb{x}) \ .$$

Therefore, we conclude that $\beta_t(\mathbb{x}) \leq \beta''_{\mathsf{MIP},t}(\mathbb{x}) = \beta'_{\mathsf{MIP},t}(\mathbb{x}) = \beta_{\mathsf{MIP},t}(\mathbb{x})$.

$\beta_t(\mathbb{x}) \geq \beta_{\mathsf{MIP},t}(\mathbb{x})$. Consider malicious provers $(\widetilde{\mathcal{P}}_{t,i})_{i\in[\mathsf{k}]}$ for the parallel repetition of MIP. We construct a malicious proof string $\widetilde{\Pi}$ for the parallel repetition of PCP such that, for every verifier randomness $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t)$,

$$\langle (\widetilde{\mathcal{P}}_{t,i})_{i\in[\mathsf{k}]}, \mathcal{V}_t(\mathbb{x}, \boldsymbol{\rho}) \rangle = 1 \implies \mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho}) = 1 \ ,$$

which implies that $\beta_t(\mathbb{x}) \geq \beta_{\mathsf{MIP},t}(\mathbb{x})$ as desired.

For every $i \in [\mathsf{k}]$, $\mathbf{Q}_i$ denotes the i-th query of $\mathbf{V}_t(\mathbb{x}; \boldsymbol{\rho})$. Let Q_j be the query list of the verifier $\mathbf{V}(\mathbb{x}; \rho_j)$ for PCP for every $j \in [t]$. By the definition of parallel repetition,

$$\mathbf{Q}_i = (\mathrm{Q}_1[i], \ldots, \mathrm{Q}_t[i]), \text{ where } \mathrm{Q}_j[i] = (i, \mathcal{V}(\mathbb{x}, \rho_j)[i]) \text{ for all } j \in [t] \ .$$

By the definition of MIP projection, $\mathcal{V}_t(\mathbb{x}, \boldsymbol{\rho})$ sends the message $(\mathcal{V}(\mathbb{x}, \rho_j)[i])_{j\in[t]}$ to the i-th prover and receives $\boldsymbol{b}_i \coloneqq \widetilde{\mathcal{P}}_{t,i}((\mathcal{V}(\mathbb{x}, \rho_j)[i])_{j\in[t]})$. Therefore, if $\mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho})$ gets $\boldsymbol{b}_i$ as answer for its i-th query $\mathbf{Q}_i$, $\mathcal{V}_t(\mathbb{x}, \boldsymbol{\rho}, (\boldsymbol{b}_i)_{i\in[\mathsf{q}]}) = 1$ implies $\mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho}) = 1$.

With the above argument, we construct a malicious proof string $\widetilde{\Pi}$ for the parallel repetition of PCP:

1. Initialize a repeated PCP string $\widetilde{\Pi}$ to be a string with an arbitrary symbol everywhere.
2. For every possible query $\mathbf{Q}$ of the repeated PCP verifier $\mathbf{V}_t$:
   a. Parse $\mathbf{Q}$ as $((i_j, a_j))_{j\in t}$).
   b. Set $\widetilde{\Pi}[\mathbf{Q}] \coloneqq \widetilde{\mathcal{P}}_{t,i_1}((a_j)_{j\in t})$.

Observe that

$$\widetilde{\Pi}[\mathbf{Q}_i] = \widetilde{\mathcal{P}}_{t,i}((\mathcal{V}(\mathbb{x}, \rho_j)[i])_{j\in[t]}) = \boldsymbol{b}_i \ .$$

Therefore, we conclude that if the parallel repeated MIP verifier $\mathcal{V}_t$ accepts, then the parallel repeated PCP verifier $\mathbf{V}_t$ with oracle access to $\widetilde{\Pi}$ also accepts.

2.4 Parallel repetition for the canonical PCP for CSPs

We discuss the proof of Lemma 2.

Fix a CSP instance $\mathbb{x}$ that is *not* satisfiable, which means that for every assignment to the variables there exists (at least) one constraint that is not satisfied by the assignment. This means that the canonical PCP for this CSP instance $\mathbb{x}$ has soundness error $\beta(\mathbb{x}) < 1$, because, no matter the PCP string, there is some probability that the PCP verifier checks a constraint that is not satisfied by the PCP string.

Suppose that the CSP instance $\mathbb{x}$ is symmetric. In other words, consider two constraints, C_1 over variables X_1 and C_2 over variables X_2 in $\mathbb{x}$; any assignment to X_1 that satisfies C_1 directly induces an assignment to X_2 that satisfies C_2 (the i-th variable in X_2 is assigned the value of the i-th variable in X_1). We outline why $\beta(\mathbb{x}) > 0$ implies that $\lim_{t\to\infty} \beta_t(\mathbb{x}) > 0$, that is, parallel repetition fails to reduce the soundness error to 0 in the limit.[4] By our Theorem 4, it suffices to argue that $\beta_{\mathsf{MIP}}(\mathbb{x}) = 1$, namely, that the MIP projection of the canonical PCP has soundness error 1 for the (unsatisfiable) CSP instance $\mathbb{x}$. Since $\beta(\mathbb{x}) > 0$, there exist an assignment a and PCP verifier randomness ρ such that $\mathbf{V}^a(\mathbb{x}; \rho) = 1$; let S_ρ be the locations of a queried by $\mathbf{V}$ with randomness ρ. Now consider the malicious MIP provers $(\widetilde{\mathcal{P}}_i)_i$ where each $\widetilde{\mathcal{P}}_i$ always answers with $a[S_\rho[i]]$ (we assume there is an implicit ordering of elements in S_ρ). Let $\mathcal{V}$ be the verifier for the MIP projection of the canonical PCP. Since the $\mathbb{x}$ is symmetric, $\mathbf{V}^a(\mathbb{x}; \rho) = 1$ implies $\mathcal{V}(\mathbb{x}, \rho_{\mathsf{MIP}}, (a[S_\rho[i]])_i) = 1$ for every MIP verifier randomness ρ_{MIP}.

Next we outline why $\beta_{t+1}(\mathbb{x}) \geq \beta_t(\mathbb{x})$ (the first part of Lemma 2). This is rather counter-intuitive: the $(t+1)$-wise repetition should be harder to win compared to t-wise repetition. However, because the CSP instance $\mathbb{x}$ is symmetric, we can design a PCP string $\widetilde{\Pi}_{t+1}$ for the $(t+1)$-wise repetition using a PCP string $\widetilde{\Pi}_t$ for the t-wise repetition without decreasing the winning probability. For simplicity, we outline the proof for $\beta_2(\mathbb{x}) \geq \beta_1(\mathbb{x}) = \beta(\mathbb{x})$, which can be directly extended to work for every $t \in \mathbb{N}$.

If $\beta(\mathbb{x}) = 0$ then the claim holds trivially so assume that $\beta(\mathbb{x}) > 0$, which means that there exist a PCP string (i.e., an assignment) $\widetilde{\pi}$ and PCP verifier randomness ρ such that $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho) = 1$. Since the CSP instance $\mathbb{x}$ is symmetric, we can use the answers used for the first PCP randomness ρ_1 also for the second PCP randomness ρ_2: we define $\widetilde{\Pi}_2$ by setting $\widetilde{\Pi}_2[(q_1, q_2)] \coloneqq (\widetilde{\pi}[q_1], \widetilde{\pi}[q_1])$ for every (q_1, q_2).

Let $\boldsymbol{\rho} = (\rho_1, \rho_2)$ be a randomness for $\mathbf{V}_2$ such that $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho_1) = 1$. We know that $\mathbf{V}(\mathbb{x}; \rho_1, \widetilde{\pi}[q_1], \widetilde{\pi}[q_2]) = 1$ where q_1 and q_2 are the two queries made by $\mathbf{V}$ under randomness ρ_1. Since every constraint in $\mathbb{x}$ checks the same function, $\mathbf{V}(\mathbb{x}; \rho, \widetilde{\pi}[q_1], \widetilde{\pi}[q_2]) = 1$ for every verifier randomness ρ. Hence $\mathbf{V}_2^{\widetilde{\Pi}_2}(\mathbb{x}; \boldsymbol{\rho}) = 1$.

[4] If $\beta(\mathbb{x}) = 0$ (no assignment satisfies any constraint) then one can show that $\beta_t(\mathbb{x}) = 0$ for every $t \in \mathbb{N}$.

There are at most $\beta(\mathbb{x}) \cdot 2^{\mathsf{vr}} \cdot 2^{\mathsf{vr}}$ choices of randomness $\boldsymbol{\rho} = (\rho_1, \rho_2)$ for $\mathbf{V}_2$ such that $\widetilde{\mathbf{V}^{\pi}}(\mathbb{x}; \rho_1) = 1$. Hence, we conclude that $\beta_2(\mathbb{x}) \geq \frac{\beta(\mathbb{x}) \cdot 2^{\mathsf{vr}} \cdot 2^{\mathsf{vr}}}{(2^{\mathsf{vr}})^2} = \beta(\mathbb{x})$.

We exemplify the above reasoning in the case of the canonical PCP $(\mathbf{P}, \mathbf{V})$ for graph 3-coloring. Fix a graph G that is not 3-colorable. Given any PCP string $\widetilde{\pi}$ for the canonical PCP, define $\widetilde{\Pi}_2 := ((\widetilde{\pi}[q_1], \widetilde{\pi}[q_1]))_{(q_1,q_2)}$. Fix a randomness $\boldsymbol{\rho} = (\rho_1, \rho_2)$. Let $(q_{1,1}, q_{1,2})$ and $(q_{2,1}, q_{2,2})$ be the two queries of $\mathbf{V}_2(G; \boldsymbol{\rho})$. Let $(\mathsf{ans}_{1,1}, \mathsf{ans}_{1,2}) := \widetilde{\Pi}_2[(q_{1,1}, q_{1,2})]$ and $(\mathsf{ans}_{2,1}, \mathsf{ans}_{2,2}) := \widetilde{\Pi}_2[(q_{2,1}, q_{2,2})]$. By construction of $\widetilde{\Pi}_2$, we know that $\mathsf{ans}_{1,1} = \mathsf{ans}_{1,2}$ and $\mathsf{ans}_{2,1} = \mathsf{ans}_{2,2}$. Since the PCP verifier $\mathbf{V}$ for graph 3-coloring always checks whether the two colors it gets are different, we know that $\mathbf{V}(G; \rho_1, \mathsf{ans}_{1,1}, \mathsf{ans}_{1,2}) = \mathbf{V}(G; \rho_2, \mathsf{ans}_{2,1}, \mathsf{ans}_{2,2})$. Therefore, if ρ_1 is an accepting randomness with respect to $\widetilde{\pi}$, $\boldsymbol{\rho}$ is an accepting randomness with respect to $\widetilde{\Pi}_2$. The same counting argument as above gives us the desired result.

2.5 Consistent parallel repetition always works

We discuss the proof of Theorem 8.

In Section 2.1 the malicious PCP string for the parallel repetition of the canonical PCP for graph 3-coloring exploits inconsistent answers across different repetitions. If the repeated PCP verifier were to check consistency of the answers to the same queries across repetitions, then that PCP string would fail to convince the repeated PCP verifier with such high probability.

This inspires the variant of parallel repetition for PCPs in Definition 7, where the repeated PCP verifier additionally checks that any duplicate queries are answered consistently (which means that the repeated PCP verifier no longer is the conjunction of independent "games" as is usually the case in parallel repetition). Theorem 8 tells us that consistent parallel repetition works for every PCP. Note that, in stark contrast, parallel repetition for MIPs always works (brings the soundness error to zero if the MIP has non-trivial soundness error) *without the need for any consistency checks across repetitions* [18].

In this overview we only briefly discuss the limiting behavior of consistent parallel repetition: we outline why if a PCP has soundness error $\beta(\mathbb{x}) < 1$ then $\lim_{t\to\infty} \hat{\beta}_t(\mathbb{x}) = 0$, where $\hat{\beta}_t$ is the soundness error of the t-wise consistent parallel repetition of the PCP.

The winning set of a PCP string is the set of randomness strings that lead the PCP verifier to accept. For the t-wise repeated PCP verifier, a choice of randomness is a list $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t)$, where each ρ_i is a choice of randomness for the given PCP verifier.

We argue that, given a PCP string $\widetilde{\Pi}$ for the repeated PCP verifier, for every $\boldsymbol{\rho}$ in the winning set of $\widetilde{\Pi}$, we can construct a malicious PCP string $\widetilde{\pi}$ from $\widetilde{\Pi}$ such that every ρ_i is in the winning set of $\widetilde{\pi}$.

The soundness error of the PCP gives an upper bound on the size of the winning set of $\widetilde{\pi}$. On the other hand, the maximum number of distinct elements for every $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t)$ in the winning set of $\widetilde{\Pi}$ is a lower bound on the size of winning set of $\widetilde{\pi}$. By a counting argument, we can deduce that if $\beta(\mathbb{x}) < 1$ then the size of the winning set of $\widetilde{\Pi}$ grows *slower* than the size of the set of all repeated PCP verifier randomness choices. This enables us to conclude that $\lim_{t\to\infty} \hat{\beta}_t(\mathbb{x}) = 0$.

References

1 Boaz Barak, Anup Rao, Ran Raz, Ricky Rosen, and Ronen Shaltiel. Strong parallel repetition theorem for free projection games. In *Proceedings of the 12th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, and of the 13th International Workshop on Randomization and Computation*, APPROX-RANDOM '09, pages 352–365, 2009.

2 Irit Dinur and Elazar Goldenberg. Locally testing direct product in the low error range. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '08, pages 613–622, 2008.

3 Irit Dinur and Or Meir. Derandomized parallel repetition via structured PCPs. *Computational Complexity*, 20(2):207–327, 2011.

4 Irit Dinur and Inbal Livni Navon. Exponentially small soundness for the direct product Z-Test. In *Proceedings of the 32nd Annual IEEE Conference on Computational Complexity*, CCC '17, pages 29:1–29:50, 2017.

5 Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '04, pages 155–164, 2004.

6 Irit Dinur and David Steurer. Direct product testing. In *Proceedings of the 29th Annual IEEE Conference on Computational Complexity*, CCC '14, pages 188–196, 2014.

7 Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45:634–652, 1998.

8 Uriel Feige and Oleg Verbitsky. Error reduction by parallel repetition – a negative result. *Combinatorica*, 22:461–478, 2002.

9 Lance Fortnow, John Rompel, and Michael Sipser. On the power of multi-prover interactive protocols. In *Theoretical Computer Science*, pages 156–161, 1988.

10 Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*, volume 17 of *Algorithms and Combinatorics*. Springer, 1998.

11 Oded Goldreich and Shmuel Safra. A combinatorial consistency lemma with application to proving the PCP theorem. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, APPROX-RANDOM '97, pages 67–84, 1997.

12 Thomas Holenstein. Parallel repetition: simplifications and the no-signaling case. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, STOC '07, pages 411–419, 2007.

13 Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. New direct-product testers and 2-query PCPs. *SIAM Journal on Computing*, 41:1722–1768, 2012.

14 Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM Journal on Computing*, 40:1871–1891, 2011.

15 Ran Raz. A parallel repetition theorem. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, STOC '95, pages 447–456, 1995.

16 Ran Raz. A counterexample to strong parallel repetition. *SIAM Journal on Computing*, 40:771–777, 2011.

17 Ran Raz and Ricky Rosen. A strong parallel repetition theorem for projection games on expanders. In *Proceedings of the 27th Annual IEEE Conference on Computational Complexity*, CCC '12, pages 247–257, 2012.

18 Oleg Verbitsky. Towards the parallel repetition conjecture. *Theoretical Computer Science*, 157:277–282, 1996.