On Parallel Repetition of PCPs

Alessandro Chiesa
alessandro.chiesa@epfl.ch
EPFL

Ziyi Guan
ziyi.guan@epfl.ch
EPFL

Burcu Yıldız
burcu.yildiz@epfl.ch
EPFL

November 24, 2023

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.

Keywords: probabilistically checkable proofs; parallel repetition

Contents

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 [Raz95].[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 [DM11]).[2] 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).

[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.

[2] This is (very) different from "naive" repetition of the PCP verifier on a PCP string. Given a PCP string $\pi\colon [\mathsf{l}] \to \Sigma$, repeating the PCP verifier $\mathbf{V}$ for t times, each time accessing π, reduces the soundness error from β to β^t, but increases the query complexity from q to $t \cdot \mathsf{q}$. In contrast, parallel repetition of a PCP does not affect the query complexity q.

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.

1.1 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*:*

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

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).[3] 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 1 (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*

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

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

Lemma 1 does not extend to non-symmetric CSPs. In Section 7.4 we give a non-symmetric instance of 3SAT (an example of a CSP) whose canonical PCP satisfies $\beta > 0$ and $\lim_{t\to\infty} \beta_t = 0$.

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*.

[3] 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.

Definition 1 (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 2 (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 2 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 2 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 1, Theorem 2) 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 2 (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 ([FRS88; Raz95; Fei98; FV02; Hol07; Rao11; Raz11; BRRRS09; RR12]), 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 2 (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 2 (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

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

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 3 (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 3 (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*

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

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

In particular, Theorem 3 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 3 achieves the desired exponential decay up to a leading multiplicative constant (not in the exponent).

Theorem 3 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. Is there a characterization for when parallel repetition of a PCP does not decrease (or perhaps strictly increases) soundness error? Theorem 1 is an example of when soundness error never decreases, and the characterization in Theorem 2 only tells us when the soundness error does not tend to zero.
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 3 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.[4] 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 [Gol98, 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) [Ver96]; however the analysis only shows a slow rate of decay. The rate of decay is known to *not* be β^t [FRS88]; in fact, sometimes parallel repetition yields a soundness error that is exponentially larger than the "ideal" β^t [Fei98; FV02; Raz11]. 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 [Raz95]; the rate of decay is studied and optimized in a line of works [Hol07; Rao11; BRRRS09; RR12].

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 $\Pi\colon [\mathsf{l}]^t \to \Sigma^t$ for which there exists $\pi\colon [\mathsf{l}] \to \Sigma$ such that $\Pi = ((\pi[q_1], \ldots, \pi[q_t]))_{(q_1,\ldots,q_t)\in[\mathsf{l}]^t}$. Introduced in [GS97], direct product tests are useful in PCP constructions, and they have been studied in a line of works [DR04; DS14; DG08; IKW12; DN17].

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).

[4] Parallel repetition for PCPs is occasionally mentioned in the literature (e.g., see [DM11]) 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.

2 Techniques

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

2.1 Section 4: 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 $\mathbf{Q}_1 = (q_{1,1}, q_{2,1})$ and $\mathbf{Q}_2 = (q_{1,2}, q_{2,2})$ be the two queries of $\mathbf{V}_2$. By construction of $\mathbf{V}$, we know that $q_{1,1} < q_{1,2}$ and $q_{2,1} < q_{2,2}$. 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 $(1, 1)$.

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 $\chi \coloneqq \{(v_1, 0), (v_2, 1), (v_3, 2), (v_4, 2)\}$

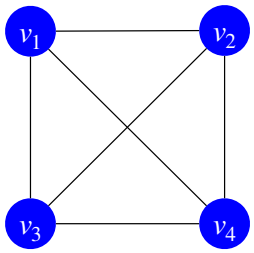

Figure 1: The 4-clique graph K_4.

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	$(0,0)$	$(1,1)$	$(1,1)$	$(1,1)$

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

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, $\min\{\chi(q_{1,1}), \chi(q_{2,1})\} \leq \min\{\chi(q_{2,1}), \chi(q_{2,2})\}$ 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 $\beta_2(K_4) \geq 1 - 1/m^2 = 35/36$. Since $\beta_1(K_4) \leq 1 - 1/m = 5/6$, we conclude that $\beta_2(K_4) > \beta_1(K_4)$.

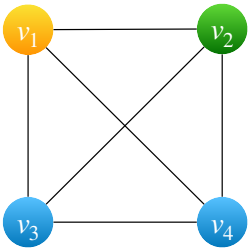

Figure 2: The 4-clique graph colored by χ.

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	$(0,0)$	$(1,1)$	$(2,2)$	$(2,2)$

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

More generally, via similar ideas, for every $m \in \mathbb{N}$ with $m \geq 6$, 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 $S \coloneqq \{\{u, v\} \in (E \setminus \{v_{n-1}, v_n\}) : \chi(u) = \chi(v)\}$ is minimized.

  The rest of the argument is analogous to the case of the 4-clique graph (see Section 4.3 for details).

- 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 $\widetilde{\pi} \coloneqq (\widetilde{\Pi}_t[i^t])_{i\in[l]}$ 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 Section 5: when does parallel repetition for PCPs work?

We comment on the proof of Theorem 2, 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 [Ver96]. 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. (In Appendix A we show that the choice of S is "tight" in the sense that there is a PCP such that, for infinitely many instances not in the language, the soundness error its t-wise parallel repetition is exactly $1/2^{\mathsf{vr}}$ for every t. Hence this lower bound cannot be improved.)

2.3 Section 6: rate of decay of parallel repetition for PCPs

We sketch the proof of Lemma 2: if a PCP is the evaluation of an MIP (Definition 2) 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 $\mathbb{x} \notin L$.

$\boldsymbol{\beta(\mathbb{x}) < 1}$ implies $\boldsymbol{\beta_t(\mathbb{x}) < 1}$. 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 $\beta_t(\mathbb{x}) < 1$.

$\boldsymbol{\beta_t(\mathbb{x}) \leq \beta_{\mathsf{MIP},t}(\mathbb{x})}$. 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 1 and Definition 2, 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 $\beta_t(\mathbb{x}) \leq \beta''_{\mathsf{MIP},t}(\mathbb{x}) = \beta'_{\mathsf{MIP},t}(\mathbb{x}) = \beta_{\mathsf{MIP},t}(\mathbb{x})$.

$\boldsymbol{\beta_t(\mathbb{x}) \geq \beta_{\mathsf{MIP},t}(\mathbb{x})}$. 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 Section 7: parallel repetition for the canonical PCP for CSPs

We discuss the proof of Lemma 1.

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.[5] By our Theorem 2, 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 1). 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 $\widetilde{\Pi}_2[(q_1, q_2)] := (\widetilde{\pi}[q_1], \widetilde{\pi}[q_1])$ 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 $\mathbf{V}_2^{\widetilde{\Pi}_2}(\mathbb{x}; \boldsymbol{\rho}) = 1$.

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 $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho_1) = 1$. 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 Section 8: consistent parallel repetition always works

We discuss the proof of Theorem 3.

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 3, where the repeated PCP verifier additionally checks that any duplicate queries are answered consistently (which means that the repeated PCP

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

verifier no longer is the conjunction of independent "games" as is usually the case in parallel repetition). Theorem 3 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* [Ver96].

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. (We prove the upper bound on $\hat{\beta}_t(\mathbb{x})$ in Section 8.)

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 $\lim_{t\to\infty} \hat{\beta}_t(\mathbb{x}) = 0$.

3 Preliminaries

3.1 Probabilistically checkable proofs

A *probabilistically checkable proof* (PCP) is an information-theoretic proof system where a probabilistic verifier has query access to a proof string produced by a prover.

Definition 3.1 (Completeness)**.** $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ *for a language* L *has* **completeness error** α *if for every instance* $\mathbb{x} \in L$*,*

$$\Pr\left[\mathbf{V}^{\pi}(\mathbb{x}) = 1 \mid \pi \leftarrow \mathbf{P}(\mathbb{x})\right] \geq 1 - \alpha(\mathbb{x}) \ .$$

Definition 3.2 (Soundness)**.** $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ *for a language* L *has* **soundness error** β *if for every instance* $\mathbb{x} \notin L$ *and (unbounded) malicious prover* $\widetilde{\mathbf{P}}$*,*

$$\Pr\left[\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}) = 1 \;\middle|\; \widetilde{\pi} \leftarrow \widetilde{\mathbf{P}}\right] \leq \beta(\mathbb{x}) \ .$$

We consider the following efficiency measures for a PCP.

- *Proof alphabet* Σ*:* the alphabet of the PCP string.
- *Proof length* l*:* the number of alphabet symbols in the PCP string.
- *Query complexity* q*:* the number of queries made by the PCP verifier to the PCP string (a query is an index in $[\mathsf{l}]$, and the answer is the symbol at the corresponding location of the PCP string). We assume without loss of generality that the verifier $\mathbf{V}$ does not make duplicate queries.
- *Verifier randomness complexity* vr*:* the number of random bits used by the PCP verifier.
- *Prover randomness complexity* pr*:* the number of random bits used by the PCP prover.

Any efficiency measure may be a function of the instance $\mathbb{x}$ (e.g., the size of $\mathbb{x}$).

Definition 3.3. *A PCP verifier* $\mathbf{V}$ *is* **non-adaptive** *if the queries it makes to the PCP string depend only on the instance and the randomness. In particular,* $\mathbf{V}$ *can be viewed as a query algorithm* $\mathbf{V}_{\mathrm{q}}$ *and a decision algorithm* $\mathbf{V}_{\mathrm{d}}$ *that work as follows.*

- $\mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho) \to \mathrm{Q}$*: On input an instance* $\mathbb{x}$ *and PCP verifier randomness* ρ*,* $\mathbf{V}_{\mathrm{q}}$ *outputs a query set* $\mathrm{Q} \subseteq [\mathsf{l}]$*.*
- $\mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho, \mathsf{ans}) \to d$*: On input an instance* $\mathbb{x}$*, PCP verifier randomness* ρ*, and query answers* $\mathsf{ans} \in \Sigma^{\mathrm{Q}}$*,* $\mathbf{V}_{\mathrm{d}}$ *outputs the decision* $d \in \{0, 1\}$ *of the PCP verifier.*

Remark 3.4. Consistent with prior literature on parallel repetition, we consider only PCPs with non-adaptive verifiers. This does not trivialize the problem because almost all PCP constructions are non-adaptive. This restriction can be lifted for some of our results (the results in Section 8 extend to adaptive verifiers).

Remark 3.5. The definition of parallel repetition for a (non-adaptive) PCP requires an ordering among the queries in the query set $\mathrm{Q} \subseteq [\mathsf{l}]$ output by the query algorithm $\mathbf{V}_{\mathrm{q}}$ (see Section 3.3). There are two natural orderings: the ordering inherited from some global ordering on $[\mathsf{l}]$ (e.g., the lexicographic order on $[\mathsf{l}]$), or an order specified by the query algorithm $\mathbf{V}_{\mathrm{q}}$ itself (e.g., by outputting Q specified as a list rather than as a set). For simplicity, we assume the ordering of Q is inherited from a global ordering on $[\mathsf{l}]$ unless otherwise specified. Sometimes different orderings result in different soundness errors under parallel repetition.

3.2 Multi-prover interactive proofs

A *one-round multi-prover interactive proof* (MIP) is an information-theoretic proof system where a probabilistic verifier sends a message to each of multiple non-communicating provers, then receives an answer from each prover, and finally outputs a decision bit (denoting whether to accept or reject).

Definition 3.6 (Completeness)**.** $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[\mathsf{k}]}, \mathcal{V})$ *for a language* L *has* **completeness error** α *if for every instance* $\mathbb{x} \in L$*,*

$$\Pr\left[\mathcal{V}(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{k}]}) = 1 \;\middle|\; \begin{array}{l} \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\ \tau \leftarrow \{0,1\}^{\mathsf{pr}} \\ (a_i)_{i\in[\mathsf{k}]} \leftarrow \mathcal{V}(\mathbb{x}, \rho) \\ \forall\, i \in [\mathsf{k}] : b_i \leftarrow \mathcal{P}_i(\mathbb{x}, \tau, a_i) \end{array}\right] \geq 1 - \alpha(\mathbb{x}) \ .$$

Definition 3.7 (Soundness)**.** $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[\mathsf{k}]}, \mathcal{V})$ *for a language* L *has* **soundness error** β *if for every instance* $\mathbb{x} \notin L$ *and (unbounded) malicious provers* $(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{k}]}$*,*

$$\Pr\left[\mathcal{V}(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{k}]}) = 1 \;\middle|\; \begin{array}{l} \rho \leftarrow \{0,1\}^{\mathsf{vr}} \\ \tau \leftarrow \{0,1\}^{\mathsf{pr}} \\ (a_i)_{i\in[\mathsf{k}]} \leftarrow \mathcal{V}(\mathbb{x}, \rho) \\ \forall\, i \in [\mathsf{k}] : b_i \leftarrow \widetilde{\mathcal{P}}_i(\tau, a_i) \end{array}\right] \leq \beta(\mathbb{x}) \ .$$

We consider the following efficiency measures for an MIP.

- *Verifier alphabet* $\Sigma_{\mathcal{V}}$*:* the alphabet for the messages from the verifier to the provers.
- *Provers alphabet* $\Sigma_{\mathcal{P}}$*:* the alphabet for the messages from the provers to the verifier.
- *Verifier randomness complexity* vr*:* the number of random bits used by the verifier.
- *Prover randomness complexity* pr*:* the total number of shared random bits used by the provers.

Any efficiency measure may be a function of the instance $\mathbb{x}$ (e.g., the size of $\mathbb{x}$).

Remark 3.8. Consistent with prior literature on parallel repetition for MIPs, we restrict our attention to one-round MIPs with non-adaptive verifiers (the MIP verifier sends its messages to all provers simultaneously); this is analogous to Remark 3.4 for PCPs. Parallel repetition for MIPs can be defined for any number of rounds (and also for adaptive verifiers), though we do not need this generality for the techniques in this paper. (Separately, parallel repetition for multi-round MIPs has not been studied so far.)

3.3 Parallel repetition for PCPs

The main object of study in this paper is parallel repetition of a PCP, defined below.

Definition 3.9. *Let* $\mathsf{PCP} = (\mathbf{P}, \mathbf{V} = (\mathbf{V}_{\mathrm{q}}, \mathbf{V}_{\mathrm{d}}))$ *be a (non-adaptive) PCP system with proof length* l*, query complexity* q*, and verifier randomness complexity* vr*. The* t**-wise parallel repetition** *of* PCP*, denoted* $(\mathbf{P}_t, \mathbf{V}_t) \coloneqq \otimes[\mathsf{PCP}, t]$*, is the PCP system defined as follows.*

- $\mathbf{P}_t(\mathbb{x})$*:*
  1. *Compute the PCP string* $\pi \leftarrow \mathbf{P}(\mathbb{x})$*.*
  2. *Output* $\Pi \coloneqq ((\pi[q_1], \ldots, \pi[q_t]))_{(q_1,\ldots,q_t)\in[\mathsf{l}]^t}$*.*

- $\mathbf{V}_t^{\Pi}(\mathbb{x})$*:*
  1. *For* $i \in [t]$*:*
     (a) *Sample PCP verifier randomness* $\rho_i \leftarrow \{0,1\}^{\mathsf{vr}}$.
     (b) *Compute the query list of the PCP verifier* $\mathrm{Q}_i \coloneqq \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_i)$.
  2. *For* $i \in [\mathsf{q}]$*:*
     (a) *Set* $\mathbf{Q}_i \coloneqq (\mathrm{Q}_1[i], \ldots, \mathrm{Q}_t[i])$.
     (b) *Query the proof string* $\mathsf{ans}_i \coloneqq \Pi[\mathbf{Q}_i]$.
  3. *Check that* $\wedge_{i\in[t]} \mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho_i, \mathsf{ans}_1[i], \ldots, \mathsf{ans}_{\mathsf{q}}[i])$.

The completeness error becomes $1-(1-\alpha)^t$ but, as discussed in this paper, the soundness error does not behave as expected. The proof alphabet becomes Σ^t and the proof length becomes l^t. The query complexity remains q, albeit over the larger alphabet. The new prover $\mathbf{P}_t$ uses the same amount of randomness as before, while the new verifier $\mathbf{V}_t$ uses $t \cdot \mathsf{vr}$ random bits.

3.4 Parallel repetition for MIPs

Below we define parallel repetition for MIPs, which in this paper plays a role in some analyses.

Definition 3.10. *Let* $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[\mathsf{k}]}, \mathcal{V})$ *be an MIP system with verifier randomness complexity* vr *and provers randomness complexity* pr*. The* t**-wise parallel repetition** *of* MIP*, denoted* $\otimes[\mathsf{MIP}, t]$*, is an MIP system* $((\mathcal{P}_{t,i})_{i\in[\mathsf{k}]}, \mathcal{V}_t)$ *that works as follows. Each MIP prover* $\mathcal{P}_{t,i}$ *receives the instance* $\mathbb{x}$ *and the shared prover randomness* $\boldsymbol{\tau}$ *as inputs. The verifier* $\mathcal{V}_t$ *receives the instance* $\mathbb{x}$ *and the verifier randomness* $\boldsymbol{\rho}$ *as inputs. Then they interact as follows.*

1. $\mathcal{V}_t(\mathbb{x}, \boldsymbol{\rho})$*:*
   (a) *Parse* $\boldsymbol{\rho}$ *as* $(\rho_1, \ldots, \rho_t)$.
   (b) *For every* $i \in [t]$*, compute the* i*-th list of verifier messages* $(a_{i,1}, \ldots, a_{i,\mathsf{k}}) \coloneqq \mathcal{V}(\mathbb{x}, \rho_i)$.
   (c) *For every* $i \in [\mathsf{k}]$*, send* $\boldsymbol{a}_i \coloneqq (a_{1,i}, \ldots, a_{t,i})$ *to the* i*-th prover* $\mathcal{P}_{t,i}$.
2. *For* $i \in [\mathsf{k}]$*,* $\mathcal{P}_{t,i}(\mathbb{x}, \boldsymbol{\tau}, \boldsymbol{a}_i)$*:*
   (a) *Parse* $\boldsymbol{\tau}$ *as* $(\tau_1, \ldots, \tau_t)$.
   (b) *Parse* $\boldsymbol{a}_i$ *as* $(a_{1,i}, \ldots, a_{t,i})$.
   (c) *For every* $j \in [t]$*, compute the MIP prover message* $b_{j,i} \coloneqq \mathcal{P}_i(\mathbb{x}, \tau_j, a_{j,i})$.
   (d) *Send the message* $\boldsymbol{b}_i \coloneqq (b_{1,i}, \ldots, b_{t,i})$ *to the verifier* $\mathcal{V}_t$.
3. $\mathcal{V}_t(\mathbb{x}, \boldsymbol{\rho}, (\boldsymbol{b}_i)_{i\in[\mathsf{k}]})$*: Check that* $\wedge_{i\in[t]} \mathcal{V}(\mathbb{x}, \rho_i, b_{i,1}, \ldots, b_{i,\mathsf{k}})$.

The completeness error becomes $1-(1-\alpha)^t$. The verifier and prover alphabets become $\Sigma_{\mathcal{V}}^t$ and $\Sigma_{\mathcal{P}}^t$, respectively, as each message now contains t original messages. The new provers and the new verifier runs the original ones t times, so the randomness increases by a multiplicative factor of t.

It is straightforward to show that parallel repetition does not increase soundness error, and does not reduce soundness error faster than exponential (in the number of repetitions).

Lemma 3.11. *Let* $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[\mathsf{k}]}, \mathcal{V})$ *be an MIP for a language* L *with soundness error* β*. For* $t \in \mathbb{N}$*, let* β_t *be the soundness error of* $\otimes[\mathsf{MIP}, t] = ((\mathcal{P}_{t,i})_{i\in[\mathsf{k}]}, \mathcal{V}_t)$ *(the* t*-wise parallel repetition of* MIP*). Then for every* $\mathbb{x} \notin L$

$$(\beta(\mathbb{x}))^t \leq \beta_t(\mathbb{x}) \leq \beta(\mathbb{x}) \ .$$

Proof. A valid strategy against $\mathcal{V}_t$ is to run t independent copies of an optimal strategy against $\mathcal{V}$. This shows that $\beta_t(\mathbb{x}) \geq (\beta(\mathbb{x}))^t$, since this approach succeeds if all copies succeed.

On the other hand, any strategy against $\mathcal{V}_t$ yields a strategy against $\mathcal{V}$ that is at least as successful, which shows that $\beta_t(\mathbb{x}) \leq \beta(\mathbb{x})$. Fix $\mathbb{x} \notin L$. Let $(\widetilde{\mathcal{P}}_{t,i})_{i\in[\mathsf{k}]}$ be optimal provers against $\mathcal{V}_t$. Let $(\tau_2, \ldots, \tau_t)$ and $(\rho_2, \ldots, \rho_t)$ be such that the verifier $\mathcal{V}_t$'s acceptance probability is maximized against $(\widetilde{\mathcal{P}}_{t,i})_{i\in[\mathsf{k}]}$. We construct malicious provers $(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{k}]}$ for MIP as follows:

$\widetilde{\mathcal{P}}_i(\tau, a_i)$:
1. For $j = 2, \ldots, t$, run $a_{j,i} \coloneqq \mathcal{V}(\mathbb{x}, \rho_j)$.
2. Run $(b_{1,i}, \ldots, b_{t,i}) \coloneqq \widetilde{\mathcal{P}}_{t,i}((\tau, \tau_2, \ldots, \tau_t), (a_i, a_{2,i}, \ldots, a_{t,i}))$.
3. Output $b_{1,i}$.

Observe that if $\mathcal{V}_t$ accepts then $\mathcal{V}$ accepts. Hence $\beta_t(\mathbb{x}) \leq \beta(\mathbb{x})$. □

If $\beta(\mathbb{x}) = 1$ then, clearly, $\beta_t(\mathbb{x}) = 1$ for every $t \in \mathbb{N}$. However, if $\beta(\mathbb{x}) < 1$ then Verbitsky [Ver96] shows that $\beta_t(\mathbb{x})$ tends to zero as t tends to infinity.[6]

Lemma 3.12 ([Ver96])**.** *Let* MIP *be an MIP for a language* L *with soundness error* β*. Let* β_t *be the soundness error of the* t*-wise parallel repetition of* MIP*. For every* $\mathbb{x} \notin L$*,*

$$\beta(\mathbb{x}) < 1 \Longrightarrow \lim_{t\to\infty} \beta_t(\mathbb{x}) = 0 \ .$$

[6]The discussion in [Ver96] considers the case of two provers but the general case of k provers is a direct extension.

4 Parallel repetition of PCP can increase soundness error

We prove that parallel repetition of a PCP can increase soundness error. We give a minimal example of a 2-query PCP for an NP-complete language.

Theorem 4.1. *There exists a 2-query PCP system* $\mathsf{PCP} \coloneqq (\mathbf{P}, \mathbf{V})$ *for an* NP*-complete language* L *with soundness error* $\beta < 1$ *(i.e.,* $\beta(\mathbb{x}) < 1$ *for every* $\mathbb{x} \notin L$*) such that the following holds. Let* β_t *be the soundness error of* $\otimes[\mathsf{PCP}, t]$ *(the* t*-wise parallel repetition of* PCP *as in Definition 3.9). For every* $\mathbb{x} \notin L$*,*

$$\lim_{t\to\infty} \beta_t(\mathbb{x}) = 1 \ .$$

Moreover, for infinitely many instances $\mathbb{x} \notin L$*,*

$$\beta_{t+1}(\mathbb{x}) > \beta_t(\mathbb{x}) \ .$$

In Section 4.1, we define the NP-complete language and a canonical PCP for this language. In Section 4.2, we show that the soundness error of the parallel repetition of the canonical PCP tends to 1 for every instance not in the language. In Section 4.3, we argue that for infinitely many instances not in the language, the soundness error of parallel repetition strictly increases with the number of repetitions.

4.1 PCP for graph 3-coloring

A 3-coloring of a graph G is a labeling of the vertices with 3 different colors such that no adjacent vertices have the same color. Graph 3-coloring 3COL is the NP-complete language where

$$\mathrm{3COL} \coloneqq \{G = (V, E) : \exists\, \chi \in \{0,1,2\}^V \text{ where } \forall\, e = \{u, v\} \in E, \chi(u) \neq \chi(v)\} \ .$$

Construction 4.2 (PCP for 3COL)**.** We define a PCP $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ for 3COL.

- $\mathbf{P}(G)$:
  1. Parse G as (V, E).
  2. Find a 3-coloring χ for G with the maximum number of edges $e = \{u, v\} \in E$ such that $\chi(u) \neq \chi(v)$.
  3. Output $\pi \coloneqq (\chi(v))_{v\in V}$ (assume that V is ordered lexicographically).
- $\mathbf{V}^{\pi}(G)$:
  1. Parse G as (V, E).
  2. Sample a random edge $e = \{u, v\} \in E$ (an edge is written in lexicographic order of the vertices).
  3. Make two queries to π: $c_u \coloneqq \pi[u]$ and $c_v \coloneqq \pi[v]$.
  4. Accept if and only if $c_u \neq c_v$.

The proof alphabet is $\{0, 1, 2\}$, the proof length is $|V|$, and the query complexity is 2. The PCP verifier uses $\log_2 |E|$ bits of randomness to sample a random edge (which defines the two queries).

If $G \in \mathrm{3COL}$, the PCP prover outputs a 3-coloring $\pi \colon V \to \{0, 1, 2\}$, which convinces the PCP verifier with probability 1. If $G \notin \mathrm{3COL}$ then, for every PCP string $\widetilde{\pi} \colon V \to \{0, 1, 2\}$, the PCP verifier accepts with probability at most $\frac{|E|-1}{|E|}$ (because for every coloring of G there is at least one edge whose vertices share the same color). More precisely, the acceptance probability is at most $\mathsf{val}(G)$, the maximum fraction of valid edges across any coloring of G.

4.2 The soundness error of parallel repetition tends to 1

Lemma 4.3. *Let* $\mathsf{PCP} \coloneqq (\mathbf{P}, \mathbf{V})$ *be the PCP system for* 3COL *in Construction 4.2. For every* $t \in \mathbb{N}$*, let* β_t *be the soundness error of* $\otimes[\mathsf{PCP}, t]$*. For every graph* $G = (V, E) \notin$ 3COL*,*

$$\beta_t(G) \geq 1 - \left(\frac{|E| - 1}{|E|}\right)^t \ .$$

Proof. Fix a graph $G = (V, E) \notin$ 3COL. Consider the following malicious prover $\widetilde{\mathbf{P}}_t$ for the t-wise parallel repetition of PCP.

$\widetilde{\mathbf{P}}_t$:
1. Initialize $\widetilde{\Pi}_t \coloneqq (0^t)^{[|V|]^t}$.
2. Let $u \in V$ be the smallest non-isolated vertex in G (in lexicographic order of V).
3. For every $(q_1, \ldots, q_t) \in [|V|]^t$:
   (a) If $\min(q_1, \ldots, q_t) = u$, set $\widetilde{\Pi}_t[(q_1, \ldots, q_t)] \coloneqq 0^t$.
   (b) Otherwise, set $\widetilde{\Pi}_t[(q_1, \ldots, q_t)] \coloneqq 1^t$.
4. Output $\widetilde{\Pi}_t$.

Let $\widetilde{\Pi}_t$ be the output of $\widetilde{\mathbf{P}}_t$. Let u be the smallest non-isolated vertex in G. We observe that the parallel repeated verifier $\mathbf{V}_t$ rejects $\widetilde{\Pi}_t$ if and only if $u \notin \mathbf{Q} \in [|V|]^t$ for every query $\mathbf{Q}$ of $\mathbf{V}_t$, which happens with probability at most $\left(\frac{|E|-1}{|E|}\right)^t$ because u is adjacent to at least one edge. Hence,

$$\beta_t(G) \geq \Pr\left[\mathbf{V}_t^{\widetilde{\Pi}_t}(G) = 1\right] \geq 1 - \left(\frac{|E| - 1}{|E|}\right)^t \ .$$

□

4.3 Parallel repetition strictly increases soundness error

In Lemma 4.4, we prove that t-wise parallel repetition of the PCP for 3COL increases soundness error. Note that below we only consider graphs with $|E| \geq 6$ because all graphs with fewer than 6 edges are 3-colorable.

Lemma 4.4. *Let* $\mathsf{PCP} \coloneqq (\mathbf{P}, \mathbf{V})$ *be the PCP system for* 3COL *in Construction 4.2. Let* β_t *be the soundness error for* t*-wise parallel repetition* $\otimes[\mathsf{PCP}, t] \coloneqq (\mathbf{P}_t, \mathbf{V}_t)$ *of* PCP*. For every* $m \in \mathbb{N}$ *with* $m \geq 6$ *there exists an instance* $G = (V, E) \notin$ 3COL *with* $|E| = m$ *such that*

$$\beta_t(G) = 1 - \frac{1}{m^t} \ .$$

Proof. Fix $m \in \mathbb{N}$ such that $m \geq 6$ (all graphs with less than 6 edges are 3-colorable).

Almost 3-colorable graphs. We say a graph $G = (V = \{v_1, \ldots, v_n\}, E)$ (where $n \coloneqq |V|$) is ***almost*** 3***-colorable*** if the followings hold:

- $|E| = m$;
- $G \notin$ 3COL;
- $G' \coloneqq (V, E \setminus \{v_{n-1}, v_n\}) \in$ 3COL.

We note that for every graph G that satisfies the three conditions above, there exists an "ordered" 3-coloring χ for G' (possibly after renaming the vertices of G') such that for every pair of vertices $u, v \in V$ with $u < v$, $\chi(u) \leq \chi(v)$.

We construct an almost 3-colorable graph $G = (V = \{v_i\}_{i\in[n]}, E)$ with $|E| = m$. Moreover, we exhibit an "ordered" 3-coloring χ for G'. For simplicity, we say that $v_i < v_j$ if $i < j$.

Let $n = 2m - 8$. We define the edge set as follows:

$$E \coloneqq \left\{\{v_i, v_{i+n/2}\} : i \in \left[\frac{n}{2} - 2\right]\right\} \cup \left\{\{v_i, v_j\} : i, j \in \left\{\frac{n}{2} - 1, \frac{n}{2}, n-1, n\right\}\right\} \quad .$$

Note that G has $m-5$ connected components: $\{v_{\frac{n}{2}-1}, v_{\frac{n}{2}}, v_{n-1}, v_n\}$ forms a 4-clique, and all other connected components are formed by 2 vertices. Therefore, G is not 3-colorable because the 4-clique is not 3-colorable. Note that $|E| = 6 + (m-6) = m$, where 6 is the number of edges in the 4-clique and $m-6$ is the number of components with a single edge. Next we show that $G' = (V, m \setminus \{\{v_{n-1}, v_n\}\}) \in 3\mathrm{COL}$ and a 3-coloring χ as required above exists. We define $\chi \in \{0,1,2\}^V$ for every $v_i \in V$:

$$\chi(v_i) \coloneqq \begin{cases} 0 & \text{if } 1 \leq i \leq n/2 - 1, \\ 1 & \text{if } n/2 \leq i \leq n-2, \\ 2 & \text{otherwise.} \end{cases}$$

We observe that χ is indeed a 3-coloring for G', and for every $u, v \in V$ such that $u < v$, $\chi(u) \leq \chi(v)$.

Lower bound: $\beta_t \geq 1 - 1/m^t$. Fix an almost 3-colorable graph $G \notin 3\mathrm{COL}$. We consider the following malicious prover $\widetilde{\mathbf{P}}_t$ for the t-wise parallel repetition.

$\widetilde{\mathbf{P}}_t$:
1. Parse the instance G as (V, E).
2. Let $n \coloneqq |V|$ and $m \coloneqq |E|$.
3. Parse V as $\{v_1, \ldots, v_n\}$ in which the indices of the vertices are determined by the lexicographic order of V.
4. Let $\chi \in \{0,1,2\}^V$ be an labeling that satisfies the following:
   (a) For every $u, v \in V$ such that $u < v$, $\chi(u) \leq \chi(v)$.
   (b) The size of the set S is minimized, where $S \coloneqq \{\{u, v\} \in (E \setminus \{v_{n-1}, v_n\}) : \chi(u) = \chi(v)\}$.
5. Output $\widetilde{\Pi}_t \coloneqq \left((\min\{\chi(q_1), \ldots, \chi(q_t)\})^t\right)_{(q_1,\ldots,q_t)\in V^t}$.

Let $\mathbf{Q}_1$ and $\mathbf{Q}_2$ be the two queries made by $\mathbf{V}_t$. By definition of $\mathbf{V}$ (see Construction 4.2), $\mathbf{Q}_1[i] < \mathbf{Q}_2[i]$ and $\{\mathbf{Q}_1[i], \mathbf{Q}_2[i]\} \in E$ for every $i \in [t]$. Let χ be defined as in $\widetilde{\mathbf{P}}_t$. Let $m_1 \coloneqq \min\{\chi(\mathbf{Q}_1[i])\}_{i\in[t]}$ and $m_2 \coloneqq \min\{\chi(\mathbf{Q}_2[i])\}_{i\in[t]}$. Recall that for every edge $\{u, v\} \neq \{v_{n-1}, v_n\}$ such that $u < v$, $\chi(u) < \chi(v)$. Therefore, $m_1 \leq m_2$. We consider the following two cases:

- *Case 1:* $m_1 < m_2$. By definition of $\widetilde{\Pi}_t$, we know that $\mathsf{ans}_1 = (m_1)^t$ and $\mathsf{ans}_2 = (m_2)^t$. Since $m_1 \neq m_2$, $\mathbf{V}_t$ accept.

- *Case 2:* $m_1 = m_2$. In this case $\mathbf{V}_t$ rejects. However, this case can only happen when $\mathbf{Q}_1 = (v_{n-1})^t$ and $\mathbf{Q}_2 = (v_n)^t$ ($\chi(u) < \chi(v)$ for every edge $\{u, v\} \neq \{v_{n-1}, v_n\}$ if $u < v$), which happens with probability $1/m^t$.

We conclude that $\beta_2(G) \geq 1 - \frac{1}{m^2}$.

Upper bound: $\beta_t \leq 1 - 1/m^t$. Fix an arbitrary graph $G \notin$ 3COL. It suffices to show that $\beta_t(G) < 1$: the number of random choices of $\mathbf{V}_t$ is m^t, hence $\beta_t(G) < 1$ implies $\beta_t(G) \leq 1 - 1/m^t$.

We show that if $\beta_t(G) = 1$, then $\beta(G) = 1$, which contradicts to our assumption that PCP has non-trivial soundness.

Assume that $\beta_t(G) = 1$. Let $\widetilde{\Pi}_t$ be the PCP string for the t-wise repetition of PCP such that $\mathbf{V}_t^{\widetilde{\Pi}_t}(G)$ always accepts. We construct a malicious prover $\widetilde{\mathbf{P}}$ for PCP as follows:

$\widetilde{\mathbf{P}}$:

1. Parse G as (V, E).
2. Output $\widetilde{\pi} := \left(\widetilde{\Pi}_t[(v^t)][1]\right)_{v \in V}$.

Let $\widetilde{\pi}$ be the output of $\widetilde{\mathbf{P}}$. For every $\rho \in \{0,1\}^{\mathsf{vr}}$, we know that $\mathbf{V}_t^{\widetilde{\Pi}_t}(G; \rho^t) = 1$, which implies that $\mathbf{V}^{\widetilde{\pi}}(G; \rho) = 1$ as desired. Therefore, $\beta(G) = 1$. □

5 A characterization result

We prove a result that characterizes when parallel repetition of a PCP "works", that is, causes the soundness error to tend to zero as the number of repetitions tends to infinity. The characterization relies on whether a corresponding MIP, which we refer to as the *MIP projection of the PCP*, has non-trivial soundness error. Informally, the MIP has one prover per PCP query, and each MIP prover answers a single query *without any consistency checks performed by the MIP verifier*.

Definition 5.1. *Let* $\mathsf{PCP} = (\mathbf{P}_{\mathsf{PCP}}, \mathbf{V}_{\mathsf{PCP}} = (\mathbf{V}_{\mathrm{q}}, \mathbf{V}_{\mathrm{d}}))$ *be a (non-adaptive) PCP system with proof alphabet* Σ*, proof length* l*, query complexity* q*, and verifier randomness* vr*. The* **MIP projection of** PCP *is an MIP with* q *provers, denoted* $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[\mathsf{q}]}, \mathcal{V})$*, that works as follows. Each MIP prover* $\mathcal{P}_i$ *receives the instance* $\mathbb{x}$ *and the shared randomness* τ *as inputs. The MIP verifier* $\mathcal{V}$ *receives the instance* $\mathbb{x}$ *and the verifier randomness* ρ *as inputs. Then MIP verifier and MIP provers interact as follows.*

1. $\mathcal{V}(\mathbb{x}, \rho)$*:*
   (a) *Compute the query list of the* PCP *verifier* $\mathrm{Q} \coloneqq \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho)$.
   (b) *Parse the query list* Q *as a tuple* $(\mathrm{q}_i)_{i\in[\mathsf{q}]}$.
   (c) *For every* $i \in [\mathsf{q}]$*, send* q_i *to the* i*-th prover* $\mathcal{P}_i$.
2. *For* $i \in [\mathsf{q}]$*,* $\mathcal{P}_i(\mathbb{x}, \tau, \mathrm{q}_i)$*:*
   (a) *Compute the PCP string* $\pi \coloneqq \mathbf{P}_{\mathsf{PCP}}(\mathbb{x}; \tau)$.
   (b) *Send the message* $b_i \coloneqq \pi[\mathrm{q}_i]$ *to the MIP verifier* $\mathcal{V}$.
3. $\mathcal{V}(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{q}]})$*: Check that* $\mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{q}]}) = 1$.

The completeness error of the MIP projection is the same as that of the PCP. The MIP prover alphabet is Σ and MIP verifier alphabet is $[\mathsf{l}]$. The MIP verifier uses vr random bits.

We show that parallel repetition of a given PCP yields arbitrarily small soundness error if and only if the corresponding MIP projection has non-trivial soundness error.

Theorem 5.2. *Let* PCP *be a (non-adaptive) PCP system for a language* L*; denote by* β_t *the soundness error of* $\otimes[\mathsf{PCP}, t]$*. Let* MIP *be the MIP projection of* PCP*, and let* β_{MIP} *be its soundness error. For every* $\mathbb{x} \notin L$*,*

$$\lim_{t\to\infty} \beta_t(\mathbb{x}) = 0 \iff \beta_{\mathsf{MIP}}(\mathbb{x}) < 1 \ .$$

In fact, if $\beta_{\mathsf{MIP}}(\mathbb{x}) = 1$ *then* $\lim_{t\to\infty} \beta_t(\mathbb{x}) \in [\frac{1}{2^{\mathsf{vr}}}, 1]$.

In Appendix A we show via examples that the above characterization is essentially tight.

5.1 Proof of Theorem 5.2

We show the two directions separately in Lemmas 5.3 and 5.5 below.

Lemma 5.3. *If* $\beta_{\mathsf{MIP}}(\mathbb{x}) = 1$ *then* $\lim_{t\to\infty} \beta_t(\mathbb{x}) \geq \frac{1}{2^{\mathsf{vr}}} > 0$.

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

For every $t \in \mathbb{N}$ and $\rho^* \in \{0,1\}^{\mathsf{vr}}$, define the set

$$W_{t,\rho^*} \coloneqq \{\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t) \in (\{0,1\}^{\mathsf{vr}})^t : \rho_t = \rho^*\} \ .$$

We construct a malicious prover $\widetilde{\mathbf{P}}_t$ that outputs a PCP string $\widetilde{\Pi}$ such that, for every $\boldsymbol{\rho} \in W_{t,\rho^*}$, $\mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho}) = 1$. This suffices because we can then conclude that

$$\lim_{t\to\infty} \beta_t(\mathbb{x}) \geq \lim_{t\to\infty} \frac{|W_{t,\rho^*}|}{|(\{0,1\}^{\mathsf{vr}})^t|} = \lim_{t\to\infty} \frac{(2^{\mathsf{vr}})^{t-1}}{(2^{\mathsf{vr}})^t} = \frac{1}{2^{\mathsf{vr}}} > 0 \ .$$

We are left to construct and analyze $\widetilde{\mathbf{P}}_t$. Let $(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{q}]}$ be (deterministic) malicious provers that make the MIP verifier always accept the instance $\mathbb{x}$ (they exist since $\beta_{\mathsf{MIP}}(\mathbb{x}) = 1$).

$\widetilde{\mathbf{P}}_t$:

1. Initialize $\widetilde{\Pi} \coloneqq \left(\sigma^t\right)^{\mathsf{l}^t}$ for some arbitrary $\sigma \in \Sigma$.
2. For every $\boldsymbol{\rho} \in W_{t,\rho^*}$ and $i \in [\mathsf{q}]$:
   (a) Parse $\boldsymbol{\rho}$ as $(\rho_1, \ldots, \rho_t)$.
   (b) Initialize $\mathsf{ans}_i \coloneqq \sigma^t$.
   (c) For every $j \in [t]$:
      i. Compute the query list of the j-th PCP verifier $\mathrm{Q}_j \coloneqq \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_j)$.
      ii. Compute the i-th MIP prover answer $\mathsf{ans}_i[j] \coloneqq \widetilde{\mathcal{P}}_i(\mathrm{Q}_j[i])$.
   (d) Set $\widetilde{\Pi}\left[(\mathrm{Q}_1[i], \ldots, \mathrm{Q}_t[i])\right] \coloneqq \mathsf{ans}_i$.
3. Output the PCP string $\widetilde{\Pi}$.

Intuitively, the PCP prover $\widetilde{\mathbf{P}}_t$ fills in the locations corresponding to the queried locations for every randomness $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t)$ in W_{t,ρ^*}, based on the answers provided by the MIP malicious provers $(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{q}]}$. No position in $\widetilde{\Pi}$ is updated more than once with different values, due to the following claim.

Claim 5.4. *For every (not necessarily distinct) $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t), \boldsymbol{\rho}' = (\rho'_1, \ldots, \rho'_t) \in W_{t,\rho^*}$ and distinct $i, j \in [\mathsf{q}]$, $(\mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_1)[i], \ldots, \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_t)[i]) \neq (\mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho'_1)[j], \ldots, \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho'_t)[j])$.*

Proof. The PCP verifier always makes q distinct queries. After fixing the randomness to ρ^* for the last repetition, we deduce that, for every distinct $i, j \in [\mathsf{q}]$, $\mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho^*)[i] \neq \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho^*)[j]$, which in turn implies that $(\mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_1)[i], \ldots, \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_t)[i]) \neq (\mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho'_1)[j], \ldots, \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho'_t)[j])$. □

Next we argue that the malicious PCP string $\widetilde{\Pi}$ output by $\widetilde{\mathbf{P}}_t$ convinces the repeated PCP verifier for every verifier randomness choice in W_{t,ρ^*}: fixing $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t) \in W_{t,\rho^*}$, we argue that $\mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho}) = 1$. Given ρ_i, we know that the MIP projection verifier queries $\mathrm{Q}_i \coloneqq \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_i)$ by definition. Moreover, since by assumption the malicious MIP projection provers always make the MIP verifier accept,

$$\mathbf{V}_{\mathrm{d}}\left(\mathbb{x}, \rho_i, \left(\widetilde{\mathcal{P}}_c\left(\mathrm{Q}_i[c]\right)\right)_{c\in[\mathsf{q}]}\right) = 1 \ .$$

Therefore, by Definition 3.9, the t-wise parallel repetition verifier $\mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho})$ accepts for every $\boldsymbol{\rho} \in W_{t,\rho^*}$ as desired. □

Lemma 5.5. *If $\beta_{\mathsf{MIP}}(\mathbb{x}) < 1$ then $\lim_{t\to\infty} \beta_t(\mathbb{x}) = 0$.*

Proof. First we prove that the soundness error of a PCP is upper bounded by that of its MIP projection.

Claim 5.6. *Let PCP be a PCP for a language L with soundness error β_{PCP}. Let MIP be the MIP projection of PCP, and let β_{MIP} be the soundness error of MIP. For every $\mathbb{x} \notin L$,*

$$\beta_{\mathsf{PCP}}(\mathbb{x}) \leq \beta_{\mathsf{MIP}}(\mathbb{x}) \ .$$

Proof. Let $\widetilde{\mathbf{P}}$ be a malicious prover against the PCP verifier $\mathbf{V}$ of PCP. Consider the malicious MIP provers $(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{q}]}$ against the MIP verifier $\mathcal{V}$ of MIP that work as follows: for every $i \in [\mathsf{q}]$, $\widetilde{\mathcal{P}}_i(\tau, a_i)$ computes $\widetilde{\pi} \leftarrow \widetilde{\mathbf{P}}(\tau)$ and answers with $\widetilde{\pi}[a_i]$.

For every verifier randomness ρ, if $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho) = 1$ then

$$\mathcal{V}(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{q}]}) = \mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{q}]}) = \mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho, (\widetilde{\pi}[a_i])_{i\in[\mathsf{q}]}) = 1 \ .$$

We conclude that $\beta_{\mathsf{PCP}}(\mathbb{x}) \leq \beta_{\mathsf{MIP}}(\mathbb{x})$. □

Next we show that performing an MIP projection and performing parallel repetition "commute".

Claim 5.7. *Let* $\mathsf{PCP} = (\mathbf{P}, \mathbf{V} = (\mathbf{V}_{\mathrm{q}}, \mathbf{V}_{\mathrm{d}}))$ *be a (non-adaptive) PCP. For every* $t \in \mathbb{N}$*, consider:*
- $\mathsf{MIP}_{t,1}$*, the MIP projection of the* t*-wise parallel repetition of* PCP*; and*
- $\mathsf{MIP}_{t,2}$*, the* t*-wise parallel repetition of the MIP projection of* PCP.

Then $\mathsf{MIP}_{t,1} = \mathsf{MIP}_{t,2}$.

Proof. Let $\mathsf{MIP}_{t,1} = ((\mathcal{P}_{1,i})_{i\in[\mathsf{q}]}, \mathcal{V}_1)$ and $\mathsf{MIP}_{t,2} = ((\mathcal{P}_{2,i})_{i\in[\mathsf{q}]}, \mathcal{V}_2)$. We show that the provers and verifier in $\mathsf{MIP}_{t,1}$ are the same as in $\mathsf{MIP}_{t,2}$ (have the same input-output behavior).

Fix an instance $\mathbb{x}$, a prover randomness $\boldsymbol{\tau} \coloneqq (\tau_1, \ldots, \tau_t) \in (\{0,1\}^{\mathsf{pr}})^t$, and a verifier randomness $\boldsymbol{\rho} \coloneqq (\rho_1, \ldots, \rho_t) \in (\{0,1\}^{\mathsf{vr}})^t$. Let $(\mathbf{P}_t, \mathbf{V}_t = (\mathbf{V}_{\mathrm{q},t}, \mathbf{V}_{\mathrm{d},t}))$ be t-wise parallel repetition of PCP and $((\mathcal{P}_i)_{i\in[\mathsf{q}]}, \mathcal{V})$ be the MIP projection of PCP.

- *Verifier messages.* For every $i \in [\mathsf{q}]$, by Definition 5.1, $\mathcal{V}_1(\mathbb{x}, \boldsymbol{\rho})$ sends $\mathbf{V}_{\mathrm{q},t}(\mathbb{x}, \boldsymbol{\rho})[i]$ to the i-th prover where $\mathbf{V}_{\mathrm{q},t}(\mathbb{x}, \boldsymbol{\rho})[i] = (\mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_1)[i], \ldots, \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_t)[i])$ by Definition 3.9. For every $i \in [\mathsf{q}]$, by Definition 3.10, $\mathcal{V}_2(\mathbb{x}, \boldsymbol{\rho})$ sends $(\mathcal{V}(\mathbb{x}, \rho_1)[i], \ldots, \mathcal{V}(\mathbb{x}, \rho_t)[i])$ to the i-th prover where $\mathcal{V}(\mathbb{x}, \rho_j)[i] = \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_j)[i]$ for every $j \in [t]$ by Definition 5.1. Therefore, we observe that $\mathcal{V}_1(\mathbb{x}, \boldsymbol{\rho}) = \mathcal{V}_2(\mathbb{x}, \boldsymbol{\rho}) = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{\mathsf{q}}) \in ([\mathsf{l}]^t)^{\mathsf{q}}$ in which for every $i \in [\mathsf{q}]$ and $j \in [t]$,
$$\boldsymbol{a}_i[j] = \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_j)[i] \ .$$
- *Prover answers.* Let $\pi \leftarrow \mathbf{P}(\mathbb{x})$. For every $i \in [\mathsf{q}]$, by Definition 5.1, $\mathcal{P}_{1,i}(\mathbb{x}, \boldsymbol{\tau}, \boldsymbol{a}_i)$ sends $\Pi[\boldsymbol{a}_i]$ to the verifier where $\Pi = ((\pi[q_1], \ldots, \pi[q_t]))_{(q_1,\ldots,q_t)\in[\mathsf{l}]^t}$ by Definition 3.9. For every $i \in [\mathsf{q}]$, by Definition 3.10, $\mathcal{P}_{2,i}(\mathbb{x}, \boldsymbol{\tau}, \boldsymbol{a}_i)$ sends $(\mathcal{P}_i(\mathbb{x}, \tau_1, \boldsymbol{a}_i[1]), \ldots, \mathcal{P}_i(\mathbb{x}, \tau_t, \boldsymbol{a}_i[t])) \in [\mathsf{l}]^t$ to the verifier where $\mathcal{P}_i(\mathbb{x}, \tau_j, \boldsymbol{a}_i[j]) = \pi[\boldsymbol{a}_i[j]]$ for every $j \in [t]$ by Definition 5.1. We conclude that, for every $i \in [\mathsf{q}]$, $\mathcal{P}_{1,i}(\mathbb{x}, \boldsymbol{\tau}, \boldsymbol{a}_i) = \mathcal{P}_{2,i}(\mathbb{x}, \boldsymbol{\tau}, \boldsymbol{a}_i) = \boldsymbol{b}_i \in \Sigma^t$ where $\boldsymbol{b}_i = (\pi[\boldsymbol{a}_i[1]], \ldots, \pi[\boldsymbol{a}_i[t]])$.
- *Verifier decision.* For every $i \in [\mathsf{q}]$, by Definition 5.1, $\mathcal{V}_1(\mathbb{x}, \boldsymbol{\rho}, (\boldsymbol{b}_i)_{i\in[\mathsf{q}]})$ checks $\mathbf{V}_{\mathrm{d},t}(\mathbb{x}, \boldsymbol{\rho}, (\boldsymbol{b}_i)_{i\in[\mathsf{q}]})$ where $\mathbf{V}_{\mathrm{d},t}(\mathbb{x}, \boldsymbol{\rho}, (\boldsymbol{b}_i)_{i\in[\mathsf{q}]}) = \bigwedge_{j\in[t]} \mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho_j, (\boldsymbol{b}_i[j])_{i\in[\mathsf{q}]})$ by Definition 3.9. For every $i \in [\mathsf{q}]$, by Definition 3.10, $\mathcal{V}_2(\mathbb{x}, \boldsymbol{\rho}, (\boldsymbol{b}_i)_{i\in[\mathsf{q}]})$ checks $\bigwedge_{j\in[t]} \mathcal{V}(\mathbb{x}, \rho_j, (\boldsymbol{b}_i[j])_{i\in[\mathsf{q}]})$ where $\mathcal{V}(\mathbb{x}, \rho_j, (\boldsymbol{b}_i[j])_{i\in[\mathsf{q}]}) = \mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho_j, (\boldsymbol{b}_i[j])_{i\in[\mathsf{q}]})$ by Definition 5.1. Therefore, $\mathcal{V}_1(\mathbb{x}, \boldsymbol{\rho}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\mathsf{q}}) = \mathcal{V}_2(\mathbb{x}, \boldsymbol{\rho}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\mathsf{q}})$ because
$$\mathcal{V}_1(\mathbb{x}, \boldsymbol{\rho}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\mathsf{q}}) = \mathcal{V}_2(\mathbb{x}, \boldsymbol{\rho}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\mathsf{q}}) = \bigwedge_{i\in[t]} \mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho_i, \boldsymbol{b}_1[i], \ldots, \boldsymbol{b}_{\mathsf{q}}[i]) = 1 \ .$$

□

We conclude the proof of the lemma by combining Claim 5.6 and Claim 5.7. Specifically, define $\mathsf{MIP}_{t,1}$ and $\mathsf{MIP}_{t,2}$ as in Claim 5.7. Let $\beta_{\mathsf{MIP}_{t,1}}$ and $\beta_{\mathsf{MIP}_{t,2}}$ be their soundness errors, respectively. Then, for every instance $\mathbb{x} \notin L$,
$$\lim_{t\to\infty} \beta_t(\mathbb{x}) \leq \lim_{t\to\infty} \beta_{\mathsf{MIP}_{t,1}}(\mathbb{x}) = \lim_{t\to\infty} \beta_{\mathsf{MIP}_{t,2}}(\mathbb{x}) = 0 \ .$$

The last equality follows from Verbitsky's result (Lemma 3.12) which shows that the soundness error of the parallel repetition of an MIP tends to 0 as the number of repetitions tends to infinity, provided that the MIP has non-trivial soundness error (which is the case here since $\beta_{\mathsf{MIP}}(\mathbb{x}) < 1$ for every $\mathbb{x} \notin L$). We summarize the proof in Figure 3 for clarity. □

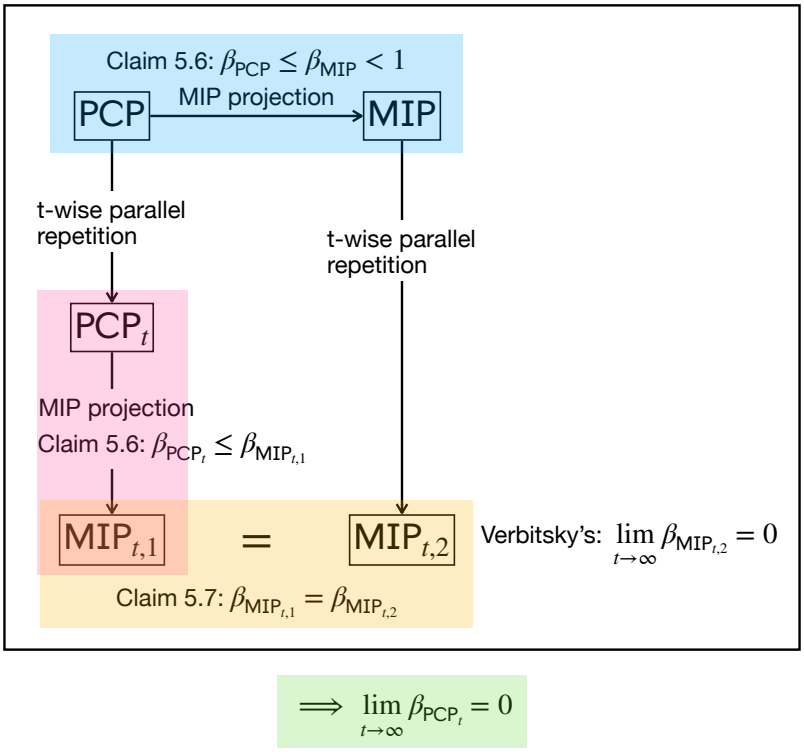

Figure 3: Using Claim 5.6 and Claim 5.7 to conclude Lemma 5.5.

Remark 5.8. In Section 4 we describe a 2-query PCP for 3COL whose parallel repetition has a soundness error that tends to 1 (on every instance not in 3COL). Theorem 5.2 gives an additional perspective: the soundness error does not tend to 0 *because the MIP projection of the PCP has soundness error* 1.

The MIP projection of the PCP for 3COL has an MIP verifier that works as follows: sample a random edge $\{u, v\}$ of the given graph G (assume without loss of generality $u < v$ lexicographically) and then send u to the first prover and v to the second prover; then receive answers ans_1 and ans_2, and check that $\mathsf{ans}_1 \neq \mathsf{ans}_2$. Evidently, $\beta_{\mathsf{MIP}}(G) = 1$ because the first MIP prover can always answer $\mathsf{ans}_1 = 0$ and the second MIP prover can always answer $\mathsf{ans}_2 = 1$ (as these answers always convince the MIP verifier).

Remark 5.9. Claim 5.6 implies that if a PCP has soundness error 1 then its MIP projection has soundness error 1. Below we show that *if a PCP has soundness error* 0 *then its MIP projection has soundness error* 0. There are natural examples of such PCPs: any NP verifier can be viewed as a PCP verifier that reads the entire PCP string, and has soundness error 0; hence its MIP projection has (unsurprisingly) soundness error 0.

Let $\mathsf{PCP} = (\mathbf{P}, \mathbf{V} = (\mathbf{V}_{\mathrm{q}}, \mathbf{V}_{\mathrm{d}}))$ be a PCP for a language L and let $\mathbb{x} \notin L$ be such that the soundness error β on $\mathbb{x}$ is 0. Suppose by way of contradiction that the MIP projection $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[\mathsf{q}]}, \mathcal{V})$ of this PCP has soundness error $\beta_{\mathsf{MIP}}(\mathbb{x}) > 0$.

Let $(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{q}]}$ be optimal (deterministic) malicious provers for the MIP projection for the instance $\mathbb{x}$. We construct a malicious PCP prover $\widetilde{\mathbf{P}}$ as follows.

$\widetilde{\mathbf{P}}$:

1. Set $\widetilde{\pi} \coloneqq \sigma^{\mathsf{l}}$ for an arbitrary $\sigma \in \Sigma$.
2. For every $\rho \in \{0,1\}^{\mathsf{vr}}$, if $\langle (\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{q}]}, \mathcal{V}(\mathbb{x}, \rho)\rangle = 1$ then set $\rho^* \coloneqq \rho$ and exit the loop.
3. Deduce the query list $\mathrm{Q} \coloneqq \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho^*)$.
4. For every $i \in [\mathsf{q}]$:
   (a) Deduce the i-th prover's answer: $\mathsf{ans}_i \coloneqq \widetilde{\mathcal{P}}_i(\mathrm{Q}[i])$
   (b) Set $\widetilde{\pi}[\mathrm{Q}[i]] \coloneqq \mathsf{ans}_i$.
5. Output $(\rho^*, \widetilde{\pi})$.

Let $(\rho^*, \widetilde{\pi})$ be the output of $\widetilde{\mathbf{P}}$. By construction and the fact that $\mathbf{V}_{\mathrm{q}}$ outputs distinct queries, we know that $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho^*) = 1$, which is contradictory to our assumption that the PCP has soundness error 0. Therefore, we conclude that if a PCP has soundness error 0, then its MIP projection has soundness error 0.

Remark 5.10. Theorem 5.2 shows that for some PCPs, the soundness error after parallel repetition approaches 1. However, we want to emphasize that the limit approaching 1 does not mean that the soundness error of the parallel repetition is 1 for some $t \in \mathbb{N}$. In fact, consider $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ for a language L with soundness error β. Let $\otimes[\mathsf{PCP}, t]$ be its parallel repetition with soundness error β_t. For every instance $\mathbb{x} \notin L$, if $\beta(\mathbb{x}) < 1$ then $\beta_t(\mathbb{x}) < 1$.

Similar to the argument in the proof of Lemma 4.4, we show that $\beta_t(\mathbb{x}) = 1$ implies $\beta(\mathbb{x}) = 1$. Let $\widetilde{\Pi}_t$ be the PCP string for $\otimes[\mathsf{PCP}, t]$ such that $\mathbf{V}_t^{\widetilde{\Pi}_t}(\mathbb{x})$ always accepts. We construct a malicious prover $\widetilde{\mathbf{P}}$ for PCP as follows:

$\widetilde{\mathbf{P}}$: Output $\widetilde{\pi} \coloneqq \left(\widetilde{\Pi}_t[(q^t)][1]\right)_{q\in\mathsf{l}}$.

Let $\widetilde{\pi}$ be the output of $\widetilde{\mathbf{P}}$. For every $\rho \in \{0,1\}^{\mathsf{vr}}$, we know that $\mathbf{V}_t^{\widetilde{\Pi}_t}(\mathbb{x}; \rho^t) = 1$, which implies that $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho) = 1$ as desired. Therefore, $\beta(G) = 1$.

6 Rate of decay for parallel repetition of PCPs

We provide some results on the rate of decay of the soundness error for the parallel repetition of a PCP (if soundness error does decrease). Let PCP be a PCP for a language L. Let β_{PCP_t} be the soundness error of the t-wise parallel repetition of PCP, and let β_{MIP_t} be the soundness error of the t-wise parallel repetition of the MIP projection of PCP. Claim 5.6 and Claim 5.7 together imply that

$$\forall\, \mathbb{x} \notin L\,,\ \forall\, t \in \mathbb{N}\,,\ \beta_{\mathsf{PCP}_t}(\mathbb{x}) \leq \beta_{\mathsf{MIP}_t}(\mathbb{x})\ .$$

In other words, the rate of decay of parallel repetition of a PCP cannot be worse than that for the corresponding MIP projection. If $\beta_{\mathsf{PCP}_t}(\mathbb{x}) = 1$ then the inequality implies that $\beta_{\mathsf{PCP}_t}(\mathbb{x}) = \beta_{\mathsf{MIP}_t}(\mathbb{x}) = 1$ (a degenerate case). Here we ask whether equality can hold even if $\beta_{\mathsf{PCP}_t}(\mathbb{x}) < 1$.

In those cases where $\beta_{\mathsf{PCP}_t}(\mathbb{x}) = \beta_{\mathsf{MIP}_t}(\mathbb{x})$, we parallel repetition for the PCP exactly follows the rate of decay for the corresponding MIP projection, for which there may be known results that apply (since parallel repetition for MIPs has been studied in a rich line of work).

Below we prove that for a particular class of PCPs, which we call *PCP evaluations of MIPs*, the soundness error in parallel repetition behaves as the soundness error of the parallel repetition of the an underlying MIP.

Definition 6.1. *Let* $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[\mathsf{k}]}, \mathcal{V})$ *be a one-round MIP system for a language* L *with verifier alphabet* $\Sigma_{\mathcal{V}}$*, prover alphabet* $\Sigma_{\mathcal{P}}$*, verifier randomness complexity* vr*, and prover randomness complexity* pr*. The* **PCP evaluation of** MIP $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ *is a PCP for* L *defined as follows.*

- $\mathbf{P}(\mathbb{x}; \tau)$*:*
  1. *Set* $\pi := (\sigma)^{\mathsf{k}\cdot|\Sigma_{\mathcal{V}}|}$*, where* σ *is an arbitrary symbol in* $\Sigma_{\mathcal{P}}$*.*
  2. *For every* $i \in [\mathsf{k}]$ *and* $a \in \Sigma_{\mathcal{V}}$*:*
     (a) *Compute the MIP prover message* $b_i := \mathcal{P}_i(\mathbb{x}, \tau, a)$*.*
     (b) *Set* $\pi[(i, a)] := b_i$ *(we implicitly cast* (i, a) *to be an integer in* $[\mathsf{k} \cdot |\Sigma_{\mathcal{V}}|]$ *to index into* π*).*
  3. *Output* π*.*
- $\mathbf{V}^{\pi}(\mathbb{x}; \rho)$*:*
  1. *Compute the list of MIP verifier messages* $(a_1, \ldots, a_{\mathsf{k}}) := \mathcal{V}(\mathbb{x}, \rho)$*.*
  2. *Make* k *queries to* π*: for every* $i \in [\mathsf{k}]$*,* $b_i := \pi[(i, a_i)]$*.*
  3. *Check that* $\mathcal{V}(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{k}]})$*.*

For PCP, the proof alphabet is $\Sigma_{\mathcal{P}}$, the proof length is $\mathsf{k} \cdot |\Sigma_{\mathcal{V}}|$, and the query complexity is k. The PCP verifier and prover randomness are the same as those of the MIP. The completeness error and the soundness error of PCP are the same as those of MIP. (We prove this for the soundness error in Lemma 6.3.)

Theorem 6.2. *Let* MIP *be a (non-adaptive) MIP system for a language* L *with soundness error less than* 1*; denote by* β_{MIP_t} *the soundness error of* $\otimes[\mathsf{MIP}, t]$*. Let* PCP *be the PCP evaluation of* MIP*, and let* β_{PCP_t} *be the soundness error of* $\otimes\,[\mathsf{PCP}, t]$*. For every instance* $\mathbb{x} \notin L$ *and every* $t \in \mathbb{N}$*,*

$$\beta_{\mathsf{PCP}_t}(\mathbb{x}) = \beta_{\mathsf{MIP}_t}(\mathbb{x}) < 1\ .$$

6.1 Soundness error of PCP evaluations

Lemma 6.3. *Let* $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[\mathsf{k}]}, \mathcal{V})$ *be a* k*-prover one-round MIP system for a language* L *with soundness error* β_{MIP}*. Let* $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ *be the PCP evaluation of* MIP *with soundness error* β_{PCP}*. For every instance* $\mathbb{x} \notin L$*,*

$$\beta_{\mathsf{MIP}}(\mathbb{x}) = \beta_{\mathsf{PCP}}(\mathbb{x})\ .$$

Proof. Fix an instance $\mathbb{x} \notin L$. We show that $\beta_{\mathsf{MIP}}(\mathbb{x}) \leq \beta_{\mathsf{PCP}}(\mathbb{x})$ and $\beta_{\mathsf{PCP}}(\mathbb{x}) \leq \beta_{\mathsf{MIP}}(\mathbb{x})$.

$\beta_{\mathsf{MIP}}(\mathbb{x}) \leq \beta_{\mathsf{PCP}}(\mathbb{x})$. Let $(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{k}]}$ be (deterministic) malicious provers for MIP such that when interacting with $(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{k}]}$, $\mathcal{V}(\mathbb{x})$ accepts with probability $\beta_{\mathsf{MIP}}(\mathbb{x})$. We consider the malicious prover $\widetilde{\mathbf{P}}$ for PCP that is defined in the same way as the honest PCP evaluation prover in Definition 6.1 with respect to $(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{k}]}$.

Let $\widetilde{\pi}$ be the output of $\widetilde{\mathbf{P}}$. Let $\rho \in \{0,1\}^{\mathsf{vr}}$ be an MIP verifier randomness such that $\langle(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{k}]}, \mathcal{V}(\mathbb{x},\rho)\rangle = 1$, which is equivalent to

$$\mathcal{V}(\mathbb{x},\rho,(b_i)_{i\in[\mathsf{k}]}) = 1 \ ,$$

where $b_i \coloneqq \widetilde{\mathcal{P}}_i(\mathcal{V}(\mathbb{x},\rho)[i]) = \widetilde{\pi}[(i,\mathcal{V}(\mathbb{x},\rho)[i])]$. Hence, by the construction of the PCP evaluation verifier (Definition 6.1), $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x};\rho) = 1$. We conclude that $\beta_{\mathsf{MIP}}(\mathbb{x}) \leq \beta_{\mathsf{PCP}}(\mathbb{x})$.

$\beta_{\mathsf{PCP}}(\mathbb{x}) \leq \beta_{\mathsf{MIP}}(\mathbb{x})$. Let $\widetilde{\mathbf{P}}$ be a (deterministic) malicious prover for PCP such that $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x})$ accepts with probability $\beta_{\mathsf{PCP}}(\mathbb{x})$, where $\widetilde{\pi} \coloneqq \widetilde{\mathbf{P}}$. For every $i \in [\mathsf{k}]$, we consider the (deterministic) malicious prover $\widetilde{\mathcal{P}}_i$ for MIP defined as follows:

$\widetilde{\mathcal{P}}_i(a_i)$:
1. Compute the PCP string $\widetilde{\pi} \coloneqq \widetilde{\mathbf{P}}$.
2. Output $\widetilde{\pi}[(i,a_i)]$.

Let $\rho \in \{0,1\}^{\mathsf{vr}}$ be a PCP verifier randomness such that $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x};\rho)$ accepts, which, by Definition 6.1, is equivalent to

$$\mathbf{V}\left(\mathbb{x},\rho,(\widetilde{\pi}[(i,\mathcal{V}(\mathbb{x};\rho)[i])])_{i\in[\mathsf{k}]}\right) = 1 \ .$$

Since for every $i \in [\mathsf{k}]$, $\widetilde{\mathcal{P}}_i$ sends the message $\widetilde{\pi}[(i,\mathcal{V}(\mathbb{x},\rho)[i])$ to $\mathcal{V}$, it is the case that $\langle(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{k}]}, \mathcal{V}(\mathbb{x},\rho)\rangle = 1$. Therefore, we conclude that $\beta_{\mathsf{MIP}}(\mathbb{x}) \geq \beta_{\mathsf{PCP}}(\mathbb{x})$. □

6.2 MIP projection and PCP evaluation are (almost) inverses

Let $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[\mathsf{k}]}, \mathcal{V})$ be a k-prover one-round MIP system for a language L with verifier alphabet $\Sigma_{\mathcal{V}}$. Let $\mathsf{PCP} = (\mathbf{P},\mathbf{V})$ be the PCP evaluation of MIP. Let $\mathsf{MIP}' = ((\mathcal{P}'_i)_{i\in[\mathsf{k}]}, \mathcal{V}')$ be the MIP projection of PCP.

Intuitively, it should be the case that MIP and MIP′ are the same proof systems. However, this is not true because messages sent by $\mathcal{V}$ are of the form $(a_i)_{i\in[\mathsf{k}]} \in \Sigma_{\mathcal{V}}^{\mathsf{k}}$, while messages sent by $\mathcal{V}'$ are of the form $((i,a_i))_{i\in[\mathsf{k}]} \in ([\mathsf{k}]\times\Sigma_{\mathcal{V}})^{\mathsf{k}}$.

The following lemma states that MIP and MIP′ are almost the same except for the above-mentioned syntactic mismatch.

Lemma 6.4. *There exists a deterministic transformation* Fix *such that*

$$\mathsf{MIP} = \mathsf{Fix}(\mathsf{MIP}') \ .$$

Moreover, $\mathsf{Fix}(\mathsf{MIP}')$ *is a* k*-prover one-round MIP system for L that has the same completeness error, soundness error, prover message alphabet, and prover and verifier randomness complexity as* MIP′.

Proof. We describe MIP′ using MIP as follows:

1. $\mathcal{V}'(\mathbb{x},\rho)$:
   (a) Compute the messages of $\mathcal{V}$: $(a_i)_{i\in[\mathsf{k}]} \coloneqq \mathcal{V}(\mathbb{x},\rho)$.
   (b) For every $i \in [\mathsf{k}]$, send (i,a_i) to the i-th prover $\mathcal{P}'_i$.

2. For $i \in [\mathsf{k}]$, $\mathcal{P}'_i(\mathbb{x}, \tau, (i, a_i))$:
   (a) Compute the answer by $\mathcal{P}_i$: $b_i := \mathcal{P}_i(\mathbb{x}, \tau, a_i)$.
   (b) Send b_i to $\mathcal{V}_2$.
3. $\mathcal{V}'(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{k}]})$: Check that $\mathcal{V}(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{k}]}) = 1$.

Fortunately, we observe that the syntactic mismatch of the messages sent by $\mathcal{V}$ and those sent by $\mathcal{V}'$ is the only issue preventing us from concluding that MIP projection and PCP evaluation are inverses of each other. Hence, we can define a transformation, denoted as Fix, that changes MIP' back to MIP.

$\mathsf{Fix}(\mathsf{MIP}' = ((\mathcal{P}'_i)_{i\in[\mathsf{k}]}, \mathcal{V}'))$: Output $\mathsf{MIP}^* = ((\mathcal{P}^*_i)_{i\in[\mathsf{k}]}, \mathcal{V}^*)$ that works as follows:
1. $\mathcal{V}^*(\mathbb{x}, \rho)$:
   (a) Compute the messages by $\mathcal{V}'$: $((i, a_i))_{i\in[\mathsf{k}]} := \mathcal{V}'(\mathbb{x}, \rho)$.
   (b) For every $i \in [\mathsf{k}]$, send a_i to the i-th prover $\mathcal{P}^*_i$.
2. For every $i \in [\mathsf{k}]$, $\mathcal{P}^*_i(\mathbb{x}, \tau, a_i)$: Send $b_i := \mathcal{P}'_i(\mathbb{x}, \tau, (i, a_i))$ to $\mathcal{V}^*$.
3. $\mathcal{V}^*(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{k}]})$: Check that $\mathcal{V}'(\mathbb{x}, \rho, (b_i)_{i\in[\mathsf{k}]}) = 1$.

It follows straightforwardly that $\mathsf{MIP} = \mathsf{Fix}(\mathsf{MIP}')$. Moreover, Fix only removes the extra index i from each verifier message. Hence, all error parameters and efficiency measure are preserved after Fix except for the verifier alphabet.

We summarize the relationship between $\mathsf{MIP}, \mathsf{PCP}$ and MIP' in Figure 4.

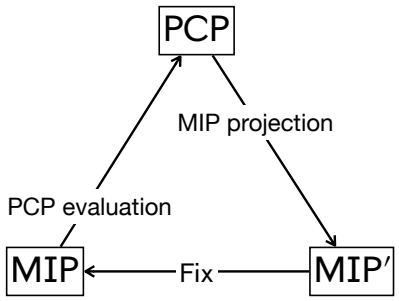

Figure 4: Relationship between $\mathsf{MIP}, \mathsf{PCP}, \mathsf{MIP}'$.

□

Corollary 6.5. *For every $t \in \mathbb{N}$, let β_{MIP_t} be the soundness error of $\otimes[\mathsf{MIP}, t]$ and $\beta_{\mathsf{MIP}'_t}$ be the soundness error of $\otimes[\mathsf{MIP}', t]$,*

$$\beta_{\mathsf{MIP}_t} = \beta_{\mathsf{MIP}'_t} \ .$$

Proof. Let Fix be the transformation in Lemma 6.4. Let $\beta_{\mathsf{MIP}^*_t}$ be the soundness error of $\otimes[\mathsf{Fix}(\mathsf{MIP}'), t]$. By the construction of Fix in Lemma 6.4,

$$\beta_{\mathsf{MIP}'_t} = \beta_{\mathsf{MIP}^*_t} \ .$$

In fact, any prover strategy for $\otimes[\mathsf{Fix}(\mathsf{MIP}'), t]$ can be converted to an equally effective prover strategy for $\otimes[\mathsf{MIP}', t]$, and vice versa. Since $\otimes[\mathsf{MIP}, t] = \otimes[\mathsf{Fix}(\mathsf{MIP}'), t]$ by Lemma 6.4, we conclude the proof. □

6.3 Proof of Theorem 6.2

Let β_{MIP} be the soundness error of MIP and β_{PCP} be the soundness error of PCP. Fix $\mathbb{x} \notin L$ and $t \in \mathbb{N}$.

Since $\beta_{\mathsf{MIP}}(\mathbb{x}) < 1$, Lemma 6.3 implies that $\beta_{\mathsf{PCP}}(\mathbb{x}) < 1$. Hence, by Remark 5.10, $\beta_{\mathsf{PCP}_t}(\mathbb{x}) < 1$.

Next we argue that $\beta_{\mathsf{PCP}_t}(\mathbb{x}) = \beta_{\mathsf{MIP}_t}(\mathbb{x})$.

Upper bound for $\beta_{\mathsf{PCP}_t}(\mathbb{x})$: $\beta_{\mathsf{PCP}_t}(\mathbb{x}) \leq \beta_{\mathsf{MIP}_t}(\mathbb{x})$. Let MIP'_t be the t-wise parallel repetition of the MIP projection of PCP; denote by $\beta_{\mathsf{MIP}'_t}$ its soundness error. Corollary 6.5 implies that

$$\beta_{\mathsf{MIP}_t}(\mathbb{x}) = \beta_{\mathsf{MIP}'_t}(\mathbb{x}) \ .$$

Let MIP''_t be the MIP projection of PCP_t with soundness error $\beta_{\mathsf{MIP}''_t}$. By Claim 5.6,

$$\beta_{\mathsf{PCP}_t}(\mathbb{x}) \leq \beta_{\mathsf{MIP}''_t}(\mathbb{x}) \ .$$

Moreover, by Claim 5.7,

$$\beta_{\mathsf{MIP}'_t}(\mathbb{x}) = \beta_{\mathsf{MIP}''_t}(\mathbb{x}) \ .$$

We conclude that $\beta_{\mathsf{PCP}_t}(\mathbb{x}) \leq \beta_{\mathsf{MIP}''_t}(\mathbb{x}) = \beta_{\mathsf{MIP}'_t}(\mathbb{x}) = \beta_{\mathsf{MIP}_t}(\mathbb{x})$ as desired. See Figure 5 for a diagram.

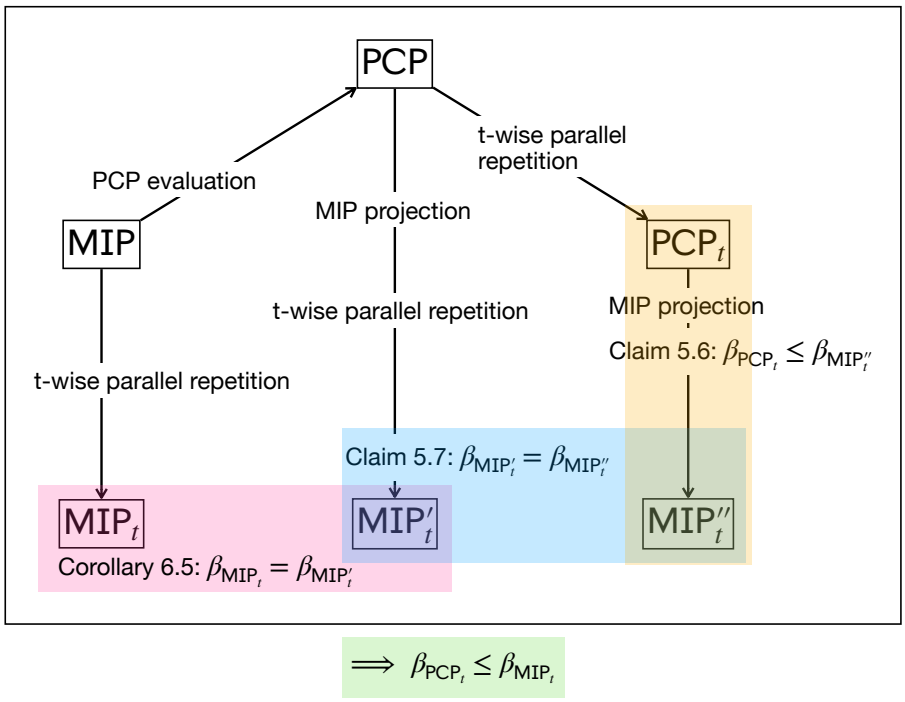

Figure 5: Proof logic for arguing that $\beta_{\mathsf{PCP}_t}(\mathbb{x}) \leq \beta_{\mathsf{MIP}_t}(\mathbb{x})$.

Lower bound for $\beta_{\mathsf{PCP}_t}(\mathbb{x})$: $\beta_{\mathsf{PCP}_t}(\mathbb{x}) \geq \beta_{\mathsf{MIP}_t}(\mathbb{x})$. Let $(\widetilde{\mathcal{P}}_{t,i})_{i\in[\mathsf{k}]}$ be malicious provers for MIP_t. We construct the following malicious prover $\widetilde{\mathbf{P}}_t$ for PCP_t:

$\widetilde{\mathbf{P}}_t(\tau)$:

1. Initialize $\widetilde{\Pi} := (\sigma^t)^{(\mathsf{k}\cdot|\Sigma_{\mathcal{V}}|)^t}$ where σ is an arbitrary symbol in $\Sigma_{\mathcal{P}}$.
2. For every $\mathbf{Q} \in ([\mathsf{k}] \times \Sigma_{\mathcal{V}})^t$:
   (a) Parse $\mathbf{Q}$ as $(i_1, a_1), \ldots, (i_t, a_t)$.
   (b) Set $\widetilde{\Pi}[\mathbf{Q}] := \widetilde{\mathcal{P}}_{t,i_1}(\tau, (a_j)_{j\in[t]})$ (we implicitly cast $\mathbf{Q}$ to be an integer in $[(\mathsf{k}\cdot|\Sigma_{\mathcal{V}}|)^t]$ to index into $\widetilde{\Pi}$).
3. Output $\widetilde{\Pi}$.

Let $\widetilde{\Pi}$ be the output of $\widetilde{\mathbf{P}}_t$. Let verifier randomness $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t) \in (\{0,1\}^{\mathsf{vt}})^t$ be such that $\langle(\widetilde{\mathcal{P}}_{t,i})_{i\in[\mathsf{k}]}, \mathcal{V}_t(\mathbb{x}, \boldsymbol{\rho})\rangle = 1$. We want to show that $\mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho})$ accepts, which implies that $\beta_{\mathsf{MIP}_t} \leq \beta_{\mathsf{PCP}_t}$ as desired.

For every $i \in [\mathsf{k}]$, we use $\mathbf{Q}_i$ to denote the i-th query of $\mathbf{V}_t(\mathbb{x}; \rho)$. By Definition 3.9,

$$\mathbf{Q}_i = (\mathrm{Q}_1[i], \ldots, \mathrm{Q}_t[i]) \ .$$

where for every $j \in [t]$, Q_j is the query list of the verifier $\mathbf{V}(\mathbb{x}; \rho_j)$ for PCP. Moreover, for every $j \in [t]$,

$$\mathrm{Q}_j[i] = (i, \mathcal{V}(\mathbb{x}, \rho_j)[i]) \ ,$$

where $\mathcal{V}$ is the verifier for MIP. We have

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

From Definition 5.1, $\mathcal{V}_t(\mathbb{x}, \boldsymbol{\rho})$ 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, $\mathcal{V}_t(\mathbb{x}, \boldsymbol{\rho}, (\boldsymbol{b}_i)_{i\in[\mathsf{q}]})$ accepts implies $\mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho})$ accepts as desired.

7 Parallel repetition for canonical PCPs

We discuss how parallel repetition behaves for a specific class of problems, constraint satisfaction problems (CSPs). In Section 7.1 we define CSPs. In Section 7.2 we construct a canonical PCP for CSPs. In Section 7.3 we prove results about the parallel repetition for the canonical PCP for CSPs. In Section 7.4 we give an example CSP to show the results in Section 7.3 cannot be generalized further.

In Appendix B we provide additional (and stronger) limitations of parallel repetition for PCPs for a non-standard generalization of CSPs, where each constraint is defined as a list of variables instead of a set of variables. These results shed more light onto how parallel repetition for PCPs fails.

7.1 Constraint satisfaction problems

We define constraint satisfaction problems (CSPs) and the corresponding language CSPSAT. We also define the class of *symmetric* CSPs, on which we focus.

Definition 7.1 (Constraint)**.** *For a finite set Σ, $l \in \mathbb{N}$, and $q \in \mathbb{N}$, a (Σ, l, q)-**constraint** is a pair $C = (S, f)$ where $S \in \binom{[l]}{q}$ and $f \colon \Sigma^S \to \{0,1\}$. An assignment $a \in \Sigma^l$ satisfies C if $f(a(S)) = 1$. Throughout, we assume that, for every S, S has an implicit ordering: $S[i]$ for every $i \in [|S|]$ is well-defined.*

Definition 7.2. *Let Σ be a finite set, $l \in \mathbb{N}$, $q \in \mathbb{N}$, and $m \in \mathbb{N}$. A (Σ, l, q, m)-**CSP** is a list $\phi = (C_1, \ldots, C_m)$ where, for every $i \in [m]$, $C_i = (S_i, f_i)$ is a (Σ, l, q)-constraint. The* **value of ϕ on assignment** *$a \in \Sigma^l$ is*

$$\mathsf{val}(\phi, a) \coloneqq \frac{1}{m} \sum_{i=1}^{m} f_i(a(S)) \ .$$

The **value of** *ϕ is*

$$\mathsf{val}(\phi) \coloneqq \max_{a \in \Sigma^l} \mathsf{val}(\phi, a) \ .$$

We say that ϕ is **satisfiable** *if $\mathsf{val}(\phi) = 1$.*

Definition 7.3 (CSPSAT)**.** *For a finite set Σ, $l \in \mathbb{N}$, $q \in \mathbb{N}$, and $m \in \mathbb{N}$,*

$$\text{CSPSAT}[\Sigma, l, q, m] \coloneqq \{\phi : \phi \text{ is a satisfiable } (\Sigma, l, q, m)\text{-CSP}\} \ .$$

Definition 7.4 (Symmetric CSP)**.** *For a finite set Σ, $l \in \mathbb{N}$, $q \in \mathbb{N}$, $m \in \mathbb{N}$, and function $f^* \colon \Sigma^q \to \{0,1\}$, a (Σ, l, q, m, f^*)-**CSP** is a (Σ, l, q, m)-CSP ϕ where, for every $C = (S, f) \in \phi$, $f \equiv f^*$ with respect to the implicit ordering of S.[7] Moreover, the language $\text{CSPSAT}[\Sigma, l, q, m, f^*]$ is the set of all satisfiable (Σ, l, q, m, f^*)-CSP.*

7.2 Canonical PCP for CSPSAT

We define a PCP for CSPSAT whose soundness error is the value of the instance ϕ.

Construction 7.5. The PCP $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ for $\text{CSPSAT}[\Sigma, l, q, m]$ is defined as follows.

- $\mathbf{P}(\phi)$:

[7] Recall that the domain of f is Σ^S and the domain of f^* is Σ^q. $f \equiv f^*$ with respect to the implicit ordering of S means that for every assignment $a \in \Sigma^l$, $f(a(S)) = f^*((a(S[i]))_{i \in q})$.

1. If ϕ is satisfiable, output a satisfying assignment $\pi \coloneqq a \in \Sigma^l$.
2. Otherwise, output an arbitrary assignment $\pi \coloneqq a \in \Sigma^l$.

- $\mathbf{V}^\pi(\phi)$:

1. Parse ϕ as $(C_i)_{i\in[m]}$.
2. Sample $i \leftarrow [m]$.
3. Parse C_i as (S_i, f_i).
4. Query π at $S_i[1], \ldots, S_i[q]$.
5. Set $a_i \coloneqq (\pi[S_i[1]], \ldots, \pi[S_i[q]])$.
6. Accept if and only if $f_i(a_i) = 1$.

Lemma 7.6. *For every finite set Σ, $l \in \mathbb{N}$, $q \in \mathbb{N}$, and $m \in \mathbb{N}$, let $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ be the PCP for $\textsc{CspSat}[\Sigma, l, q, m]$ in Construction 7.5. Let α and β be the completeness error and soundness error of PCP. The following holds for every instance ϕ.*
- *Completeness: If $\phi \in \textsc{CspSat}[\Sigma, l, q, m]$, $\alpha(\phi) = 0$.*
- *Soundness: If $\phi \notin \textsc{CspSat}[\Sigma, l, q, m]$, $\beta(\phi) = \mathsf{val}(\phi)$.*

PCP has proof alphabet Σ, query complexity q, proof length l, and verifier randomness complexity $\log_2 m$.

Remark 7.7 ($3\textsc{col}$ as $\textsc{CspSat}$)**.** $3\textsc{col}$ (used in Section 4) is a special case of $\textsc{CspSat}$. Specifically, every graph $G = (V, E)$ can be mapped to a $(\{0,1,2\}, |V|, 2, |E|)$-CSP ϕ such that $G \in 3\textsc{col}$ if and only if $\phi \in \textsc{CspSat}[\{0,1,2\}, |V|, 2, |E|]$. The mapping is as follows:

$G \to \phi$:
1. Initialize ϕ as an empty list.
2. Let $f\colon \{0,1,2\}^2 \to \{0,1\}$ be such that $f(a,b) = 1$ if and only if $a \neq b$.
3. For every edge $e = \{u, v\} \in E$:
   (a) Let $S_e \coloneqq \{u, v\}$.
   (b) Let $C_e \coloneqq (S, f)$.
   (c) Append C_e to ϕ.
4. Output ϕ.

Moreover, the PCP in Construction 4.2 for $3\textsc{col}$ is a special case of the PCP for $\textsc{CspSat}$ in Construction 7.5. Therefore, all results we show in this section apply to the PCP in Construction 4.2.

7.3 Parallel repetition for symmetric CSPs

In Section 5, we show that parallel repetition of a PCP brings the soundness error to 0 precisely when the soundness error of the MIP projection is less than 1. In this section, we characterize further the behavior of parallel repetition for the canonical PCP in Construction 7.5.

For the rest of this section, we fix a finite set Σ, $l \in \mathbb{N}$, $q \in \mathbb{N}$, $m \in \mathbb{N}$, and function $f^*\colon \Sigma^q \to \{0,1\}$. Let PCP be the PCP for $\textsc{CspSat}[\Sigma, l, q, m, f^*]$ in Construction 7.5. For every $t \in \mathbb{N}$, let $\beta(\phi)$ be the soundness error of PCP, and let $\beta_t(\phi)$ be the soundness error of $\otimes[\mathsf{PCP}, t]$. In particular, $\beta_1(\phi) = \beta(\phi)$.

We show two results.

- In Lemma 7.8, we use the characterization in Theorem 5.2 to argue that parallel repetition fails for every unsatisfiable (symmetric) CSP instance ϕ such that $\beta(\phi) > 0$.

- In Lemma 7.9, we show that for every (symmetric) CSP soundness error of parallel repetition is nondecreasing as a function of t.

Note that we are not interested in the case where $\beta(\phi) = 0$ because it implies that $\beta_t(\phi) = 0$ for every $t \in \mathbb{N}$. We briefly argue for the contrapositive of this claim: $\beta_t(\phi) > 0$ implies $\beta(\phi) > 0$. If $\beta_t(\phi) > 0$ then there exists some PCP string $\widetilde{\Pi}_t$ for $\otimes[\mathsf{PCP}, t]$ that makes $\mathbf{V}_t$ accepts for some randomness $\boldsymbol{\rho}$. For every $i \in [t]$, since the PCP verifier $\mathbf{V}$ does not make duplicate queries, we can construct a PCP string that makes $\mathbf{V}$ accepts under $\boldsymbol{\rho}[i]$ using the answers, given by $\widetilde{\Pi}_t$, to $\boldsymbol{\rho}[i]$ in the i-th repetition. Hence $\beta(\phi) > 0$.

Lemma 7.8. *For every instance $\phi \notin \textsc{CspSat}[\Sigma, l, q, m, f^*]$,*

$$\beta(\phi) > 0 \implies \lim_{t\to\infty} \beta_t(\phi) > 0 \ .$$

Proof. Fix $\phi \notin \textsc{CspSat}[\Sigma, l, q, m, f^*]$. Since $\beta(\phi) > 0$, there exist a PCP string $\widetilde{\pi} \in \Sigma^l$ and verifier randomness $\rho \in \{0,1\}^{\mathsf{vr}}$ such that $\mathbf{V}^{\widetilde{\pi}}(\phi; \rho) = 1$. We construct malicious provers $(\widetilde{\mathcal{P}}_i)_{i\in[q]}$ for the MIP projection of PCP, where the provers use their shared randomness to compute $\widetilde{\pi} \in \Sigma^l$ and $\rho \in \{0,1\}^{\mathsf{vr}}$. For every $i \in [q]$,

$\widetilde{\mathcal{P}}_i(a_i)$:
1. Compute the query list $\mathrm{Q} \coloneqq \mathbf{V}_{\mathrm{q}}(\phi, \rho)$.
2. Output $b_i \coloneqq \widetilde{\pi}[\mathrm{Q}[i]]$.

Let $i \in [m]$ be the index of the constraint $C_i = (S_i, f_i)$ the PCP verifier $\mathbf{V}$ selects according to the randomness ρ. Since $\mathbf{V}^{\widetilde{\pi}}(\phi; \rho) = 1$, we know that $f_i(b_1, \ldots, b_q) = 1$. Moreover, since $f_i = f^*$ for every $i \in [m]$, the verifier $\mathcal{V}$ for the MIP projection accepts because $f^*(b_1, \ldots, b_q) = 1$. Hence, the MIP projection has soundness error 1. By Theorem 5.2, $\lim_{t\to\infty} \beta_t(\phi) > 0$. □

Lemma 7.9. *For every instance $\phi \notin \textsc{CspSat}[\Sigma, l, q, m, f^*]$ and $t \in \mathbb{N}$,*

$$\beta_{t+1}(\phi) \geq \beta_t(\phi) \ .$$

Proof. Fix $\phi \notin \textsc{CspSat}[\Sigma, l, q, m, f^*]$. Let $\widetilde{\mathbf{P}}_t$ be a malicious prover for $\otimes[\mathsf{PCP}, t]$. We construct a malicious prover $\widetilde{\mathbf{P}}_{t+1}$ for $\otimes[\mathsf{PCP}, t+1]$.

$\widetilde{\mathbf{P}}_{t+1}$:
1. Compute $\widetilde{\Pi}_t \leftarrow \widetilde{\mathbf{P}}_t$.
2. Set $\widetilde{\Pi}_{t+1} \coloneqq (\sigma^{t+1})^{l^{t+1}}$, where σ is an arbitrary symbol in Σ.
3. For every $(q_1, \ldots, q_{t+1}) \in [l]^{t+1}$:
   (a) Let $(\mathsf{ans}_1, \ldots, \mathsf{ans}_t) \coloneqq \widetilde{\Pi}_t[(q_1, \ldots, q_t)]$.
   (b) Set $\widetilde{\Pi}_{t+1}[(q_1, \ldots, q_{t+1})] \coloneqq (\mathsf{ans}_1, \ldots, \mathsf{ans}_t, \mathsf{ans}_t)$.
4. Output $\widetilde{\Pi}_{t+1}$.

Consider $\boldsymbol{\rho} \in (\{0,1\}^{\mathsf{vr}})^t$ such that $\mathbf{V}_t^{\widetilde{\Pi}_t}(\phi; \boldsymbol{\rho}) = 1$. Since ϕ is symmetric, we know that for every $(\rho_1, \ldots, \rho_t, \rho_{t+1}) \in (\{0,1\}^{\mathsf{vr}})^{t+1}$ such that $(\rho_1, \ldots, \rho_t) = \boldsymbol{\rho}$, $\mathbf{V}_{t+1}^{\widetilde{\Pi}_{t+1}}(\phi; (\rho_1, \ldots, \rho_{t+1})) = 1$. Therefore, we conclude that

$$\beta_{t+1}(\phi) \geq \frac{\beta_t(\phi) \cdot (2^{\mathsf{vr}})^t \cdot 2^{\mathsf{vr}}}{(2^{\mathsf{vr}})^{t+1}} = \beta_t(\phi) \ .$$

□

7.4 Parallel repetition for non-symmetric CSPs

In previous sections, we investigate the behavior of parallel repetition for symmetric CSPs. In particular, we know that for every unsatisfiable symmetric CSP ϕ, $\beta(\phi) > 0$ implies $\lim_{t\to\infty} \beta_t(\phi) > 0$. A natural question to ask is what happens for non-symmetric CSPs.

Lemma 7.10. *Let β be the soundness error of $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ for* CspSat *as defined in Construction 7.5 and β_t be the soundness error of $\otimes[\mathsf{PCP}, t]$ for every $t \in \mathbb{N}$. There exists a non-symmetric CSP $\phi_1 \notin$ CspSat such that*

$$0 < \beta(\phi_1) < 1 \wedge \lim_{t\to\infty} \beta_t(\phi_1) = 0 \ ;$$

in addition, there exists another non-symmetric CSP $\phi_2 \notin$ CspSat such that

$$0 < \beta(\phi_2) < 1 \wedge \lim_{t\to\infty} \beta_t(\phi_2) > 0 \ .$$

To construct non-symmetric CSPs, we consider a well-known NP-complete language 3Sat:

$$\text{3Sat} \coloneqq \{\text{3CNF formula } \psi : \text{there exists a satisfying assignment for } \psi\} \ .$$

We show that every 3CNF formula ψ with l variables and m clauses can be mapped to a $(\{0,1\}, \max\{3,l\}, 3, m)$-CSP ϕ such that $\psi \in$ 3Sat if and only if $\phi \in$ CspSat$[\{0,1\}, \max\{3,l\}, 3, m]$. The mapping is as follows:

$\psi \to \phi$:
1. Set $l' \coloneqq \max\{3, l\}$.
2. Initialize ϕ as an empty list.
3. Parse ψ as $\psi_1 \wedge \cdots \wedge \psi_m$ where ψ_i is a clause for every $i \in [m]$.
4. For every $i \in [m]$:
   (a) Parse ψ_i to obtain the variables $x_{j_1}, x_{j_2}, x_{j_3}$ it contains where $j_1, j_2, j_3 \in [l]$ such that $j_1 \le j_2 \le j_3$.
   (b) Pick arbitrary $j'_1, j'_2, j'_3 \in [l']$ such that $j'_1 < j'_2 < j'_3$ and $\{j_1, j_2, j_3\} \subseteq \{j'_1, j'_2, j'_3\}$.
   (c) Set $S_i \coloneqq \{j'_1, j'_2, j'_3\}$ (where the implicit ordering is $j'_1 < j'_2 < j'_3$).
   (d) Let $f_i \colon \{0,1\}^3 \to \{0,1\}$ be such that $f_i(a_1, a_2, a_3) = 1$ if and only if $\psi_i(a_1, a_2, a_3) = 1$.
   (e) Set $C_i \coloneqq (S_i, f_i)$.
   (f) Append C_i to ϕ.
5. Output ϕ.

Note that the resulting ϕ is in general not symmetric: for two different clauses 3CNF_i and 3CNF_j, the corresponding constraint functions C_i and C_j are not necessarily the same.

Proof. We first construct $\psi_1 \notin$ 3Sat such that the CSP ϕ_1 obtained by the mapping above satisfies $\beta(\phi_1) > 0$ but $\lim_{t\to\infty} \beta_t(\phi_1) = 0$.

$$\psi_1 \coloneqq \begin{array}{l} (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \\ \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \end{array} \ .$$

Observe that ψ_1 is a 3CNF formula that consists of all possible clauses using three variables. ψ_1 has $l = 3$ variables and $m = 8$ clauses. Moreover, $\psi_1 \notin$ 3Sat because each clause rules out exactly one assignment in $\{0,1\}^3$ and there are only 2^3 possible assignments.

Let ϕ_1 be the CSP obtained from ψ_1. Observe that ϕ_1 is non-symmetric.

We first argue that $0 < \beta(\phi_1) < 1$. We consider an assignment $a = (a_1, a_2, a_3) \in \{0,1\}^3$ of (x_1, x_2, x_3) such that $\Pr[\mathbf{V}^a(\phi_1)] = \beta(\phi_1)$. Note that a satisfies all except one clause: $\psi \coloneqq l_1 \vee l_2 \vee l_3$ where for every $i \in [3]$, $l_i = x_i$ if $a_i = 0$ and $l_i = \neg x_i$ otherwise. In particular, this implies that $\beta(\phi_1) = 7/8$.

Now we show that $\lim_{t\to\infty} \beta_t(\phi_1) = 0$. It suffices to show $\beta_{\mathsf{MIP}}(\phi_1) < 1$ where β_{MIP} is the soundness error for the MIP projection $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[3]}, \mathcal{V})$ of PCP according to Theorem 5.2. Assume for the sake of contradiction that $\beta_{\mathsf{MIP}}(\phi_1) = 1$. Let $(\widetilde{\mathcal{P}}_i)_{i\in[3]}$ be the (deterministic) malicious provers for the MIP projection that makes the verifier $\mathcal{V}$ always accept. We define a satisfying assignment $a \in \{0,1\}^3$ for ϕ_1 as follows:

$$a \coloneqq (\widetilde{\mathcal{P}}_1(x_1), \widetilde{\mathcal{P}}_2(x_2), \widetilde{\mathcal{P}}_3(x_3)) \ .$$

For every $i \in [3]$, $\mathcal{V}$ always sends the message x_i to the i-th prover since the variables have the implicit ordering $x_1 < x_2 < x_3$. Thus, the MIP verifier $\mathcal{V}$ always accepts implies that $\mathbf{V}^a(\phi_1)$ always accepts, contradicting to $\beta(\phi_1) < 1$. Therefore, we conclude that $\beta_{\mathsf{MIP}}(\phi_1) < 1$, which implies $\lim_{t\to\infty} \beta_t(\phi_1) = 0$.

Now we construct $\psi_2 \notin 3\text{SAT}$ such that the CSP ϕ_2 obtained by the mapping above satisfies $\lim_{t\to\infty} \beta_t(\phi_2) > 0$:

$$\psi_2 \coloneqq \begin{array}{l}(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge \\ (\neg x_3 \vee x_4 \vee x_5) \wedge (\neg x_3 \vee x_4 \vee \neg x_5) \wedge (\neg x_3 \vee \neg x_4 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)\end{array} \ .$$

Observe that ψ_2 is a 3CNF formula that consists of all possible clauses with literals $\{x_1, x_2, x_3\}$ and all possible clauses with literal $\{x_4, x_5, \neg x_3\}$. ψ_2 has $l = 5$ variables and $m = 8$ clauses. $\psi_2 \notin 3\text{SAT}$ since if x_3 is assigned to 0, one of the first four clauses cannot be satisfied; and if x_3 is assigned to 1, one the the last four clauses cannot be satisfied.

Since $\psi_2 \notin 3\text{SAT}$, $\text{val}(\phi_2) < 1$. By Lemma 7.6, $\beta(\phi_2) = \text{val}(\phi_2) < 1$.

Now we show that $\lim_{t\to\infty} \beta_t(\phi_2) > 0$. It suffices to show $\beta_{\mathsf{MIP}}(\phi_2) = 1$ where β_{MIP} is the soundness error for the MIP projection $\mathsf{MIP} = ((\mathcal{P}_i)_{i\in[3]}, \mathcal{V})$ of PCP according to Theorem 5.2. Consider the malicious provers for MIP defined as follows:

- $\widetilde{\mathcal{P}}_1(a_1)$: Output 0.
- $\widetilde{\mathcal{P}}_2(a_2)$: Output 1.
- $\widetilde{\mathcal{P}}_3(a_3)$: Output 1.

Observe that $\langle(\widetilde{\mathcal{P}}_i)_{i\in[3]}, \mathcal{V}(\phi_2)\rangle = 1$ for all verifier randomness since every clause of ψ_2 evaluates to 1 given the provers' responses.

□

8 Consistent parallel repetition

We describe a variant of parallel repetition for PCPs, which we call *consistent parallel repetition*. Briefly, the verifier additionally performs a consistency check across all repetitions.

Definition 8.1 (Consistent parallel repetition)**.** *Let* $\mathsf{PCP} = (\mathbf{P}, \mathbf{V} = (\mathbf{V}_{\mathrm{q}}, \mathbf{V}_{\mathrm{d}}))$ *be a (non-adaptive) PCP system with proof alphabet* Σ*, proof length* l*, query complexity* q*, and verifier randomness* vr*. The* t**-wise consistent parallel repetition** *of* PCP*, denoted* $\hat{\otimes}(\mathsf{PCP}, t)$*, is a PCP system defined as follows.*

- $\mathbf{P}_t(\mathbb{x})$*:*
  1. *Compute the PCP string* $\pi \leftarrow \mathbf{P}(\mathbb{x})$.
  2. *Set* $\Pi := ((\pi[q_1], \ldots, \pi[q_t]))_{(q_1,\ldots,q_t)\in[\mathsf{l}]^t}$.
  3. *Output* Π.
- $\mathbf{V}_t^{\Pi}(\mathbb{x})$*:*
  1. *Initialize* $\pi := (\bot)^{\mathsf{l}}$ *where* $\bot \notin \Sigma$.
  2. *For* $i \in [t]$*:*
     (a) *Sample PCP verifier randomness* $\rho_i \leftarrow \{0,1\}^{\mathsf{vr}}$.
     (b) *Compute the query list of the PCP verifier* $\mathrm{Q}_i := \mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho_i)$.
  3. *For* $i \in [\mathsf{q}]$*:*
     (a) *Set* $\mathbf{Q}_i := (\mathrm{Q}_1[i], \ldots, \mathrm{Q}_t[i])$.
     (b) *Query the proof string* $\mathsf{ans}_i := \Pi[\mathbf{Q}_i]$.
     (c) *For* $j \in [t]$*:*
        i. *If* $\pi[\mathbf{Q}_i[j]] = \bot$*, set* $\pi[\mathbf{Q}_i[j]] := \mathsf{ans}_i[j]$.
        ii. *Else if* $\pi[\mathbf{Q}_i[j]] \neq \mathsf{ans}_i[j]$*, reject.*
  4. *Check that* $\bigwedge_{i\in[t]} \mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho_i, \mathsf{ans}_1[i], \ldots, \mathsf{ans}_{\mathsf{q}}[i])$.

(The lines in blue are the differences with standard parallel repetition, in Definition 3.9.)

As in (standard) parallel repetition, the proof alphabet becomes Σ^t, the proof length becomes l^t, and the verifier randomness becomes $t \cdot \mathsf{vr}$; the query complexity remains q.

The completeness error becomes $1 - (1-\alpha)^t$, as in (standard) parallel repetition. In contrast, in this case, we show that the soundness error tends to 0 as t tends to infinity (the desired behavior).

Before the theorem, we state a counting problem and a preliminary solution, which we use in our theorem. The analysis of the counting problem is postponed to Appendix C.

Definition 8.2. *For every alphabet* Σ*, length* $n \in \mathbb{N}$*, and number of distinct entries* $m \in \mathbb{N}$*. We define*

$$\mathcal{K}(\Sigma, n, m) := |\{s = (s_1, \ldots, s_n) \in \Sigma^n : |\{s_1, \ldots, s_n\}| \leq m\}| \quad .$$

Lemma 8.3. *For every alphabet* Σ *and integers* $n \in \mathbb{N}$ *and* $m \leq |\Sigma|$*,*

$$\mathcal{K}(\Sigma, n, m) = \sum_{i=1}^{m} \binom{|\Sigma|}{i} \sum_{j=1}^{i} \binom{i}{j} \cdot (-1)^{i-j} \cdot j^n \leq \binom{|\Sigma|}{m} \cdot m^n \quad ,$$

in which the upper bound is attained if and only if $m = 1$.

Theorem 8.4. *Let* $\mathsf{PCP} = (\mathbf{P}, \mathbf{V} = (\mathbf{V}_{\mathrm{q}}, \mathbf{V}_{\mathrm{d}}))$ *be a (non-adaptive) PCP system for a language* L *with soundness error* β *and verifier randomness complexity* vr*. Let* $\hat{\beta}_t$ *be the soundness error for* $\hat{\otimes}(\mathsf{PCP}, t) = (\mathbf{P}_t, \mathbf{V}_t)$ *(the* t*-wise consistent parallel repetition of* PCP*). For every* $t \in \mathbb{N}$ *and* $\mathbb{x} \notin L$*,*

$$\beta(\mathbb{x}) = 0 \implies \hat{\beta}_t(\mathbb{x}) = 0$$

and

$$0 < \beta(\mathbb{x}) < 1 \implies \hat{\beta}_t(\mathbb{x}) < \frac{\mathcal{K}(\{0,1\}^{2^{\mathsf{vr}}}, t, \beta(\mathbb{x}) \cdot 2^{\mathsf{vr}})}{(2^{\mathsf{vr}})^t} = O_{\mathbb{x}}(1) \cdot \beta(\mathbb{x})^t \ ,$$

where $O_{\mathbb{x}}(1)$ *hides a constant that is at most* $\binom{2^{\mathsf{vr}}}{\beta(\mathbb{x}) \cdot 2^{\mathsf{vr}}}$*.*

Corollary 8.5. *Let* $\mathsf{PCP} = (\mathbf{P}, \mathbf{V} = (\mathbf{V}_{\mathrm{q}}, \mathbf{V}_{\mathrm{d}}))$ *be a (non-adaptive) PCP system for a language* L *with soundness error* β *and verifier randomness complexity* vr*. Let* $\hat{\beta}_t$ *be the soundness error for* $\hat{\otimes}(\mathsf{PCP}, t) = (\mathbf{P}_t, \mathbf{V}_t)$ *(the* t*-wise consistent parallel repetition of* PCP*). For every* $\mathbb{x} \notin L$*,*

$$\beta(\mathbb{x}) < 1 \implies \lim_{t \to \infty} \hat{\beta}_t(\mathbb{x}) = 0 \ .$$

Corollary 8.5 directly follows from Theorem 8.4 because $O_{\mathbb{x}}(1)$ is a constant and $\lim_{t \to \infty} \beta(\mathbb{x})^t = 0$.

8.1 Proof of Theorem 8.4

Fix $t \in \mathbb{N}$ and $\mathbb{x} \notin L$. Let $\widetilde{\mathbf{P}}_t$ be an optimal malicious prover for $\hat{\otimes}(\mathsf{PCP}, t)$ given instance $\mathbb{x}$. Let $\widetilde{\Pi}$ be the output of $\widetilde{\mathbf{P}}_t$. We say that $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t) \in (\{0,1\}^{\mathsf{vr}})^t$ has *set cardinality* k, denoted $|\{\boldsymbol{\rho}\}| = k$, if $|\{\rho_1, \ldots, \rho_t\}| = k$. We construct a malicious prover $\widetilde{\mathbf{P}}$ for PCP as follows.

$\widetilde{\mathbf{P}}$:

1. Run $\widetilde{\Pi} \leftarrow \widetilde{\mathbf{P}}_t$.
2. Initialize $\widetilde{\pi} \coloneqq (\sigma)^{\mathsf{l}}$, where σ is an arbitrary symbol in Σ.
3. Set $\boldsymbol{\rho}^* \coloneqq (0^{\mathsf{vr}})^t$.
4. For every $\boldsymbol{\rho} \in (\{0,1\}^{\mathsf{vr}})^t$:
   (a) Run $d \leftarrow \mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho})$, and let $((\mathbf{Q}_i, \mathsf{ans}_i))_{i \in [t]}$ be the list of queries made by $\mathbf{V}_t$ to $\widetilde{\Pi}$ along with their corresponding answers.
   (b) If $d = 1$ and $|\{\boldsymbol{\rho}\}| \geq |\{\boldsymbol{\rho}^*\}|$:
      i. Set $\boldsymbol{\rho}^* \coloneqq \boldsymbol{\rho}$.
      ii. For every $i \in [t]$ and $j \in [\mathsf{q}]$, set $\widetilde{\pi}[\mathbf{Q}_i[j]] \coloneqq \mathsf{ans}_i[j]$.
5. Output $(\boldsymbol{\rho}^*, \widetilde{\pi})$.

Let $(\boldsymbol{\rho}^*, \widetilde{\pi})$ be the output of $\widetilde{\mathbf{P}}$. By Definition 3.2,

$$\Pr\left[\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho) = 1 \;\middle|\; \begin{array}{l} (\boldsymbol{\rho}^*, \widetilde{\pi}) \leftarrow \widetilde{\mathbf{P}} \\ \rho \leftarrow \{0,1\}^{\mathsf{vr}} \end{array}\right] \leq \beta(\mathbb{x}) \ .$$

The *winning set* $W_{\widetilde{\pi}}$ for $\widetilde{\pi}$ is defined as:

$$W_{\widetilde{\pi}} \coloneqq \{\rho \in \{0,1\}^{\mathsf{vr}} : \mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho) = 1\} \ .$$

By the definition of soundness error of PCPs (Definition 3.2), $|W_{\widetilde{\pi}}| \leq \beta(\mathbb{x}) \cdot 2^{\mathsf{vr}}$. Moreover, by construction of $\widetilde{\pi}$, parsing $\boldsymbol{\rho}^*$ as $(\rho_1, \ldots, \rho_t)$, we know from Definition 8.1 (in particular the consistency check) that, for

every $i \in [t]$, $\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}; \rho_i) = 1$. Therefore,

$$|\{\boldsymbol{\rho}^*\}| \leq |W_{\widetilde{\pi}}| \leq \beta(\mathbb{x}) \cdot 2^{\mathsf{vr}} \ .$$

We similarly define the *winning set* $W_{t,\widetilde{\Pi}}$ for $\widetilde{\Pi}$:

$$W_{t,\widetilde{\Pi}} \coloneqq \{\boldsymbol{\rho} \in (\{0,1\}^{\mathsf{vr}})^t : \mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho}) = 1\} \ .$$

Note that $\boldsymbol{\rho}^*$ is a randomness in $W_{t,\widetilde{\Pi}}$ with maximum set cardinality, in other words, $\boldsymbol{\rho}^*$ has maximum number of distinct elements. Hence, every randomness in $W_{t,\widetilde{\Pi}}$ has at most $\beta(\mathbb{x}) \cdot 2^{\mathsf{vr}}$ number of distinct elements. By Lemma 8.3,

$$\left|W_{t,\widetilde{\Pi}}\right| \leq \mathcal{K}(\{0,1\}^{2^{\mathsf{vr}}}, t, \beta(\mathbb{x}) \cdot 2^{\mathsf{vr}}) \leq \binom{2^{\mathsf{vr}}}{\beta(\mathbb{x}) \cdot 2^{\mathsf{vr}}} \cdot (\beta(\mathbb{x}) \cdot 2^{\mathsf{vr}})^t \ .$$

By the definition of winning set and optimality of $\widetilde{\mathbf{P}}_t$,

$$\hat{\beta}_t(\mathbb{x}) = \frac{\left|W_{t,\widetilde{\Pi}}\right|}{(2^{\mathsf{vr}})^t} \ ,$$

which implies that

$$\hat{\beta}_t(\mathbb{x}) \leq \frac{\mathcal{K}(\{0,1\}^{2^{\mathsf{vr}}}, t, \beta(\mathbb{x}) \cdot 2^{\mathsf{vr}})}{(2^{\mathsf{vr}})^t} \leq \binom{2^{\mathsf{vr}}}{\beta(\mathbb{x}) \cdot 2^{\mathsf{vr}}} \cdot \beta(\mathbb{x})^t \ .$$

Therefore, if $\beta(\mathbb{x}) = 0$ then $\hat{\beta}_t(\mathbb{x}) = 0$ as desired.

We are left show that if $0 < \beta(\mathbb{x}) < 1$ then

$$\left|W_{t,\widetilde{\Pi}}\right| < \mathcal{K}(\{0,1\}^{2^{\mathsf{vr}}}, t, \beta(\mathbb{x}) \cdot 2^{\mathsf{vr}}) \ .$$

Suppose for the sake of contradiction that $0 < \beta(\mathbb{x}) < 1$ and

$$\left|W_{t,\widetilde{\Pi}}\right| = \mathcal{K}(\{0,1\}^{2^{\mathsf{vr}}}, t, \beta(\mathbb{x}) \cdot 2^{\mathsf{vr}}) \ .$$

This implies that there exists a malicious proof $\widetilde{\Pi}$ for $\hat{\otimes}(\mathsf{PCP}, t)$ such that for every $\boldsymbol{\rho} \in (\{0,1\}^{\mathsf{vr}})^t$ with $|\{\boldsymbol{\rho}\}| \leq \beta(\mathbb{x}) \cdot 2^{\mathsf{vr}}$, $\mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \boldsymbol{\rho}) = 1$. In particular, for every $\rho \in \{0,1\}^{\mathsf{vr}}$, $\mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \rho^t) = 1$. Consider $\widetilde{\pi} \coloneqq \left(\widetilde{\Pi}[q^t]\right)_{q \in [\mathsf{l}]}$. Since every query of $\mathbf{V}_t^{\widetilde{\Pi}}(\mathbb{x}; \rho^t)$ is in the form q^t for some $q \in [\mathsf{l}]$ (the same randomness is repeated t times),

$$\Pr[\mathbf{V}^{\widetilde{\pi}}(\mathbb{x}) = 1] = 1 \ ,$$

which contradicts our assumption that $\beta(\mathbb{x}) < 1$.

Remark 8.6 (duplicate queries)**.** The proof of Theorem 8.4 does not use the fact that the query list of a PCP verifier does not have duplicate queries. In particular, the soundness error of consistent parallel repetition tends to 0 also for PCP verifiers that output query lists where the same query appears multiple times.

A On tightness of the characterization

We provide an example showing that the characterization in Theorem 5.2 is essentially tight, in the sense that there is a PCP such that on certain instances not in the language parallel repetition leads, in the limit, to soundness error $1/2^{\mathsf{vr}}$. After that, by building on this example, we provide several remarks shedding additional light onto properties of the characterization.

Lemma A.1. *There exists a PCP for a language L with verifier randomness complexity* vr *such that there exist infinitely many instances $\mathbb{x} \notin L$ such that, for every $t \in \mathbb{N}$, $\beta_t(\mathbb{x}) = 1/2^{\mathsf{vr}}$. In particular,*

$$\lim_{t\to\infty} \beta_t(\mathbb{x}) = \frac{1}{2^{\mathsf{vr}}} \ .$$

Proof. Consider the following language L that involves systems of linear equations:

$$L \coloneqq \left\{ \left(n, (a_{j,k}, i_{j,k})_{j,k\in[n]}, (b_j)_{j\in[n]}\right) \in \mathbb{N} \times (\mathbb{Z} \times [n])^{n^2} \times \mathbb{Z}^n : \begin{array}{l} \exists\, x \in \mathbb{Z}^n, \forall j \in [n], \\ a_{j,1} \cdot x[i_{j,1}] + \cdots + a_{j,n} \cdot x[i_{j,n}] = b_j \end{array} \right\} .$$

More specifically, an instance $\mathbb{x} \in L$ is a system of linear equations:

$$\begin{cases} a_{1,1} \cdot x[i_{1,1}] + a_{1,2} \cdot x[i_{1,2}] + \cdots + a_{1,n} \cdot x[i_{1,n}] = b_1 \\ a_{2,1} \cdot x[i_{2,1}] + a_{2,2} \cdot x[i_{2,2}] + \cdots + a_{2,n} \cdot x[i_{2,n}] = b_2 \\ \qquad\qquad\qquad\qquad \vdots \\ a_{n,1} \cdot x[i_{n,1}] + a_{n,2} \cdot x[i_{n,2}] + \cdots + a_{n,n} \cdot x[i_{n,n}] = b_n \end{cases} .$$

Below we define a PCP $\mathsf{PCP} = (\mathbf{P}, (\mathbf{V}_{\mathrm{q}}, \mathbf{V}_{\mathrm{d}}))$ for L, which simply selects at random one of the linear equations and checks it.

- $\mathbf{P}(\mathbb{x})$:
  1. Parse $\mathbb{x}$ as $\left(n, (a_{j,k}, i_{j,k})_{j,k\in[n]}, (b_j)_{j\in[n]}\right)$.
  2. Output $x \in \mathbb{Z}^n$ that satisfies the equations. If no such x exists, output an arbitrary element in $\mathbb{Z}^n$.
- $\mathbf{V}_{\mathrm{q}}(\mathbb{x}, \rho)$:
  1. Parse $\mathbb{x}$ as $\left(n, (a_{j,k}, i_{j,k})_{j,k\in[n]}, (b_j)_{j\in[n]}\right)$.
  2. Use ρ to sample $j \in [n]$.
  3. Output $(i_{j,1}, \ldots, i_{j,n})$.
- $\mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho, \mathsf{ans}_1, \ldots, \mathsf{ans}_n)$:
  1. Parse $\mathbb{x}$ as $\left(n, (a_{j,k}, i_{j,k})_{j,k\in[n]}, (b_j)_{j\in[n]}\right)$.
  2. Use ρ to sample $j \in [n]$.
  3. Check that $a_{j,1} \cdot \mathsf{ans}_1 + \cdots + a_{j,n} \cdot \mathsf{ans}_n = b_j$.

The proof alphabet is $\Sigma = \mathbb{Z}$, proof length is $\mathsf{l} = n$, query complexity is $\mathsf{q} = n$, and verifier randomness complexity is $\mathsf{vr} = \log n$, where n is the first element of $\mathbb{x}$.

For $n \geq 2$, consider the instance $\mathbb{x} \coloneqq \left(n, (a_{j,k}, i_{j,k})_{j,k\in[n]}, (b_j)_{j\in[n]}\right)$ where for every $j, k \in [n]$:

- $a_{j,k} \coloneqq 1$;
- $i_{j,k} \coloneqq ((j + k - 2) \bmod n) + 1$;

- $b_j \coloneqq j$.

Explicitly, $\mathbb{x}$ is the following system of linear equations:

$$\begin{cases} x[1]+x[2]+x[3]+\cdots+x[n-2]+x[n-1]+\quad x[n] & =1 \\ x[2]+x[3]+x[4]+\cdots+x[n-1]+\quad x[n]\quad+\quad x[1] & =2 \\ x[3]+x[4]+x[5]+\cdots+\quad x[n]\quad+\quad x[1]\quad+\quad x[2] & =3 \\ \qquad\qquad\qquad\qquad\qquad\vdots & \\ x[n]+x[1]+x[2]+\cdots+x[n-3]+x[n-2]+x[n-1] & =n \end{cases} .$$

For every $j \in [n]$, the j-th equation for $\mathbb{x}$ is equivalent to $\sum_{i\in[n]} x[i] = b_j$. Therefore, $\mathbb{x} \notin L$ because we can satisfy at most $1 < n$ equations simultaneously.

First we argue that $\beta_t(\mathbb{x}) \geq 1/n$. We construct a malicious prover for the repeated PCP.

$\widetilde{\mathbf{P}}_t$:
1. Initialize $\widetilde{\Pi} \coloneqq (0^t)^{n^t}$.
2. For every $(q_1, \ldots, q_t) \in [n]^t$: if $q_1 = 1$ then set $\widetilde{\Pi}[(q_1, \ldots, q_t)] \coloneqq (q_1, \ldots, q_t)$.
3. Output $\widetilde{\Pi}$.

For every verifier randomness $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_t) \in (\{0,1\}^{\log n})^t$, we view ρ_i as an element in $[n]$ for simplicity. Consider a verifier randomness $\boldsymbol{\rho} \coloneqq (\rho_1, \rho_2, \ldots, \rho_t)$ for some arbitrary $(\rho_\ell)_{2\leq\ell\leq t} \in ([n])^{t-1}$ and $\rho_1 = 1$. By construction of $\widetilde{\mathbf{P}}_t$ and the instance $\mathbb{x}$, we know that for every $\ell \in [n]$,

$$\mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho_\ell, \mathsf{ans}_{\ell,1}, \ldots, \mathsf{ans}_{\ell,n}) = \mathbf{V}_{\mathrm{d}}(\mathbb{x}, \rho_\ell, \rho_\ell, 0, \ldots, 0) = 1 \ .$$

Therefore, there are at least $(2^{\log n})^{t-1}$ accepting verifier randomness as desired.

Next we argue that $\beta_t(\mathbb{x}) \leq 1/n$. Let $\mathbf{Q} \coloneqq (\mathrm{Q}_1, \ldots, \mathrm{Q}_n) \in ([n]^t)^n$ be a possible query list of $\mathbf{V}_t$. We observe that any rotation of $\mathbf{Q}$, namely $(\mathrm{Q}_{((1+\ell-1) \bmod n)+1}, \ldots, \mathrm{Q}_{((n+\ell-1) \bmod n)+1})$ for any $\ell \in [n]$, is also a possible query list. Recall that we assume $\mathrm{Q}_i \neq \mathrm{Q}_j$ for every $i \neq j$. We can see the rotations of $\mathbf{Q}$ are distinct, which means that they're checking different equations. Note that for our instance $\mathbb{x}$, we know that at most 1 equation out of n are satisfiable. Therefore, among rotations of a given query list $\mathbf{Q}$, at most 1 of them can be accepted. We can use the rotation operation to create equivalence classes for verifier randomness, each class of size n. Hence, $\beta_t(\mathbb{x}) \leq 1/n$ as desired. □

Remark A.2. As a sanity check, we show that for the same instance $\mathbb{x}$ in the proof, the MIP projection of the PCP has soundness error 1. We construct malicious provers for the MIP projection.

- $\widetilde{\mathcal{P}}_1(a_1)$: Output a_1.
- For $2 \leq i \leq n$, $\widetilde{\mathcal{P}}_i(a_i)$: Output 0.

We fix a verifier randomness $\rho \in \{0,1\}^{\log n}$ and let j be the sampled index from $[n]$ using ρ. We know that the queries made by the verifier is $\mathrm{Q} \coloneqq (i_{j,1}, \ldots, i_{j,n})$ from construction of $\mathbf{V}_{\mathrm{q}}$. By construction, $\widetilde{\mathcal{P}}_1$ always sends $i_{j,1}$. Moreover, all other provers send 0 always. Therefore, the decision algorithm $\mathbf{V}_{\mathrm{d}}$ always accepts because $i_{j,1} = b_j$ for every j.

Remark A.3. There are examples where the MIP projection has soundness error 1 (Section 4 and Remark A.2), and we mention a class of problems that has MIP projection with soundness error 0 (Remark 5.9). But what about in between? Here we argue that there are infinitely many instance for the language L defined in the proof of Lemma A.1 such that the MIP projection of the PCP has soundness error between 0 and 1.

Consider the following set S of instances:

$$S \coloneqq \left\{ (n, (a_{j,k}, i_{j,k})_{j,k\in[n]}, (b_j)_{j\in[n]}) \in \mathbb{N} \times (\mathbb{Z} \times [n])^{n^2} \times \mathbb{Z}^n : \forall\, j,k \in [n], a_{j,k} = 1, i_{j,k} = k, b_j = j \right\} \ ,$$

where an instance $\mathbb{x} \in S$ is the following system of linear equations for some $n \in \mathbb{N}$:

$$\begin{cases} x[1] + x[2] + \cdots + x[n] = 1 \\ x[1] + x[2] + \cdots + x[n] = 2 \\ \qquad\qquad \vdots \\ x[1] + x[2] + \cdots + x[n] = n \end{cases} .$$

Note that any $\mathbb{x}$ in S is not in L because at most one of the equations can be satisfied. Fix an arbitrary instance $\mathbb{x} \coloneqq (n, (a_{j,k}, i_{j,k})_{j,k\in[n]}, (b_j)_{j\in[n]}) \in S$.

Observe that, for every verifier randomness, the query list is always $(1, \ldots, n)$. Hence, both the PCP and its MIP projection have soundness error at most $1/n$.

On the other hand, we construct a malicious prover $\widetilde{\mathbf{P}}$ for PCP and malicious provers $(\widetilde{\mathcal{P}}_i)_{i\in[n]}$ for the MIP projection as follows:

- $\widetilde{\mathbf{P}}$: Output $(b_1, 0, \ldots, 0)$.
- $\widetilde{\mathcal{P}}_1(a_1)$: Output b_1.
- For $2 \le i \le n$, $\widetilde{\mathcal{P}}_i(a_i)$: Output 0.

Hence, both the MIP projection verifier and the PCP verifier accept if they check the first equation, which means that the soundness error for $\mathbb{x}$ (for both the MIP projection and the PCP) is at least $1/n$.

Remark A.4. If the soundness error of the MIP projection is less than 1 then does it equal that of the PCP? Remarks 5.9 and A.3 exemplifies some cases where these two soundness errors are equal. Nevertheless, here we show examples where the soundness error of MIP projection can be different from the soundness error of the PCP (with both being non-trivial). We revisit the PCP and the language L in the proof of Lemma A.1.

Consider the following set S of instances:

$$S \coloneqq \left\{ \begin{array}{l} (n, (a_{j,k}, i_{j,k})_{j,k\in[n]}, (b_j)_{j\in[n]}) \in \mathbb{Z} \times (\mathbb{Z} \times [n])^{n^2} \times \mathbb{Z}^n : \\ \forall\, j,k \in [n], a_{j,k} = 1, b_j = j, \\ \forall\, j \in \{1, n\}, k \in [n], i_{j,k} = k, \\ \forall\, j \in [2, n-1], k \in [n], i_{j,k} = ((j + k - 2) \bmod n) + 1 \end{array} \right\} .$$

We can write an instance $\mathbb{x} \in S$ as follows for some $n \in \mathbb{N}$:

$$\begin{cases} x[1] + x[2] + x[3] + \cdots + x[n-2] + x[n-1] + x[n] = 1 \\ x[2] + x[3] + x[4] + \cdots + x[n-1] + x[n] + x[1] = 2 \\ x[3] + x[4] + x[5] + \cdots + x[n] + x[1] + x[2] = 3 \\ \qquad\qquad\qquad\qquad \vdots \\ x[n-1] + x[n] + x[1] + \cdots + x[n-4] + x[n-3] + x[n-2] = n-1 \\ x[1] + x[2] + x[3] + \cdots + x[n-2] + x[n-1] + x[n] = n \end{cases} .$$

None of the instances in S belongs to L since at most one of the equations can be satisfied. Let $\mathbb{x} \coloneqq (n, (a_{j,k}, i_{j,k})_{j,k\in[n]}, (b_j)_{j\in[n]}) \in S$ be an arbitrary instance.

Since at most one of the equations can be satisfied for a fixed $x \in \mathbb{Z}^n$, the probability that the PCP verifier accepts $\mathtt{x}$ is at most $1/n$. On the other hand, the soundness error of $\mathtt{x}$ for the MIP projection is $(n-1)/n$. Consider following malicious provers for the MIP projection.

- $\widetilde{\mathcal{P}}_1(a_1)$: Output a_1.
- For $2 \leq i \leq n$, $\widetilde{\mathcal{P}}_i(a_i)$: Output 0.

It is easy to see that $(\widetilde{\mathcal{P}}_i)_{i\in[\mathsf{q}]}$ provides a satisfying assignment for the first $n-1$ equations. Thus, the soundness error of $\mathtt{x}$ for the MIP projection is at least $(n-1)/n$. Moreover, it is at most $(n-1)/n$ since verifier checks the first and last equations with the same answers of the provers and they are not simultaneously satisfiable.

B Constraint satisfaction problems with ordered constraints

We prove stronger results compared to Section 7 for a non-standard generalization of CSPs.

- In Appendix B.1, we define *ordered CSPs*, which are a type of CSP where each constraint is defined as a *list* of variables instead of a *set* of variables. We consider the canonical PCP for ordered CSPs. This PCP equals Construction 7.5 (for standard CSPs) except that instead of sampling a random constraint and checking its satisfiability, the verifier samples a random ordered constraint and checks its satisfiability.
- In Appendix B.2, we show that for ordered CSPs that are symmetric the soundness error of parallel repetition is non-decreasing with respect to the number of repetitions. Moreover, we conclude that if the soundness error for the symmetric ordered CSP is greater than 0 then the soundness error of the parallel repetition of the canonical PCP tends to 1. (In Section 7, we only show that the soundness error of the parallel repetition of the canonical PCP for a symmetric standard CSP tends to a constant greater than 0.)
- In Appendix B.3 we show that, for a class of non-symmetric ordered CSPs, the soundness error of the parallel repetition of the canonical PCP is upper bounded by the soundness error of the canonical PCP.
- In Appendix B.4 we complement Appendix A by exhibiting a PCP such that, for every rational number $c \in (0,1]$, parallel repetition drives the soundness error of certain instances to c (in the limit). This construction builds on the results above.

B.1 CSPs with ordered constraints

We discuss a variant of constraint satisfaction problems.

Definition B.1 (Ordered constraint)**.** *For every finite set Σ, $l \in \mathbb{N}$, and $q \in \mathbb{N}$, an* **ordered (Σ, l, q)-constraint** *is a pair $C = (M, f)$ where M is a list consisting of q elements in $[l]$ and $f\colon \Sigma^q \to \{0,1\}$. An assignment $a \in \Sigma^l$ satisfies C if $f(a[M[1]], \ldots, a[M[q]]) = 1$.*

Remark B.2. We discuss the difference between constraints (Definition 7.1) and ordered constraints (Definition B.1). Consider a set $S \in \binom{[l]}{q}$ with an implicit ordering and a function $f\colon \Sigma^S \to \{0,1\}$. There is a unique (Σ, l, q)-constraint corresponding to the set S and the function f, namely, $C = (S, f)$. In contrast, there are $q!$ possible ordered (Σ, l, q)-constraints corresponding to the set S and the function f. Specifically, for every permutation $p\colon [q] \to [q]$, consider the ordered constraint $C_p \coloneqq (M_p, f_p)$ where:
- M_p is the list of q elements in $[l]$ obtained by permuting S according to p; and
- $f_p\colon \Sigma^q \to \{0,1\}$ is the function that, on input $(a_1, \ldots, a_q)$, applies f to the function $v_p\colon S \to \Sigma$ that assigns the value a_i to $S[p(i)]$ for every $i \in [q]$.

Definition B.3 (Constraint satisfaction problem with ordered constraints)**.** *Let Σ be a finite set, $l \in \mathbb{N}$, $q \in \mathbb{N}$, and $m \in \mathbb{N}$. An* **ordered (Σ, l, q, m)-CSP** *is a list $\phi = (C_1, \ldots, C_m)$ where, for every $i \in [m]$, $C_i = (M_i, f_i)$ is an ordered (Σ, l, q)-constraint.*

The **value of ϕ on assignment** $a \in \Sigma^l$ *is*

$$\mathsf{val}(\phi, a) \coloneqq \frac{1}{m} \sum_{i=1}^{m} f_i(a(M_i[1]), \ldots, a(M_i[q])) \ .$$

The **value of** ϕ *is*

$$\mathsf{val}(\phi) \coloneqq \max_{a \in \Sigma^l} \mathsf{val}(\phi, a) \ .$$

We say that ϕ is **satisfiable** *if* $\mathsf{val}(\phi) = 1$.

Remark B.4. We give an example of an ordered CSP. Consider the NP-complete language independent set:

$$\textsc{IndSet} \coloneqq \{(G = (V, E), k) : \exists\, S \subseteq V \text{ s.t. } |S| \geq k \text{ and } S \text{ is an independent set of } G\} \ .$$

Recall that, in a graph, an independent set is a set of vertices for which no pair of vertices share an edge. We consider $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ for IndSet as follows.

- $\mathbf{P}((G, k))$:
  1. Parse G as a graph (V, E).
  2. Find an independent set $S \coloneqq \{v_1, \ldots, v_k\}$ in G. If no independent set of size k exists, set S to be k arbitrary vertices.
  3. Output the PCP string π that contains a list of the k vertices in S (in arbitrary order).
- $\mathbf{V}^{\pi}((G, k))$:
  1. Parse G as a graph (V, E).
  2. Sample $q_1, q_2 \in [k]^2$ at random such that $q_1 \neq q_2$.
  3. Make two queries to π: $v_{q_1} \coloneqq \pi[q_1]$ and $v_{q_2} \coloneqq \pi[q_2]$.
  4. Accept if and only if $v_{q_1} \neq v_{q_2}$ and $\{v_{q_1}, v_{q_2}\} \notin E$.

While an instance for IndSet can be expressed as a CSP, the above PCP is not the canonical PCP for CSPs in Construction 7.5: for each constraint C with $S = \{q_1, q_2\}$, the verifier of the canonical PCP for CSPs always queries $\{q_1, q_2\}$ in the same order; however, the above PCP verifier queries q_1 before q_2 for some randomness and q_2 before q_1 for some other randomness.

In contrast, ordered CSPs allow different query orders for the same query set. Hence we can map (G, k) to an ordered (Σ, l, q, m, f^*)-CSP ϕ such that $\phi \in \text{OCSPSAT}[\Sigma, l, q, m, f^*]$ if and only if $(G, k) \in \textsc{IndSet}$.

$(G, k) \to \phi$:

1. For every $(i, j) \in [k]^2$ such that $i \neq j$:
   (a) Set $M \coloneqq (i, j)$.
   (b) Let $f \colon V^2 \to \{0, 1\}$ be such that $f(v_1, v_2) = 1$ if and only if $v_1 \neq v_2$ and $\{v_1, v_2\} \notin E$.
   (c) Set $C \coloneqq (M, f)$.
   (d) Append C to ϕ.
2. Output ϕ.

The canonical PCP for this ordered CSP matches with the PCP constructed above for IndSet.

B.2 Parallel repetition for symmetric ordered CSPs

Fix a finite set Σ, $l \in \mathbb{N}$, $q \in \mathbb{N}$, $m \in \mathbb{N}$, and function $f^* \colon \Sigma^q \to \{0, 1\}$. Let PCP be the canonical PCP for $\text{OCSPSAT}[\Sigma, l, q, m, f^*]$ (as in Construction 7.5). Let $\beta_{\mathsf{MIP}}(\phi)$ be the soundness error for the MIP projection of PCP, and for every $t \in \mathbb{N}$ let $\beta_t(\phi)$ be the soundness error for $\otimes[\mathsf{PCP}, t]$.

Lemma B.5 (Analogue of Lemma 7.9)**.** *For every instance $\phi \notin \text{OCSPSAT}[\Sigma, l, q, m, f^*]$ and $t \in \mathbb{N}$,*

$$\beta_{t+1}(\phi) \geq \beta_t(\phi) \ .$$

Proof. The same argument in the proof of Lemma 7.9 works. We include the proof for completeness. Fix $\phi \notin \text{OCSPSAT}[\Sigma, l, q, m, f^*]$. Let $\widetilde{\mathbf{P}}_t$ be a malicious prover for $\otimes[\mathsf{PCP}, t]$. We construct a malicious prover $\widetilde{\mathbf{P}}_{t+1}$ for $\otimes[\mathsf{PCP}, t+1]$.

$\widetilde{\mathbf{P}}_{t+1}$:
1. Compute $\widetilde{\Pi}_t \leftarrow \widetilde{\mathbf{P}}_t$.
2. Set $\widetilde{\Pi}_{t+1} \coloneqq (\sigma^{t+1})^{l^{t+1}}$, where σ is an arbitrary symbol in Σ.
3. For every $(q_1, \ldots, q_{t+1}) \in [l]^{t+1}$:
   (a) Let $(\mathsf{ans}_1, \ldots, \mathsf{ans}_t) \coloneqq \widetilde{\Pi}_t[(q_1, \ldots, q_t)]$.
   (b) Set $\widetilde{\Pi}_{t+1}[(q_1, \ldots, q_{t+1})] \coloneqq (\mathsf{ans}_1, \ldots, \mathsf{ans}_t, \mathsf{ans}_t)$.
4. Output $\widetilde{\Pi}_{t+1}$.

Consider $\boldsymbol{\rho} \in (\{0,1\}^{\mathsf{vr}})^t$ such that $\mathbf{V}_t^{\widetilde{\Pi}_t}(\phi; \boldsymbol{\rho}) = 1$. Since ϕ is symmetric, we know that for every $(\rho_1, \ldots, \rho_t, \rho_{t+1}) \in (\{0,1\}^{\mathsf{vr}})^{t+1}$ such that $(\rho_1, \ldots, \rho_t) = \boldsymbol{\rho}$, $\mathbf{V}_{t+1}^{\widetilde{\Pi}_{t+1}}(\phi; (\rho_1, \ldots, \rho_{t+1})) = 1$. Therefore, we conclude that

$$\beta_{t+1}(\phi) \geq \frac{\beta_t(\phi) \cdot (2^{\mathsf{vr}})^t \cdot 2^{\mathsf{vr}}}{(2^{\mathsf{vr}})^{t+1}} = \beta_t(\phi) \ .$$

□

Lemma B.6. *For every instance $\phi = ((M_i, f^*))_{i \in [m]} \notin \text{OCSPSAT}[\Sigma, l, q, m, f^*]$ where every M_i is in ascending order (with respect to the lexicographic order of $[l]$),*

$$\beta_{\mathsf{MIP}}(\phi) = 1 \implies \lim_{t \to \infty} \beta_t(\phi) = 1 \ .$$

Proof. Fix $\phi = ((M_i, f^*))_{i \in [m]} \notin \text{OCSPSAT}[\Sigma, l, q, m, f^*]$ where every M_i is in ascending order. Let $(\widetilde{\mathcal{P}}_i)_{i \in [\mathsf{q}]}$ be the malicious provers for the MIP projection that make the MIP projection verifier always accept. Let $\rho^* \in \{0,1\}^{\mathsf{vr}}$ be such that $(q_1^*, \ldots, q_q^*) \coloneqq \mathbf{V}_{\mathrm{q}}(\phi, \rho^*)$ is the smallest query list among all possible query lists of $\mathbf{V}$ (with respect to the global ordering of $[l]$), which is possible because queries of $\mathbf{V}$ is ordered.

We construct a malicious prover $\widetilde{\mathbf{P}}_t$ for $\otimes[\mathsf{PCP}, t]$ as follows.

$\widetilde{\mathbf{P}}_t$:
1. For every $i \in [q]$, compute $\mathsf{ans}_{q_i^*} \coloneqq \widetilde{\mathcal{P}}_i(q_i^*)$.
2. For every $(q_1, \ldots, q_t) \in [\mathrm{l}]^t$, set $\widetilde{\Pi}[(q_1, \ldots, q_t)] \coloneqq (\mathsf{ans}_q)^t$ where

$$q \coloneqq \min\left(\left(\{q_1, \ldots, q_t\} \cap \{q_1^*, \ldots, q_q^*\}\right) \cup \{q_q^*\}\right) \ .$$

3. Output $\widetilde{\Pi}$.

Consider an arbitrary randomness $\boldsymbol{\rho} \in (\{0,1\}^{\mathsf{vr}})^t$ for $\mathbf{V}_t$ such that $\rho^* \in \boldsymbol{\rho}$. For every $i \in [q]$, let $\mathbf{Q}_i \in [l]^t$ be the query lists of $\mathbf{V}_t(\phi; \boldsymbol{\rho})$. Since $(q_1^*, \ldots, q_q^*)$ is the smallest query set, for every $i \in [q]$,

$$\min\left(\left(\mathbf{Q}_i \cap (q_1^*, \ldots, q_q^*)\right) \cup \{q_q^*\}\right) = q_i^* \ .$$

Since the MIP projection provers make the MIP projection verifier accepts, $\mathbf{V}_t^{\widetilde{\Pi}}(\phi; \boldsymbol{\rho})$ accepts. Thus,

$$\lim_{t \to \infty} \beta_t(\mathbb{x}) \geq \lim_{t \to \infty} \left(1 - \left(\frac{2^{\mathsf{vr}} - 1}{2^{\mathsf{vr}}}\right)^t\right) = 1 \ .$$

□

Remark B.7. Let β be the soundness error of the PCP in Construction 7.5, β_{MIP} be the soundness error of the MIP projection of the PCP, and β_t be the soundness error of the parallel repetition of the PCP.

In Lemma 7.8, we show that, for every instance $\phi \notin \textsc{CspSat}[\Sigma, l, q, m, f^*]$, $\beta(\phi) > 0$ implies $\lim_{t\to\infty} \beta_t(\phi) > 0$. By Theorem 5.2, $\beta_{\mathsf{MIP}}(\phi) = 1$. Hence, for every $\phi \notin \textsc{OCspSat}[\Sigma, l, q, m, f^*]$ and $t \in \mathbb{N}$,

$$\beta(\phi) > 0 \implies \lim_{t\to\infty} \beta_t(\phi) = 1 \ .$$

B.3 Parallel repetition for non-symmetric ordered CSPs

Fix a finite set Σ, $l \in \mathbb{N}$, $q \in \mathbb{N}$, $m \in \mathbb{N}$, and function $f^* \colon \Sigma^q \to \{0,1\}$. Let PCP be the canonical PCP for $\textsc{OCspSat}[\Sigma, l, q, m]$ (as in Construction 7.5). Let $\beta_{\mathsf{MIP}}(\phi)$ be the soundness error for the MIP projection of PCP, and for every $t \in \mathbb{N}$ let $\beta_t(\phi)$ be the soundness error for $\otimes[\mathsf{PCP}, t]$.

Definition B.8. *An ordered (Σ, l, q, m)-CSP ϕ is a* **full ordered CSP** *if for every set $S \in \binom{[l]}{q}$ with an implicit ordering and permutation $p \colon [q] \to [q]$ there exists a constraint function $f \colon \Sigma^q \to \{0,1\}$ such that*

$$\Big(\big(p(S[1]), \ldots, p(S[q])\big), f\Big) \in \phi \ .$$

In particular, a full ordered CSP has $m = \binom{l}{q} \cdot q!$ constraints.

Lemma B.9. *Let $\phi \notin \textsc{OCspSat}[\Sigma, l, q, m]$ be a full ordered CSP. Then*

$$\beta_t(\phi) \le \beta(\phi) \ .$$

The ordered CSP constructed from IndSet in Remark B.4 is an example of the ordered CSP specified in Lemma B.9. Hence, for every $(G, k) \notin \textsc{IndSet}$ and $t \in \mathbb{N}$,

$$\beta_t((G,k)) = \beta((G,k)) \ .$$

Proof of Lemma B.9. Let $\otimes[\mathsf{PCP}, t] = (\mathbf{P}_t, \mathbf{V}_t)$ be the t-wise parallel repetition of PCP. We consider a directed hypergraph $G = (V, E)$ with multi-edges (E is a multiset) that captures the acceptance probability of $\mathbf{V}_t$. The vertices in G is the set of all t-wise queries:

$$V \coloneqq \{(q_1, \ldots, q_t) \in [l]^t\} \ .$$

To define the edge set E, we first consider the following set $\mathcal{C}$:

$$\mathcal{C} \coloneqq \{S = \{v_1, \ldots, v_l\} \in V^l : \forall\, i \neq j \in [l], \forall\, k \in [t], v_i[k] \neq v_j[k]\} \ .$$

For every set $S = \{v_1, \ldots, v_l\} \in \mathcal{C}$, we define the following set of edges

$$E_S \coloneqq \{(v_{i_1}, \ldots, v_{i_q}) \in S^q : i_1, \ldots, i_q \text{ are pairwise distinct elements in } [l]\} \ .$$

Our edge set E for the graph G is thus defined as follows:

$$E \coloneqq \bigcup_{S \in \mathcal{C}} E_S \ ,$$

where $\bigcup$ here is the multiset union operator.

Now we argue that the graph G indeed captures the acceptance probability of $\mathbf{V}_t$. In particular, we show that for every $S \in \mathcal{C}$, at most $\beta(\phi) \cdot m$ many edges in E_S correspond to an accepting randomness (note that every vertex $v \in V$ correspond to some randomness because of our assumption on ϕ). We first see how this claim leads to an upper bound on the soundness error for $\otimes[\mathsf{PCP}, t]$ and then prove this claim.

Observe that the set $\mathcal{C}$ defined above is equivalent to the following set

$$\left\{ \begin{array}{l} S \in V^l : S = \mathsf{set}\left((r_1, \ldots, r_t)^T\right) \text{ such that } (r_1 = [l]) \\ \wedge \left(\forall\, i \in [2,t], (r_i \in [l]^l) \wedge (\exists \text{ a permutation } p_i \colon [l] \to [l], \forall j \in [l], r_i[j] = p_i[j])\right) \end{array} \right\} .$$

Hence,

$$|\mathcal{C}| = (l!)^{t-1} \ ,$$

which implies

$$|E| = (l!)^{t-1} \cdot \binom{l}{q} \ .$$

Let $\widetilde{\Pi}$ be an arbitrary PCP proof for $\otimes[\mathsf{PCP}, t]$. We define the winning set $W_{\widetilde{\Pi},S}$ of $\widetilde{\Pi}$ on S as follows:

$$W_{\widetilde{\Pi},S} \coloneqq \{\boldsymbol{\rho} \in (\{0,1\}^{\mathsf{vr}})^t : \mathbf{Q} \subseteq S \text{ where } \mathbf{Q} \coloneqq \mathbf{V}_t(\phi; \boldsymbol{\rho}) \wedge \mathbf{V}_t^{\widetilde{\Pi}}(\phi; \boldsymbol{\rho}) = 1)\} \ .$$

We show that $\Pr\left[\mathbf{V}_t^{\widetilde{\Pi}}(\phi) = 1\right] = \frac{\sum_{S \in \mathcal{C}} \left|W_{\widetilde{\Pi},S}\right|}{|E|}$. First note that for every $S \in \mathcal{C}$, and for every $e \in E_S$, $e \in S^q \subseteq V^q$ is a valid query list for $\mathbf{V}_t$ by definition. On the other hand, consider an arbitrary randomness $\boldsymbol{\rho} \in (\{0,1\}^{\mathsf{vr}})^t$ and let $\mathbf{Q} \in ([l]^t)^q$ be the corresponding query list of $\mathbf{V}_t(\phi; \boldsymbol{\rho})$. We know that for every $i \neq j \in [q]$ and $k \in [t]$, $\mathbf{Q}[i][k] \neq \mathbf{Q}[j][k]$. Therefore, there is some $S \in \mathcal{C}$ such that $\{\mathbf{Q}[1], \ldots, \mathbf{Q}[q]\} \subseteq S$, which implies that $\mathbf{Q} \in E_S$. For every $e \in E$, let m_e be the multiplicity of e in E. We observe that $m_e = ((l-q)!)^{t-1}$ for every $e \in E$, which implies that

$$\frac{\sum_{S \in \mathcal{C}} \left|W_{\widetilde{\Pi},S}\right|}{|E|} = \frac{m_e \cdot \sum_{\boldsymbol{\rho} \in (\{0,1\}^{\mathsf{vr}})^t} \mathbb{1}_{\mathbf{V}_t^{\widetilde{\Pi}}(\phi;\boldsymbol{\rho})=1}}{m_e \cdot (2^{\mathsf{vr}})^t} = \Pr\left[\mathbf{V}_t^{\widetilde{\Pi}}(\phi) = 1\right] \ .$$

By the claim above, we conclude

$$\beta_t(\phi) = \max_{\widetilde{\Pi}} \Pr\left[\mathbf{V}_t^{\widetilde{\Pi}}(\phi) = 1\right] = \max_{\widetilde{\Pi}} \frac{\sum_{S \in \mathcal{C}} \left|W_{\widetilde{\Pi},S}\right|}{|E|} \leq \frac{\sum_{S \in \mathcal{C}} \beta(\phi) \cdot m}{|E|} = \frac{(l!)^{t-1} \cdot \beta(\phi) \cdot \frac{l!}{(l-q)!}}{(l!)^{t-1} \cdot \frac{l!}{(l-q)!}} = \beta(\phi)$$

as desired.

We are left to argue that for every $S = (v_1, \ldots, v_l) \in \mathcal{C}$, at most $\beta(\phi) \cdot m$ many edges in E_S correspond to an accepting randomness. In other words, $\left|W_{\widetilde{\Pi},S}\right| \leq \beta(\phi) \cdot m$. We define a malicious prover for PCP as follows.

$\widetilde{\mathbf{P}}$:

1. For every $i \in [l]$, compute the answer $\mathsf{ans}_i \coloneqq \widetilde{\Pi}[v_i]$.
2. For every $i \in [l]$, set $\widetilde{\pi}[i] \coloneqq \mathsf{ans}_i[1]$.
3. Output $\widetilde{\pi}$.

Let $\widetilde{\pi} \coloneqq \widetilde{\mathbf{P}}$. Note that for every $(\rho_1, \ldots, \rho_t) \in (\{0,1\}^{\mathsf{vr}})^t$ such that $\mathbf{V}_t^{\widetilde{\Pi}}(\phi; (\rho_1, \ldots, \rho_t)) = 1$, we get

$\mathbf{V}^{\widetilde{\pi}}(\phi;\rho_1)=1$. Thus,

$$\left|W_{\widetilde{\Pi},S}\right| \leq \beta(\phi)\cdot 2^{\mathsf{vr}} = \beta(\mathbb{x})\cdot m \ .$$

□

B.4 More on tightness of the characterization

Corollary B.10. *Let* $m \coloneqq \binom{l}{q}\cdot q!$. *Let* $\phi \notin \text{OCSPSAT}[\Sigma,l,q,m,f^*]$ *be such that for every* $M \in \binom{[l]}{q}$ *it holds that* $(M,f^*)\in\phi$. *For every* $t\in\mathbb{N}$,

$$\beta_t(\phi)=\beta(\phi) \ .$$

Proof. Combining Lemma B.5 and Lemma B.9 gives the desired result. □

Remark B.11. We can generalize Lemma A.1 using Corollary B.10. We show that there exists a PCP for a language L such that for every $c\in\mathbb{N}$ there exist infinitely many instances $\mathbb{x}\notin L$ such that for every $t\in\mathbb{N}$

$$\beta_t(\mathbb{x})=\frac{c}{2^{\mathsf{vr}}} \ .$$

Consider the language

$$\text{OCSPSAT} \coloneqq \bigcup_{\text{alphabet }\Sigma, l\in\mathbb{N}, q\in\mathbb{N}, m\in\mathbb{N}} \text{OCSPSAT}[\Sigma,l,q,m] \ .$$

Let PCP be the PCP in Construction 7.5. Fix $c\in\mathbb{N}$ and consider an arbitrary $l\in\mathbb{N}$ such that $m\coloneqq l!\geq c$.

Let $\mathcal{M}$ be the set of all lists of size l with alphabet $[l]$; note that $|\mathcal{M}|=l!$. Let $f\colon\Sigma^q\to\{0,1\}$ be such that for every $(a_1,\ldots,a_q)\in[l]^q$, $f(a_1,\ldots,a_q)$ outputs 1 if $(a_1,\ldots,a_q)$ is one of first c elements in $\mathcal{M}$ and 0 otherwise (assume $\mathcal{M}$ has an implicit ordering). We define

$$\phi_{c,l}\coloneqq (M,f)_{M\in\mathcal{M}} \ .$$

Hence, $\phi_{c,l}$ is an ordered $([l],l,l,m,f)$-CSP with $\text{val}(\phi_{c,l})=c/m$, which implies $\beta(\phi_{c,l})=c/m$ by Lemma 7.6. We conclude that $\beta_t(\phi_{c,l})=c/m$ for every $t\in\mathbb{N}$ from Corollary B.10.

Remark B.12. There exists a PCP for a language L such that for every rational number $\frac{a}{b}\in[0,1]$ there exist infinitely many instances $\mathbb{x}\notin L$ such that for every $t\in\mathbb{N}$

$$\beta_t(\mathbb{x})=\frac{a}{b} \ .$$

Let $c\in\mathbb{N}$ be such that $b\cdot c=l!$ for some $l\in\mathbb{N}$. We can construct infinitely many $\phi\notin\text{OCSPSAT}$ such that $\beta_t(\phi)=c'/l!=a/b$ where $c'\coloneqq a\cdot c$ according to the argument in Remark B.11.

Theorem 5.2 shows, for every PCP for a language L and $\mathbb{x}\notin L$, if the MIP projection has soundness error less than 1, then $\lim_{t\to\infty}\beta_t(\mathbb{x})\in[\frac{1}{2^{\mathsf{vr}}},1]$. The above observation complements Theorem 5.2 by showing that there exists a PCP for a language L such that, for every rational number $\frac{a}{b}$, there exists infinitely many instances $\mathbb{x}\notin L$ whose corresponding PCP verifier has randomness complexity vr such that

$$\frac{a}{b}\in\left[\frac{1}{\mathsf{vr}},1\right] \text{ and } \lim_{t\to\infty}\beta_t(\mathbb{x})=\frac{a}{b} \ .$$

C Proof of Lemma 8.3

We restate Lemma 8.3 for convenience.

Lemma 8.3. *For every alphabet Σ and integers $n \in \mathbb{N}$ and $m \leq |\Sigma|$,*

$$\mathcal{K}(\Sigma, n, m) = \sum_{i=1}^{m} \binom{|\Sigma|}{i} \sum_{j=1}^{i} \binom{i}{j} \cdot (-1)^{i-j} \cdot j^n \leq \binom{|\Sigma|}{m} \cdot m^n \ ,$$

in which the upper bound is attained if and only if $m = 1$.

Proof. We fix $n \in \mathbb{N}$ throughout the proof.

We prove the statement for the exact value of $\mathcal{K}(\Sigma, n, m)$ by strong induction on the alphabet size. *Base case:* $|\Sigma| = 1$. For every alphabet Σ with a single letter and $m = 1$,

$$\mathcal{K}(\Sigma, n, m) = 1 = \sum_{i=1}^{m} \binom{|\Sigma|}{i} \sum_{j=1}^{i} \binom{i}{j} \cdot (-1)^{i-j} \cdot j^n \ .$$

Inductive step: $|\Sigma| > 1$. Let Σ be an arbitrary finite set such that $|\Sigma| > 1$. Assume that for every $\Sigma' \subsetneq \Sigma$ and $m \leq |\Sigma'|$,

$$\mathcal{K}(\Sigma', n, m) = \sum_{i=1}^{m} \binom{|\Sigma'|}{i} \sum_{j=1}^{i} \binom{i}{j} \cdot (-1)^{i-j} \cdot j^n \ .$$

By Definition 8.2, for every Σ and $m \leq |\Sigma|$, $\mathcal{K}(\Sigma, n, m) = |\mathcal{S}(\Sigma, n, m)|$ where $\mathcal{S}(\Sigma, n, m) \coloneqq \{s = (s_1, \ldots, s_n) \in \Sigma^n : |\{s_1, \ldots, s_n\}| \leq m\}$. For every $m \leq |\Sigma|$, we can reformulate $\mathcal{S}(\Sigma, n, m)$ as follows:

$$\begin{aligned}
\mathcal{S}(\Sigma, n, m) &= \{s = (s_1, \ldots, s_n) \in \Sigma^n : |\{s_1, \ldots, s_n\}| \leq m\} \\
&= \bigcup_{m'=1}^{m} \{s = (s_1, \ldots, s_n) \in \Sigma^n : |\{s_1, \ldots, s_n\}| = m'\} \\
&= \bigcup_{m'=1}^{m} \{s = (s_1, \ldots, s_n) \in \Sigma^n : \exists\, \Sigma' \subseteq \Sigma \text{ s.t. } |\{s_1, \ldots, s_n\}| = \left|\Sigma'\right| = m' \wedge s \in (\Sigma')^n\} \\
&= \bigcup_{m'=1}^{m} \bigcup_{\Sigma' \subseteq \Sigma : |\Sigma'| = m'} \{s = (s_1, \ldots, s_n) \in \Sigma^n : |\{s_1, \ldots, s_n\}| = m' \wedge s \in (\Sigma')^n\} \\
&= \bigcup_{m'=1}^{m} \bigcup_{\Sigma' \subseteq \Sigma : |\Sigma'| = m'} \{s = (s_1, \ldots, s_n) \in (\Sigma')^n : |\{s_1, \ldots, s_n\}| = m'\} \\
&= \bigcup_{m'=1}^{m} \bigcup_{\Sigma' \subseteq \Sigma : |\Sigma'| = m'} \left(\mathcal{S}(\Sigma', n, m') \setminus \mathcal{S}(\Sigma', n, m' - 1)\right) \ .
\end{aligned}$$

Above, unions are over pairwise disjoint sets because for every $s \in \Sigma^n$ there is only one $\Sigma' \subseteq \Sigma$ such that $|\Sigma'| = m'$ and $s \in (\Sigma')^n$ where $m' = |\{s_1, \ldots, s_n\}|$. By the induction hypothesis, we deduce the desired

formula:

$$\begin{aligned}
\mathcal{K}(\Sigma,n,m) &= \sum_{m'=1}^{m} \sum_{\Sigma'\subseteq\Sigma:|\Sigma'|=m'} \big(\mathcal{K}(\Sigma',n,m') - \mathcal{K}(\Sigma',n,m'-1)\big) \\
&= \sum_{m'=1}^{m} \sum_{\Sigma'\subseteq\Sigma:|\Sigma'|=m'} \left(\sum_{i=1}^{m'} \binom{m'}{i} \sum_{j=1}^{i} \binom{i}{j} \cdot (-1)^{i-j} \cdot j^n - \sum_{i=1}^{m'-1} \binom{m'}{i} \sum_{j=1}^{i} \binom{i}{j} \cdot (-1)^{i-j} \cdot j^n \right) \\
&= \sum_{m'=1}^{m} \sum_{\Sigma'\subseteq\Sigma:|\Sigma'|=m'} \binom{m'}{m'} \sum_{j=1}^{m'} \binom{m'}{j} \cdot (-1)^{m'-j} \cdot j^n \\
&= \sum_{m'=1}^{m} \binom{|\Sigma|}{m'} \binom{m'}{m'} \sum_{j=1}^{m'} \binom{m'}{j} \cdot (-1)^{m'-j} \cdot j^n \\
&= \sum_{i=1}^{m} \binom{|\Sigma|}{i} \sum_{j=1}^{i} \binom{i}{j} \cdot (-1)^{i-j} \cdot j^n \ .
\end{aligned}$$

Next, we prove the upper bound. Similar to before, we can reformulate $\mathcal{S}(\Sigma,n,m)$ as follows:

$$\begin{aligned}
\mathcal{S}(\Sigma,n,m) &= \{s \in \Sigma^n : \exists\, \Sigma' \subseteq \Sigma \text{ s.t. } \left|\Sigma'\right| = m, s \in (\Sigma')^n\} \\
&= \bigcup_{\Sigma'\subseteq\Sigma \text{ s.t. } |\Sigma'|=m} \{s \in \Sigma^n : \ s \in (\Sigma')^n\} \\
&= \bigcup_{\Sigma'\subseteq\Sigma \text{ s.t. } |\Sigma'|=m} (\Sigma')^n \ .
\end{aligned}$$

Therefore,

$$\mathcal{K}(\Sigma,n,m) \leq \sum_{\Sigma'\subseteq\Sigma \text{ s.t. } |\Sigma'|=m} \left|(\Sigma')^n\right| = \binom{|\Sigma|}{m} \cdot m^n \ .$$

Note that $\{\Sigma' \subseteq \Sigma \text{ s.t. } |\Sigma'| = m\}$ is a set of pairwise disjoint sets if and only if $m = 1$. Hence,

$$\mathcal{K}(\Sigma,n,1) = \sum_{\Sigma'\subseteq\Sigma \text{ s.t. } |\Sigma'|=m} \left|(\Sigma')^n\right| = |\Sigma|$$

and, for every $m > 1$,

$$\mathcal{K}(\Sigma,n,m) < \sum_{\{\Sigma'\subseteq\Sigma \text{ s.t. } |\Sigma'|=m\}} \left|(\Sigma')^n\right| = \binom{|\Sigma|}{m} \cdot m^n \ .$$

□

D A minimal counterexample

The basic phenomenon that makes parallel repetition for PCPs fail can be illustrated via a simple PCP for the empty language. (In contrast, Theorem 4.1 considers a natural PCP for an NP-complete language.)

Consider a PCP verifier $\mathbf{V}$ that, given access to a PCP string $\widetilde{\pi}\colon [3] \to \{0,1\}$ (a 3-bit PCP string), samples two queries $(q_1, q_2) \leftarrow \{(i,j) \in [3]^2 : i < j\}$ and checks that $\widetilde{\pi}[q_1] \neq \widetilde{\pi}[q_2]$. Observe that for every $\widetilde{\pi}\colon [3] \to \{0,1\}$ the probability that $\mathbf{V}$ accepts is at most $\frac{2}{3}$. In other words, the soundness error β is $\frac{2}{3} < 1$.

The t-wise parallel repetition of this PCP works as follows. A PCP string is a function $\widetilde{\Pi}\colon [3]^t \to \{0,1\}^t$. The PCP verifier $\mathbf{V}_t$ independently samples t query pairs $(q_{1,1}, q_{1,2}), \ldots, (q_{t,1}, q_{t,2})$ (each pair like $\mathbf{V}$ would); queries $\widetilde{\Pi}$ at $(q_{1,1}, \ldots, q_{t,1})$ and $(q_{1,2}, \ldots, q_{t,2})$; and accepts if

$$\forall\, i \in [t]\,,\ \widetilde{\Pi}[(q_{1,1}, \ldots, q_{t,1})][i] \neq \widetilde{\Pi}[(q_{1,2}, \ldots, q_{t,2})][i] \ .$$

Since $q_{i,1} < q_{i,2}$ for every $i \in [t]$, $(q_{1,1}, \ldots, q_{t,1})$ and $(q_{1,2}, \ldots, q_{t,2})$ become more and more distinguishable as t increases. Therefore, it is possible to construct $\widetilde{\Pi}$ such that $\widetilde{\Pi}[(q_{1,1}, \ldots, q_{t,1})] = 0^t$ with high probability and $\widetilde{\Pi}[(q_{1,2}, \ldots, q_{t,2})] = 1^t$ with high probability, which convinces $\mathbf{V}_t$ with probability approaching 1.

For example, consider $\widetilde{\Pi}$ such that $\widetilde{\Pi}[(q_1, \ldots, q_t)] = 0^t$ if $1 \in (q_1, \ldots, q_t)$ and $\widetilde{\Pi}[(q_1, \ldots, q_t)] = 1^t$ otherwise. We want to lower bound the probability that $\mathbf{V}_t$ accepts $\widetilde{\Pi}$. Recall that $\mathbf{V}_t$ rejects if there is some $i \in [t]$ such that $\widetilde{\Pi}[(q_{1,1}, \ldots, q_{t,1})][i] = \widetilde{\Pi}[(q_{1,2}, \ldots, q_{t,2})][i]$. Since the PCP string $\widetilde{\Pi}$ that we constructed answers always 0^t or 1^t, $\mathbf{V}_t^{\widetilde{\Pi}}$ rejects if and only if the answers to queries $(q_{1,1}, \ldots, q_{t,1})$ and $(q_{1,2}, \ldots, q_{t,2})$ are both 0^t or both 1^t. Since $q_{i,1} < q_{i,2}$ for every $i \in [t]$, $1 \notin (q_{1,2}, \ldots, q_{t,2})$, which implies that the answer to the second query is 1^t. Thus, $\mathbf{V}_t^{\widetilde{\Pi}}$ rejects if and only if the answers to both queries are 1^t, which happens if and only if $1 \notin (q_{1,1}, \ldots, q_{t,1})$. Therefore,

$$\Pr\left[\mathbf{V}_t^{\widetilde{\Pi}} = 1\right] = 1 - \Pr\left[1 \notin (q_{1,1}, \ldots, q_{t,1})\right] \geq 1 - \left(\frac{2}{3}\right)^t \to 1 \text{ as } t \to \infty \ .$$

Parallel repetition does not work even if we modify $\mathbf{V}$ to sample random queries q_1 and q_2 such that $q_1 \neq q_2$ (without enforcing that $q_1 < q_2$). Now $(q_{1,1}, \ldots, q_{t,1})$ and $(q_{1,2}, \ldots, q_{t,2})$ are identically distributed, but it is still possible to construct a PCP string $\widetilde{\Pi}\colon [3]^t \to \{0,1\}^t$ that consists of only 0^t and 1^t such that parallel repetition cannot reduce soundness error below some non-zero constant (independent of t). We give an example. Note that the (modified) PCP verifier $\mathbf{V}$ accepts $\widetilde{\pi} \coloneqq (0,0,1)$ if it queries $(2,3)$. Now consider the PCP string $\widetilde{\Pi}\colon [3]^t \to \{0,1\}^t$ such that

$$\widetilde{\Pi}[(q_1, \ldots, q_t)] \coloneqq \begin{cases} 0^t & \text{if } q_t = 2 \\ 1^t & \text{if } q_t = 3 \\ * & \text{otherwise (if } q_t = 1) \end{cases} \ .$$

The "$*$" values in $\widetilde{\Pi}$ can be an arbitrary symbol in $\{0,1\}^t$. Consider the following set of query pairs of $\mathbf{V}_t$:

$$S \coloneqq \{((q_{1,1}, q_{1,2}), \ldots, (q_{t,1}, q_{t,2})) : (q_{t,1}, q_{t,2}) = (2,3)\} \ .$$

By construction of $\widetilde{\Pi}$, $\mathbf{V}_t^{\widetilde{\Pi}}$ accepts if the two sampled queries are in S. We conclude that

$$\Pr\left[\mathbf{V}_t^{\widetilde{\Pi}} = 1\right] \geq \frac{|S|}{3^t} = \frac{3^{t-1}}{3^t} = \frac{1}{3} \ .$$

Acknowledgments

The authors are partially supported by the Ethereum Foundation. 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 2024 for valuable comments and suggestions; in particular, they suggested the simple counterexample in Appendix D.

References

[BRRRS09] 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. 2009, pp. 352–365.

[DG08] 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. 2008, pp. 613–622.

[DM11] Irit Dinur and Or Meir. "Derandomized Parallel Repetition via Structured PCPs". In: *Computational Complexity* 20.2 (2011), pp. 207–327.

[DN17] 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. 2017, 29:1–29:50.

[DR04] 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. 2004, pp. 155–164.

[DS14] Irit Dinur and David Steurer. "Direct Product Testing". In: *Proceedings of the 29th Annual IEEE Conference on Computational Complexity*. CCC '14. 2014, pp. 188–196.

[FRS88] Lance Fortnow, John Rompel, and Michael Sipser. "On the Power of Multi-Prover Interactive Protocols". In: *Theoretical Computer Science*. 1988, pp. 156–161.

[FV02] Uriel Feige and Oleg Verbitsky. "Error Reduction by Parallel Repetition – A Negative Result". In: *Combinatorica* 22 (2002), pp. 461–478.

[Fei98] Uriel Feige. "A Threshold of Ln n for Approximating Set Cover". In: *Journal of the ACM* 45 (1998), pp. 634–652.

[GS97] 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. 1997, pp. 67–84.

[Gol98] Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*. Vol. 17. Algorithms and Combinatorics. Springer, 1998.

[Hol07] Thomas Holenstein. "Parallel repetition: simplifications and the no-signaling case". In: *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*. STOC '07. 2007, pp. 411–419.

[IKW12] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. "New Direct-Product Testers and 2-Query PCPs". In: *SIAM Journal on Computing* 41 (2012), pp. 1722–1768.

[RR12] 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. 2012, pp. 247–257.

[Rao11] Anup Rao. “Parallel Repetition in Projection Games and a Concentration Bound”. In: *SIAM Journal on Computing* 40 (2011), pp. 1871–1891.

[Raz11] Ran Raz. “A Counterexample to Strong Parallel Repetition”. In: *SIAM Journal on Computing* 40 (2011), pp. 771–777.

[Raz95] Ran Raz. “A parallel repetition theorem”. In: *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*. STOC ’95. 1995, pp. 447–456.

[Ver96] Oleg Verbitsky. “Towards the parallel repetition conjecture”. In: *Theoretical Computer Science* 157 (1996), pp. 277–282.