Yet Another Simple Proof of the PCRP Theorem

Naoto Ohsaka
CyberAgent, Inc., Tokyo, Japan

Abstract

The *Probabilistically Checkable Reconfiguration Proof* (PCRP) theorem, proven by Hirahara and Ohsaka (STOC 2024) [22] and Karthik C. S. and Manurangsi [29], provides a new PCP-type characterization of **PSPACE**: A language L is in **PSPACE** if and only if there exists a probabilistic verifier $\mathcal{V}$ and a pair of polynomial-time computable proofs $\pi^{\mathsf{ini}}, \pi^{\mathsf{end}}$ such that the following hold for every input x:

- If $x \in L$, then $\pi^{\mathsf{ini}}(x)$ can be transformed into $\pi^{\mathsf{end}}(x)$ by repeatedly flipping a single bit of the proof at a time, while making $\mathcal{V}(x)$ to accept every intermediate proof with probability 1.
- If $x \notin L$, then any such transformation induces a proof that is rejected by $\mathcal{V}(x)$ with probability more than $\frac{1}{2}$.

The PCRP theorem finds many applications in **PSPACE**-hardness of approximation for reconfiguration problems.

In this paper, we present an alternative proof of the PCRP theorem that is "simpler" than those of Hirahara and Ohsaka [22] and Karthik C. S. and Manurangsi [29]. Our PCRP system is obtained by combining simple *robustization* and *composition* steps in a modular fashion, which renders its analysis more intuitive. The crux of implementing the robustization step is an error-correcting code that enjoys both *list decodability* and *reconfigurability*, the latter of which enables to reconfigure between a pair of codewords, while avoiding getting too close to any other codewords.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Error-correcting codes

Keywords and phrases reconfiguration problems, hardness of approximation, probabilistic proof systems

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.122

Category Track A: Algorithms, Complexity and Games

1 Introduction

Given a combinatorial problem and a pair of its feasible solutions, can we find a "step-by-step" transformation from one to the other that maintains the feasibility at every intermediate state? *Combinatorial reconfiguration* is a brand-new field in theoretical computer science, aimed at studying connectivity problems over the solution space of a combinatorial problem, called the *source problem*. One canonical reconfiguration problem is 3-SAT Reconfiguration [15], whose source problem is 3-SAT: For a satisfiable 3-CNF formula φ and a pair of its satisfying assignments $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}}$, we shall transform σ^{ini} into σ^{end} by repeatedly flipping a single variable assignment at a time, while always preserving that φ is satisfied. Formally, a *reconfiguration sequence* from σ^{ini} to σ^{end} is a sequence of assignments to φ, denoted by $\vec{\sigma} = (\sigma^{(1)}, \dots, \sigma^{(T)})$, such that $\sigma^{(1)} = \sigma^{\mathsf{ini}}$, $\sigma^{(T)} = \sigma^{\mathsf{end}}$, and $\sigma^{(t)}$ and $\sigma^{(t+1)}$ differ in at most one variable.

3-SAT Reconfiguration

Input:	a 3-CNF formula φ and a pair of its satisfying assignments $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}}$.
Output:	is there a reconfiguration sequence from σ^{ini} to σ^{end} consisting only of satisfying assignments?

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).
Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis
Article No. 122; pp. 122:1–122:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Since Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [27] initiated the unified framework for reconfiguration, many reconfiguration problems have been formulated, including those of Boolean satisfiability, constraint satisfaction problems, and graph problems.

Of particular interest has been to elucidate their computational complexity. On the hardness side, a reconfiguration problem generally becomes **PSPACE**-complete if its source problem is intractable (say, **NP**-complete), e.g., reconfiguration problems of 3-SAT [15], 4-Coloring [8], and Independent Set [19, 20]. On the tractable side, a source problem in **P** frequently leads to a reconfiguration problem that also belongs to **P**, e.g., 2-SAT [15], Matching [27], and Spanning Tree [27]. Some exceptions are known, such as 3-Coloring [10] and Shortest Path [7]. Having established the complexity results of popular reconfiguration problems, researchers have turned their attention to parameterized complexity [5, 6, 9, 31] and restricted graph classes [18, 28, 30, 43]. We refer the readers to the surveys by Bousquet, Mouawad, Nishimura, and Siebertz [9], Mynhardt and Nasserasr [32], Nishimura [33], and van den Heuvel [42] as well as the Combinatorial Reconfiguration wiki [24] for more hardness and algorithmic results.

In this paper, we consider *approximability* of reconfiguration problems, which has been recently studied from both hardness and algorithmic sides, e.g., [21, 22, 23, 29, 34, 35, 36, 37, 38, 39]. For a reconfiguration problem, its *approximate version* affords to relax the feasibility of intermediate solutions, but requires to optimize the "worst" feasibility during reconfiguration. For example, in Maxmin 3-SAT Reconfiguration [27] – an approximate version of 3-SAT Reconfiguration – we are allowed to use any *non-satisfying* assignments to produce a reconfiguration sequence, but are required to maximize the *minimum* fraction of satisfied clauses.

Maxmin 3-SAT Reconfiguration

Input:	a 3-CNF formula φ and a pair of its satisfying assignments $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}}$.
Output:	a reconfiguration sequence $\vec{\sigma}$ from σ^{ini} to σ^{end}.
Goal:	maximize the minimum fraction of satisfied clauses of φ, where the minimum is taken over all assignments of $\vec{\sigma}$.

Solving Maxmin 3-SAT Reconfiguration approximately, we may be able to find a "reasonable" reconfiguration sequence consisting of almost-satisfying assignments, so that we can mange No instances of 3-SAT Reconfiguration.

Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [27, Theorems 4 and 5] showed that Maxmin SAT Reconfiguration and Maxmin Clique Reconfiguration are **NP**-hard to approximate. Their proofs do not bring **PSPACE**-hardness because of relying on the **NP**-hardness of approximating the corresponding source problems (i.e., Max SAT [26] and Max Clique [25]). Since many reconfiguration problems are **PSPACE**-complete, **NP**-hardness results seem not optimal. In fact, [27] posed **PSPACE**-hardness of approximation for reconfiguration problems as an open problem. The significance of showing **PSPACE**-hardness compared to **NP**-hardness is that it not only rules out polynomial-time algorithms under $\mathbf{P} \neq \mathbf{PSPACE}$, but further disproves the existence of a witness (especially a reconfiguration sequence) of polynomial length under $\mathbf{NP} \neq \mathbf{PSPACE}$.

1.1 PCRP Theorem, Its Consequences, and Our Contribution

Recently, Hirahara and Ohsaka [22] and Karthik C. S. and Manurangsi [29] established a reconfiguration analogue of the PCP theorem [2, 3], a.k.a. the *Probabilistically Checkable Reconfiguration Proof* (PCRP) theorem, which provides a new PCP-type characterization of

PSPACE. For a pair of proofs $\pi^{\mathsf{ini}}, \pi^{\mathsf{end}} \in \{0,1\}^n$, a *reconfiguration sequence* from π^{ini} to π^{end} is a sequence $\vec{\pi} = (\pi^{(1)}, \dots, \pi^{(T)})$ over $\{0,1\}^n$ such that $\pi^{(1)} = \pi^{\mathsf{ini}}$, $\pi^{(T)} = \pi^{\mathsf{end}}$, and $\pi^{(t)}$ and $\pi^{(t+1)}$ differ in at most one bit for every t.

▶ **Theorem 1.1** (PCRP theorem [22, 29]; see Theorem 3.1 for the formal statement)**.** *A language $L \subseteq \{0,1\}^*$ is in* **PSPACE** *if and only if there exists a probabilistic verifier $\mathcal{V}$ with randomness complexity $\mathcal{O}(\log n)$ and query complexity $\mathcal{O}(1)$ and a pair of polynomial-time computable proofs $\pi^{\mathsf{ini}}, \pi^{\mathsf{end}} \colon \{0,1\}^* \to \{0,1\}^*$ such that the following hold for every input $x \in \{0,1\}^*$:*

- *(Completeness) If $x \in L$, then there exists a reconfiguration sequence $\vec{\pi}$ from $\pi^{\mathsf{ini}}(x)$ to $\pi^{\mathsf{end}}(x)$ such that $\mathcal{V}(x)$ accepts every proof in $\vec{\pi}$ with probability 1.*
- *(Soundness) If $x \notin L$, then every reconfiguration sequence $\vec{\pi}$ from $\pi^{\mathsf{ini}}(x)$ to $\pi^{\mathsf{end}}(x)$ contains some proof that is rejected by $\mathcal{V}(x)$ with probability more than $\frac{1}{2}$.*

The PCRP theorem, along with a series of gap-preserving reductions [21, 22, 23, 29, 34, 35, 36], implies that several reconfiguration problems are **PSPACE**-hard to approximate, thereby resolving the open problem due to Ito, Demaine, Harvey, Papadimitriou, Sideri, Uehara, and Uno [27] affirmatively.

▶ **Corollary 1.2** (from Theorem 1.1 and [21, 22, 23, 29, 34, 35, 36])**.** *There exists a universal constant $\varepsilon \in (0,1)$ such that approximate versions of the following reconfiguration problems are* **PSPACE***-hard to approximate within a factor of $1 \pm \varepsilon$:*

3-SAT Reconfiguration, 2-CSP Reconfiguration, Independent Set Reconfiguration, Vertex Cover Reconfiguration, Dominating Set Reconfiguration, Set Cover Reconfiguration, Max Cut Reconfiguration, and Nondeterministic Constraint Logic.

In this paper, we present an alternative proof of the PCRP theorem that is "simpler" than those of Hirahara and Ohsaka [22] and Karthik C. S. and Manurangsi [29]. Our technical contribution is to develop a reconfiguration analogue of **robustization** [4, 12] and **composition** [2, 3]. Below, we review existing proofs of the PCRP theorem [22, 29], followed by our proof and techniques.

1.2 Recap of Two Proofs due to [22, 29]

We recapitulate two proofs of the PCRP theorem due to Hirahara and Ohsaka [22] and Karthik C. S. and Manurangsi [29]. Let $L \subseteq \{0,1\}^*$ be a **PSPACE**-complete language, say, 3-SAT Reconfiguration,[1] for which we will design a PCRP system. Suppose that we are given a satisfiable 3-CNF formula φ over n variables and a pair of its satisfying assignments $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}} \in \{0,1\}^n$. Both [22, 29] employ the following common strategy:

- Encode each assignment to φ using an *error-correcting code* $\mathsf{Enc} \colon \{0,1\}^n \to \{0,1\}^{p(n)}$.
- Check if an input string $f \in \{0,1\}^{p(n)}$ is the encoded satisfying assignment to φ (i.e., $f = \mathsf{Enc}(\sigma)$ for some assignment $\sigma \in \{0,1\}^n$), by running the *assignment tester* [12] (a.k.a. *PCP of proximity* [4]) on f along with an auxiliary proof $\pi \in \{0,1\}^*$.

Ideally, the existence of a reconfiguration sequence $\vec{\sigma}$ from σ^{ini} to σ^{end} consisting of satisfying assignments to φ should imply the existence of a reconfiguration sequence $\vec{\Pi}$ from $\mathsf{Enc}(\sigma^{\mathsf{ini}}) \circ \pi^{\mathsf{ini}}$ to $\mathsf{Enc}(\sigma^{\mathsf{end}}) \circ \pi^{\mathsf{end}}$ consisting only of accepting proofs, where π^{ini} and π^{end}

[1] Hirahara and Ohsaka [22] adopted Succinct Reach and Karthik C. S. and Manurangsi [29] adopted 2-CSP Reconfiguration, but the choice of **PSPACE**-complete languages does not so matter.

are auxiliary proofs accepted by the assignment tester along with $\mathsf{Enc}(\sigma^{\mathsf{ini}})$ and $\mathsf{Enc}(\sigma^{\mathsf{end}})$, respectively. On the other hand, if there is a reconfiguration sequence $\vec{\Pi}$ from $\mathsf{Enc}(\sigma^{\mathsf{ini}}) \circ \pi^{\mathsf{ini}}$ to $\mathsf{Enc}(\sigma^{\mathsf{end}}) \circ \pi^{\mathsf{end}}$ in which every proof is accepted with a high probability (say, 0.99), then $(\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}}; \varphi)$ must be a YES instance of 3-SAT Reconfiguration. The challenge for the former requirement (i.e., completeness) is that every reconfiguration sequence between different codewords *must* induce a string that is "far" from every codeword, which would be rejected by the assignment tester, whereas for the latter requirement (i.e., soundness) is that we should be able to extract from $\vec{\Pi}$ a reconfiguration sequence $\vec{\sigma}$ consisting of satisfying assignments to φ. Hirahara and Ohsaka [22] and Karthik C. S. and Manurangsi [29] took slightly different approaches to these challenges.

Hirahara–Ohsaka's System

Hirahara and Ohsaka [22] first create a *pair* of fresh copies of the input string, denoted by f and g, so that one of them can be arbitrarily edited as long as the other is the encoded satisfying assignment. Introduce then a special symbol $\perp$ to indicate that a particular copy is in the "in transition" state. The verifier $\mathcal{V}_{\mathrm{HO}}$ of [22] is given oracle access to a pair of strings $f, g \in \{0, 1, \perp\}^{p(n)}$, supposed to be the encoding of adjacent satisfying assignments, and an auxiliary proof $\pi \in \{0, 1, \perp\}^{\ell(n)}$, supposed to assert that $f \circ g$ is as expected. Here, we say that a pair of assignments are *adjacent* if they differ in at most one variable. Informally, $\mathcal{V}_{\mathrm{HO}}$ performs the following three stages:

1. $\mathcal{V}_{\mathrm{HO}}$ makes sure that f or g is a codeword of Enc (by using its local tester). If this is not the case, then it immediately rejects.
2. $\mathcal{V}_{\mathrm{HO}}$ determines if f or g contains "many" $\perp$'s (by random sampling), and if so, then it immediately accepts. This stage is crucial for achieving the perfect completeness.
3. Otherwise (i.e., both f and g contain "few" $\perp$'s), $\mathcal{V}_{\mathrm{HO}}$ runs an assignment tester on $f \circ g \circ \pi$ (as if $f \circ g \circ \pi$ does not contain any $\perp$) to ensure that $f \circ g$ is indeed the encoding of adjacent satisfying assignments to φ, which is a crucial stage for the soundness.

One subtle issue is that implementing $\mathcal{V}_{\mathrm{HO}}$ causes minor changes in the assignment tester, which makes its analysis less intuitive.

Karthik C. S.–Manurangsi's System

Karthik C. S. and Manurangsi [29] first create fresh *four* copies of the input string, denoted by f_1, f_2, f_3, f_4, so that whenever three of them are the encoded adjacent satisfying assignments to φ, the remaining copy can be whatever. Create also a special variable v^* to denote which copy is currently "in transition." The verifier $\mathcal{V}_{\mathrm{KM}}$ of [29] is given oracle access to four strings $f_1, f_2, f_3, f_4 \in \{0, 1\}^{p(n)}$, three of which are supposed to be the encoding of adjacent satisfying assignments, four auxiliary proofs $\pi_1, \pi_2, \pi_3, \pi_4 \in \{0, 1\}^{\ell(n)}$, where each π_k should assert that $\{f_i\}_{i \neq k}$ are as expected, and the special variable $v^* \in [4]$. Roughly speaking, $\mathcal{V}_{\mathrm{KM}}$ works as follows: Suppose that $\mathcal{V}_{\mathrm{KM}}$ finds v^* to take a value $k \in [4]$; i.e., f_k is on the way of transition. Then, $\mathcal{V}_{\mathrm{KM}}$ calls an assignment tester on $\{f_i\}_{i \neq k} \circ \pi_k$ to confirm that $\{f_i\}_{i \neq k}$ are indeed the encoding of adjacent satisfying assignments to φ.

Unlike $\mathcal{V}_{\mathrm{HO}}$ of [22], implementing $\mathcal{V}_{\mathrm{KM}}$ does not require any modification in assignment testers. However, $\mathcal{V}_{\mathrm{KM}}$ entails four copies of the input string and the special variable v^* to take the majority decoding, which makes the (soundness) analysis slightly complicated. Another drawback common to [22, 29] is that both $\mathcal{V}_{\mathrm{HO}}$ and $\mathcal{V}_{\mathrm{KM}}$ apply error-correcting codes and assignment testers all at once, and thus, we are not able to analyze their effects *separately*.

1.3 Overview of Our Proof

We now outline our proof of the PCRP theorem. Our proof also relies on error-correcting codes and assignment testers as in [22, 29], but does not involve either the special symbol $\perp$ or the special variable v^*. The crucial difference from [22, 29] is that our PCRP system is obtained by combining simple **robustization** and **composition** steps in a modular fashion, which renders its analysis more intuitive. This can be thought of as an extension of **alphabet reduction** [11] for Maxmin 2-CSP Reconfiguration due to [35].

Starting Point (Section 3.1)

The starting point for which we build a PCRP system is a **PSPACE**-complete problem called Succinct Reach [14, 22, 40], defined as follows: Given a circuit $C\colon \{0,1\}^{2n} \to \{0,1\}$ that represents an exponentially large graph G over $\{0,1\}^n$ such that a pair of "vertices" $\alpha, \beta \in \{0,1\}^n$ are adjacent in G if and only if $C(\alpha \circ \beta) = 1$, we are required to decide if there exists an undirected path from 0^n to 1^n in G.[2] Succinct Reach can be thought of as a reconfiguration problem, which asks if there exists a *reconfiguration edge sequence* $(\alpha^{(1)} \circ \beta^{(1)}, \dots, \alpha^{(T)} \circ \beta^{(T)})$ from $0^n \circ 0^n$ to $1^n \circ 1^n$ such that $\alpha^{(t)} = \alpha^{(t+1)}$ or $\beta^{(t)} = \beta^{(t+1)}$ for every t, and every $(\alpha^{(t)}, \beta^{(t)})$ is an edge of G.[3] Such a sequence has the following interpretation [22] under the *token jumping model* [28]: Given a pair of "tokens" initially placed on self-loop $(0^n, 0^n)$ of G, we can transfer them to self-loop $(1^n, 1^n)$ of G by repeatedly moving a single token while preserving that the two tokens are always adjacent in G.

Robustization (Section 3.2)

The first step is **robustization** [4, 12], which reduces Succinct Reach to a "robust" version of Circuit SAT Reconfiguration. Given a circuit $\Phi\colon \{0,1\}^{2p(n)} \to \{0,1\}$ and a pair of its satisfying assignments $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}}$, Circuit SAT Reconfiguration asks to decide if there exists a reconfiguration sequence $\vec{\sigma}$ from σ^{ini} to σ^{end} consisting only of satisfying assignments to Φ. We shall achieve the following requirement in reducing an input C of Succinct Reach to an input $(\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}}; \Phi)$ of Circuit SAT Reconfiguration (Lemma 3.7):

- (*Perfect completeness*) If C is a YES instance, then $(\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}}; \Phi)$ is a YES instance.
- (*Robust soundness*) If C is a NO instance, then every possible reconfiguration sequence $\vec{\sigma}$ from σ^{ini} to σ^{end} contains some assignment that is $\Omega(1)$-far from every satisfying assignment to Φ.

In the same manner as [22, 29], we first encode each "vertex" represented by $C\colon \{0,1\}^{2n} \to \{0,1\}$ using an error-correcting code $\mathsf{Enc}\colon \{0,1\}^n \to \{0,1\}^{p(n)}$, whose relative distance will be denoted by ρ. Obviously, two (supposedly satisfying) assignments $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}} \in \{0,1\}^{2p(n)}$ should be defined as $\sigma^{\mathsf{ini}} := \mathsf{Enc}(0^n) \circ \mathsf{Enc}(0^n)$ and $\sigma^{\mathsf{end}} := \mathsf{Enc}(1^n) \circ \mathsf{Enc}(1^n)$. The central question is how to design a new circuit $\Phi\colon \{0,1\}^{2p(n)} \to \{0,1\}$, which takes a pair of strings $f, g \in \{0,1\}^{p(n)}$, so as to meet the above requirements.

The first naive attempt mimics the existing robustization, e.g., [11]: Our (seemingly promising) circuit accepts only $\mathsf{Enc}(\alpha) \circ \mathsf{Enc}(\beta)$ such that $C(\alpha \circ \beta) = 1$, which, however, fails to derive the perfect completeness. Intuitively, there is no consecutive path between distinct codewords of Enc, making the reconfiguration impossible.

[2] Without loss of generality, we can assume that G is *undirected* and has *self-loops* at every vertex.
[3] We remark that reconfiguration edge sequences are different from reconfiguration sequences, in which every adjacent pair of strings differ in at most *one* bit.

▶ **Example 1.3** (first failed attempt). Construct a circuit $\widetilde{\Phi}_1$ such that $\widetilde{\Phi}_1(f \circ g) = 1$ if and only if

1. both f and g are some codewords of Enc, and
2. if $f = \mathsf{Enc}(\alpha)$ and $g = \mathsf{Enc}(\beta)$, then $C(\alpha \circ \beta) = 1$.

Consider any reconfiguration sequence $\vec{\sigma}$ from σ^{ini} to σ^{end}. Since $\mathsf{Enc}(0^n)$ and $\mathsf{Enc}(1^n)$ are ρ-far from each other, $\vec{\sigma}$ must contain some assignment $\sigma^\circ = f^\circ \circ g^\circ$ such that f° is $\frac{\rho}{2}$-far from every codeword of Enc; i.e., σ° is $\frac{\rho}{4}$-far from every satisfying assignment to $\widetilde{\Phi}_1$, losing the perfect completeness. ⌟

Given the first attempt's failure, one may think of forcing the circuit to accept a pair of strings that are up to $\frac{\rho}{2}$-close to some codewords. Unfortunately, this modification now reduces the robust soundness to $o(1)$. The nature behind this failure is that moving one bit away from the radius of the ball results in a rejecting string, even though it is only one bit away from an accepting string.

▶ **Example 1.4** (second failed attempt). Construct a circuit $\widetilde{\Phi}_2$ such that $\widetilde{\Phi}_2(f \circ g) = 1$ if and only if

1. both f and g are $\frac{\rho}{2}$-close to some codewords of Enc, and
2. if f and g are $\frac{\rho}{2}$-close to $\mathsf{Enc}(\alpha)$ and $\mathsf{Enc}(\beta)$, respectively, then $C(\alpha \circ \beta) = 1$.

Then, the following issue arises: Even if an assignment $\sigma \in \{0,1\}^{2p(n)}$ is $\frac{\rho}{2}$-far from $\mathsf{Enc}(\alpha) \circ \mathsf{Enc}(\beta)$ for every strings $\alpha, \beta \in \{0,1\}^n$, we cannot exclude the possibility that σ is still $o(1)$-close to some satisfying assignment to $\widetilde{\Phi}_2$. Let $f \in \{0,1\}^{p(n)}$ be a string that is $\frac{\rho}{2}$-close to both $\mathsf{Enc}(0^n)$ and $\mathsf{Enc}(1^n)$. Consider the following reconfiguration sequence $\vec{\sigma}$ from σ^{ini} to σ^{end} via $f \circ f$:

$$\big(\mathsf{Enc}(0^n) \circ \mathsf{Enc}(0^n), \cdots, f \circ \mathsf{Enc}(0^n), \cdots, f \circ f, \cdots, f \circ \mathsf{Enc}(1^n), \cdots, \mathsf{Enc}(1^n) \circ \mathsf{Enc}(1^n)\big).$$

Changing a few bits of f, we can reach a new string $f^\star$ that is $\frac{\rho}{2}$-close to $\mathsf{Enc}(0^n)$ but $\frac{\rho}{2}$-far from any other codeword of Enc. Since $\widetilde{\Phi}_2(f^\star \circ f^\star) = 1$ by definition, $f \circ f$ is $o(1)$-close to a satisfying assignment (i.e., $f^\star \circ f^\star$) to $\widetilde{\Phi}_2$. Similarly, every assignment in $\vec{\sigma}$ can be shown $o(1)$-close to some satisfying assignment to $\widetilde{\Phi}_2$. ⌟

The crux of realizing the perfect completeness and robust soundness simultaneously is the following error-correcting code, which enjoys both *list decodability* and *reconfigurability*.

▶ **Theorem 1.5** (informal; see Theorem 3.4 [16] and Lemma 3.5). *There exists an error-correcting code* $\mathsf{Enc}\colon \{0,1\}^n \to \{0,1\}^{p(n)}$ *for some polynomial* p *such that the following hold*:

- (*List decodability*) *There exists a polynomial-time algorithm that list-decodes* Enc *up to relative radius* $\frac{1}{3}$.
- (*Reconfigurability*) *For every distinct strings* $\alpha \neq \beta \in \{0,1\}^n$, *there exists a reconfiguration sequence from* $\mathsf{Enc}(\alpha)$ *to* $\mathsf{Enc}(\beta)$ *in which every string is*
  - $\frac{1}{4}$*-close to at least either* $\mathsf{Enc}(\alpha)$ *or* $\mathsf{Enc}(\beta)$, *and*
  - $\frac{1}{3}$*-far from* $\mathsf{Enc}(\gamma)$ *for every string* $\gamma \neq \alpha, \beta$.

The latter property enables to reconfigure between a distinct pair of codewords, while avoiding getting too (say, $\frac{1}{3}$) close to any other codewords. Specifically, a standard concatenation of Reed–Solomon codes and Hadamard codes [1, 13, 16] suffices to meet the list decodability [16] and the reconfigurability, which only involves an elementary proof. Note that the reconfigurability of Hadamard codes has been investigated by Ohsaka [35].

Owing to Theorem 1.5, we eventually implement the actual circuit $\Phi\colon \{0,1\}^{2p(n)} \to \{0,1\}$ such that $\Phi(f \circ g) = 1$ if and only if

1. both f and g are $\frac{1}{4}$-close to some codewords of Enc, and
2. if f and g are $\frac{1}{3}$-close to $\mathsf{Enc}(\alpha)$ and $\mathsf{Enc}(\beta)$, respectively, then $C(\alpha \circ \beta) = 1$.

(The difference from $\widetilde{\Phi}_1$ and $\widetilde{\Phi}_2$ of Examples 1.3 and 1.4 is highlighted.) Intuitively speaking, the *threshold gap* between the first and second conditions enables us to bypass the issues of the two naive failed attempts at the same time.

Composition (Section 3.3)

The second step is **composition** [2, 3], which builds a PCRP system for a "robust" version of Circuit SAT Reconfiguration. Given a circuit $\Phi\colon \{0,1\}^{2p(n)} \to \{0,1\}$ and a pair of its satisfying assignments $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}}$ produced by the **robustization** step, we shall construct a verifier $\mathcal{V}$ and a pair of proofs $\Pi^{\mathsf{ini}}, \Pi^{\mathsf{end}} \in \{0,1\}^{\mathrm{poly}(n)}$ such that the following hold (Lemma 3.8):

- (*Perfect completeness*) If C is a YES instance, then there exists a reconfiguration sequence $\vec{\Pi}$ from Π^{ini} to Π^{end} such that $\mathcal{V}$ accepts every proof in $\vec{\Pi}$ with probability 1.
- (*Soundness*) If C is a NO instance, then every reconfiguration sequence $\vec{\Pi}$ from Π^{ini} to Π^{end} contains some proof that is rejected by $\mathcal{V}$ with probability more than $\frac{1}{2}$.

A naive failed attempt is to just feed Φ to an assignment tester $\mathcal{A}$ as in the **NP** regime [11]. Given oracle access to an assignment $\sigma \in \{0,1\}^{2p(n)}$ and an auxiliary proof $\pi \in \{0,1\}^{\ell(n)}$, $\mathcal{A}$ meets the following conditions:

- If $\Phi(\sigma) = 1$, then there exists a proof π such that $\mathcal{A}^{\sigma\circ\pi}(\Phi)$ accepts with probability 1.
- If σ is ε-far from every satisfying assignment to Φ, then $\mathcal{A}^{\sigma\circ\pi}(\Phi)$ rejects with probability $\Omega(\varepsilon)$ for every proof π.

Define $\Pi^{\mathsf{ini}} \coloneqq \sigma^{\mathsf{ini}} \circ \pi^{\mathsf{ini}}$ and $\Pi^{\mathsf{end}} \coloneqq \sigma^{\mathsf{end}} \circ \pi^{\mathsf{end}}$ for some proofs $\pi^{\mathsf{ini}}, \pi^{\mathsf{end}} \in \{0,1\}^{\ell(n)}$ such that both $\mathcal{A}^{\Pi^{\mathsf{ini}}}(\Phi)$ and $\mathcal{A}^{\Pi^{\mathsf{end}}}(\Phi)$ accept with probability 1. The robust soundness of Circuit SAT Reconfiguration implies the aforementioned soundness. On the other hand, the perfect completeness would be broken because we need to reconfigure between Π^{ini} and Π^{end}, which may be significantly different.

We resolve this issue by creating *twins* of proofs π_1, π_2 so that a kind of "redundancy" is ensured. Our actual PCRP verifier $\mathcal{V}$ runs $\mathcal{A}^{\sigma\circ\pi_1}(\Phi)$ and $\mathcal{A}^{\sigma\circ\pi_2}(\Phi)$ independently, and accepts if (at least) either of the runs accepted. The perfect completeness is almost immediate by definition of $\mathcal{V}$. If σ is ε-far from every satisfying assignment to Φ, then $\mathcal{V}$ rejects with probability $\Omega(\varepsilon^2)$ for every alleged proofs π_1, π_2, implying the desired soundness.

2 Preliminaries

Notations and Definitions

For a nonnegative integer $n \in \mathbb{N}$, let $[n] \coloneqq \{1, 2, \ldots, n\}$. The base of logarithms is 2. A *sequence* of a finite number of elements $a^{(1)}, \ldots, a^{(T)}$ is denoted by $\vec{a} = (a^{(1)}, \ldots, a^{(T)})$, and we write $a \in \vec{a}$ to indicate that a appears in $\vec{a}$ (at least once). The symbol $\circ$ stands for a concatenation of two strings, $\langle \cdot, \cdot \rangle$ for the inner product, $\mathbb{F}_q$ with a prime power q for the finite field with q elements, 0^n for $\underbrace{0 \cdots 0}_{n \text{ times}}$, and 1^n for $\underbrace{1 \cdots 1}_{n \text{ times}}$. Let Σ denote a finite set called *alphabet*. For a string $f \in \Sigma^n$ and an index set $I \subseteq [n]$, we use $f|_I \in \Sigma^I$ to denote the

restriction of f to I. The *relative Hamming distance* between two strings $f, g \in \Sigma^n$, denoted by $\delta(f,g)$, is defined as the fraction of positions at which f and g differ; namely,

$$\delta(f,g) \coloneqq \mathop{\mathbb{P}}_{i\sim[n]}\Big[f(i) \neq g(i)\Big]. \tag{2.1}$$

We say that f is *ε-close to g* if $\delta(f,g) \leqslant \varepsilon$ and *ε-far from g* if $\delta(f,g) > \varepsilon$. Analogous notations are used for a set of strings $S \subseteq \Sigma^n$; e.g., $\delta(f,S) \coloneqq \min_{g\in S} \delta(f,g)$ and f is ε-close to S if $\delta(f,S) \leqslant \varepsilon$.

2.1 Error-Correcting Codes

Here, we introduce error-correcting codes, followed by two examples.

▶ **Definition 2.1** (error-correcting code)**.** For a real $\rho \in [0,1]$, a function $\mathsf{Enc}\colon \Sigma^k \to \Sigma^n$ is an *error-correcting code* with *relative distance* ρ if $\delta(\mathsf{Enc}(\alpha), \mathsf{Enc}(\beta)) \geqslant \rho$ for every distinct strings $\alpha \neq \beta \in \Sigma^k$. Each $\mathsf{Enc}(\alpha)$ for $\alpha \in \Sigma^k$ is called a *codeword* of Enc. Denote by $\mathsf{Enc}(\cdot)$ the set of all codewords of Enc.

▶ **Definition 2.2** (Reed–Solomon code [41])**.** For a finite field $\mathbb{F}$ and two positive integers k, n such that $k \leqslant n \leqslant |\mathbb{F}|$, the *Reed–Solomon code* is defined as a function $\mathsf{RS}_{\mathbb{F}}\colon \mathbb{F}^k \to \mathbb{F}^n$ such that $\mathsf{RS}_{\mathbb{F}}(\alpha) \coloneqq (f_\alpha(x_1), \ldots, f_\alpha(x_n))$ for each string $\alpha \in \mathbb{F}^k$, where $f_\alpha(x) \coloneqq \sum_{i\in[k]} \alpha(i) \cdot x^{i-1}$ and $x_1, \ldots, x_n$ are n distinct elements of $\mathbb{F}$.

▶ **Definition 2.3** (Hadamard code)**.** For a positive integer n, the *Hadamard code* is defined as a function $\mathsf{Had}\colon \{0,1\}^n \to \{0,1\}^{2^n}$ such that $\mathsf{Had}(\alpha) \coloneqq (\langle \alpha, x_1\rangle, \ldots, \langle \alpha, x_{2^n}\rangle)$ for each string $\alpha \in \{0,1\}^n$, where x_i is the i^{th} string of $\{0,1\}^n$.[4]

The relative distance of Reed–Solomon codes and Hadamard codes is $\frac{n-k+1}{n}$ and $\frac{1}{2}$, respectively, see, e.g., [17]. Moreover, $\mathsf{Had}(\alpha)$ and $\mathsf{Had}(\beta)$ for every distinct strings $\alpha \neq \beta \in \{0,1\}^n$ differ in *exactly half* the bits.

2.2 Verifiers and Assignment Testers

Subsequently, we introduce the notion of *verifier*, followed by *assignment tester* [12] (a.k.a. *PCP of proximity* [4]).

▶ **Definition 2.4** (verifier)**.** A *verifier* with *randomness complexity* $r\colon \mathbb{N} \to \mathbb{N}$ and *query complexity* $q\colon \mathbb{N} \to \mathbb{N}$ is a probabilistic polynomial-time algorithm $\mathcal{V}$ that given an input $x \in \{0,1\}^*$, tosses $r = r(|x|)$ random bits $R \in \{0,1\}^r$ and uses R to generate a sequence of $q = q(|x|)$ queries $I = (i_1, \ldots, i_q)$ and a circuit $D\colon \{0,1\}^q \to \{0,1\}$. Given an input $x \in \{0,1\}^*$ and oracle access to a *proof* $\pi \in \{0,1\}^*$, denote $\mathcal{V}$'s (randomized) output by $\mathcal{V}^\pi(x) \coloneqq D(\pi|_I)$ over the randomness of R. We say that $\mathcal{V}(x)$ *accepts* π or simply $\mathcal{V}^\pi(x)$ *accepts* if $\mathcal{V}^\pi(x) = 1$, and that $\mathcal{V}^\pi(x)$ *rejects* if $\mathcal{V}^\pi(x) = 0$.

▶ **Definition 2.5** (assignment tester [4, 12])**.** An *assignment tester* with *rejection rate* $\kappa > 0$ is a verifier $\mathcal{A}$ such that for a polynomial-size circuit $\Phi\colon \{0,1\}^n \to \{0,1\}$ (as *explicit* input) and oracle access to its assignment $y \in \{0,1\}^n$ (as *implicit* input) and a proof $\pi \in \{0,1\}^*$, the following hold:

[4] The inner product is taken modulo 2.

- (Perfect completeness) If $\Phi(y) = 1$, then there exists a proof $\pi \in \{0,1\}^*$ such that $\mathcal{A}(x)$ accepts $y \circ \pi$ with probability 1; namely,

$$\exists \pi \in \{0,1\}^*, \quad \mathbb{P}\Big[\mathcal{A}^{y\circ\pi}(x) = 1\Big] = 1. \tag{2.2}$$

- (Soundness) If y is ε-far from every satisfying assignment to Φ, then for every proof $\pi \in \{0,1\}^*$, $\mathcal{A}(x)$ rejects $y \circ \pi$ with probability more than $\kappa \cdot \varepsilon$; namely,

$$\forall \pi \in \{0,1\}^*, \quad \mathbb{P}\Big[\mathcal{A}^{y\circ\pi}(x) = 0\Big] > \kappa \cdot \varepsilon. \tag{2.3}$$

There exists an assignment tester with randomness complexity $\mathcal{O}(\log n)$, query complexity $\mathcal{O}(1)$, and rejection rate $\Omega(1)$, described as follows.

▶ **Theorem 2.6** ([4, 12]). *There exists an assignment tester $\mathcal{A}$ with randomness complexity $r(n) = \mathcal{O}(\log n)$, query complexity $q(n) = \mathcal{O}(1)$, and rejection rate $\kappa > 0$. Moreover, for a polynomial-size circuit $\Phi\colon \{0,1\}^n \to \{0,1\}$ and its satisfying assignment $y \in \{0,1\}^n$, a proof $\pi \in \{0,1\}^*$ such that $\mathcal{A}^{y\circ\pi}(x)$ accepts with probability 1 can be computed in polynomial time.*

3 A Simple Proof of the PCRP Theorem

In this section, we present a simple proof of the *Probabilistically Checkable Reconfiguration Proof* (PCRP) theorem. For a pair of proofs $\pi^{\mathsf{ini}}, \pi^{\mathsf{end}} \in \{0,1\}^n$, a *reconfiguration sequence* from π^{ini} to π^{end} is defined as a sequence $\vec{\pi} = (\pi^{(1)}, \dots, \pi^{(T)})$ over $\{0,1\}^n$ such that $\pi^{(1)} = \pi^{\mathsf{ini}}$, $\pi^{(T)} = \pi^{\mathsf{end}}$, and $\pi^{(t)}$ and $\pi^{(t+1)}$ differ in at most one bit for every t. The PCRP theorem is formally stated as follows.

▶ **Theorem 3.1** (Probabilistically Checkable Reconfiguration Proof theorem [22, 29]). *A language $L \subseteq \{0,1\}^*$ is in* **PSPACE** *if and only if there exists a verifier $\mathcal{V}$ with randomness complexity $r(n) = \mathcal{O}(\log n)$ and query complexity $q(n) = \mathcal{O}(1)$ and a pair of polynomial-time computable proofs $\pi^{\mathsf{ini}}, \pi^{\mathsf{end}}\colon \{0,1\}^* \to \{0,1\}^*$ such that the following hold for every input $x \in \{0,1\}^*$:*

- *(Completeness) If $x \in L$, then there exists a reconfiguration sequence $\vec{\pi} = (\pi^{(1)}, \dots, \pi^{(T)})$ from $\pi^{\mathsf{ini}}(x)$ to $\pi^{\mathsf{end}}(x)$ such that $\mathcal{V}(x)$ accepts every proof $\pi^{(t)}$ in $\vec{\pi}$ with probability 1; namely,*

$$\forall t \in [T], \quad \mathbb{P}\Big[\mathcal{V}^{\pi^{(t)}}(x) = 1\Big] = 1. \tag{3.1}$$

- *(Soundness) If $x \notin L$, then every reconfiguration sequence $\vec{\pi} = (\pi^{(1)}, \dots, \pi^{(T)})$ from $\pi^{\mathsf{ini}}(x)$ to $\pi^{\mathsf{end}}(x)$ contains some proof $\pi^{(t)}$ that is rejected by $\mathcal{V}(x)$ with probability more than $\frac{1}{2}$; namely,*

$$\exists t \in [T], \quad \mathbb{P}\Big[\mathcal{V}^{\pi^{(t)}}(x) = 0\Big] > \frac{1}{2}. \tag{3.2}$$

Since the "if" direction of Theorem 3.1 is obvious (see, e.g., [22]), we prove the "only-if" direction; i.e., we will construct a PCRP system for a **PSPACE**-complete language in the remainder of this section.

3.1 Succinct Reach and PSPACE-completeness

We first introduce a **PSPACE**-complete problem called Succinct Reach, e.g., [14, 40], for which we design a PCRP system. The input of Succinct Reach is specified by a polynomial-size circuit $C\colon \{0,1\}^{2n} \to \{0,1\}$. Informally, C "succinctly" represents an exponentially large graph G over $\{0,1\}^n$ such that a pair of "vertices" $\alpha, \beta \in \{0,1\}^n$ are adjacent in G if and only if $C(\alpha \circ \beta) = 1$. Hereafter, we restrict C to satisfy the following conditions:

(C1) $C(\alpha \circ \beta) = C(\beta \circ \alpha)$ for every strings $\alpha, \beta \in \{0,1\}^n$ (i.e., G is *undirected*), and
(C2) $C(\alpha \circ \alpha) = 1$ for every string $\alpha \in \{0,1\}^n$ (i.e., G has *self-loops* at every vertex).
For four strings $\alpha^{\mathsf{ini}}, \beta^{\mathsf{ini}}, \alpha^{\mathsf{end}}, \beta^{\mathsf{end}} \in \{0,1\}^n$, a *reconfiguration edge sequence* from $\alpha^{\mathsf{ini}} \circ \beta^{\mathsf{ini}}$ to $\alpha^{\mathsf{end}} \circ \beta^{\mathsf{end}}$ is a sequence $(\alpha^{(1)} \circ \beta^{(1)}, \dots, \alpha^{(T)} \circ \beta^{(T)})$ over $\{0,1\}^{2n}$ such that $\alpha^{(1)} \circ \beta^{(1)} = \alpha^{\mathsf{ini}} \circ \beta^{\mathsf{ini}}$, $\alpha^{(T)} \circ \beta^{(T)} = \alpha^{\mathsf{end}} \circ \beta^{\mathsf{end}}$, and $(\alpha^{(t)} = \alpha^{(t+1)}$ or $\beta^{(t)} = \beta^{(t+1)})$ for every t. We are now ready to define the Succinct Reach problem.

▶ **Problem 3.2.** Given a polynomial-size circuit $C\colon \{0,1\}^{2n} \to \{0,1\}$ that satisfies (C1) and (C2), Succinct Reach asks to decide if there exists a reconfiguration edge sequence $(\alpha^{(1)} \circ \beta^{(1)}, \dots, \alpha^{(T)} \circ \beta^{(T)})$ from $0^n \circ 0^n$ to $1^n \circ 1^n$ such that $C(\alpha^{(t)} \circ \beta^{(t)}) = 1$ for every $t \in [T]$.

As an immediate corollary of Hirahara and Ohsaka [22, Proposition 5.3], we have the **PSPACE**-completeness of Succinct Reach.

▶ **Corollary 3.3** (from [22, Proposition 5.3])**.** *Succinct Reach is* **PSPACE***-complete.*

3.2 Reducing Succinct Reach to "Robust" Version of Circuit SAT Reconfiguration

We construct a polynomial-time reduction from Succinct Reach to a "robust" version of Circuit SAT Reconfiguration. Given a polynomial-size circuit $\Phi\colon \{0,1\}^n \to \{0,1\}$ and a pair of its satisfying assignments $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}} \in \{0,1\}^n$, Circuit SAT Reconfiguration asks to decide if there exists a reconfiguration sequence $\vec{\sigma} = (\sigma^{(1)}, \dots, \sigma^{(T)})$ from σ^{ini} to σ^{end} such that $\Phi(\sigma^{(t)}) = 1$ for every t. The *robust soundness* requests that every reconfiguration sequence $\vec{\sigma}$ from σ^{ini} to σ^{end} contains some assignment that is $\Omega(1)$-far from every satisfying assignment to Φ. To achieve this requirement, we encode each "vertex" represented by a polynomial-size circuit $C\colon \{0,1\}^{2n} \to \{0,1\}$ of Succinct Reach using a special error-correcting code that enjoys *list decodability* and *reconfigurability*.

Concatenated Codes, List Decodability, and Reconfigurability

Our error-correcting code Enc is obtained as a standard concatenation of Reed–Solomon codes and Hadamard codes [13], see also [1, Chapter 19] and [16, Chapter 8]. Let $n \in \mathbb{N}$ be a positive integer that is a power of 2 for simplicity of notation. Define $B \coloneqq 2^8 = 256$, $\varepsilon_0 \coloneqq \frac{1}{B}$, and $\mathbb{F} \coloneqq \mathbb{F}_{2^{\log(Bn)}}$. Note that $|\mathbb{F}| = Bn$. The *concatenation* of a Reed–Solomon code $\mathsf{RS}_{\mathbb{F}}\colon \mathbb{F}^n \to \mathbb{F}^{Bn}$ with a Hadamard code $\mathsf{Had}\colon \{0,1\}^{\log(Bn)} \to \{0,1\}^{Bn}$, denoted by $\mathsf{Had} \circ \mathsf{RS}_{\mathbb{F}}\colon \{0,1\}^{n\log(Bn)} \to \{0,1\}^{(Bn)^2}$, is defined as follows. By a canonical bijection between elements in $\mathbb{F}$ and strings in $\{0,1\}^{\log|\mathbb{F}|}$, we consider $\mathsf{RS}_{\mathbb{F}}$ as an *outer code* from $\{0,1\}^{n\log|\mathbb{F}|}$ to $\mathbb{F}^{Bn}$ whereas Had as an *inner code* from $\mathbb{F}$ to $\{0,1\}^{|\mathbb{F}|}$. For each string $\alpha \in \{0,1\}^{n\log(Bn)}$, we define $(\mathsf{Had} \circ \mathsf{RS}_{\mathbb{F}})(\alpha)$ as

$$(\mathsf{Had} \circ \mathsf{RS}_{\mathbb{F}})(\alpha) \coloneqq \Big(\mathsf{Had}\big(\mathsf{RS}_{\mathbb{F}}(\alpha)_1\big), \dots, \mathsf{Had}\big(\mathsf{RS}_{\mathbb{F}}(\alpha)_{Bn}\big)\Big), \tag{3.3}$$

where $\mathsf{RS}_{\mathbb{F}}(\alpha)_k \in \mathbb{F}$ is the k^{th} symbol of $\mathsf{RS}_{\mathbb{F}}(\alpha) \in \mathbb{F}^{Bn}$. Define finally $\mathsf{Enc}\colon \{0,1\}^n \to \{0,1\}^{p(n)}$ as $\mathsf{Enc}(\alpha) \coloneqq (\mathsf{Had} \circ \mathsf{RS}_{\mathbb{F}})(\alpha \circ 0^{n\log(Bn)-n})$ for each string $\alpha \in \{0,1\}^n$, where $p(n) \coloneqq (Bn)^2$. Note that the relative distance of $\mathsf{Had} \circ \mathsf{RS}_{\mathbb{F}}$ (and thus Enc) is at least the product of the relative distances of the outer and inner codes, i.e., $\frac{1}{2} \cdot \frac{Bn-n+1}{Bn} > \frac{1-\varepsilon_0}{2}$.

Guruswami [16] established the *list decodability* of $\mathsf{Had} \circ \mathsf{RS}_{\mathbb{F}}$ for general finite field $\mathbb{F}$, which implies the following.

▶ **Theorem 3.4** (list decodability of Enc [16, Theorem 8.2]). *There exists a polynomial-time list-decoding algorithm $\mathcal{D}_{\mathsf{Enc}}$ that takes a string $f \in \{0,1\}^{p(n)}$ and produces a list of all codewords in $\mathsf{Enc}(\cdot)$ that are ρ-close to f, where*

$$\rho \coloneqq \frac{1-\sqrt{\varepsilon_0}}{2} - \mathcal{O}\big(\tfrac{1}{n^2}\big). \tag{3.4}$$

Moreover, Enc enjoys the following *reconfigurability* property.

▶ **Lemma 3.5** (reconfigurability of Enc). *For every distinct strings $\alpha \neq \beta \in \{0,1\}^n$, there exists a reconfiguration sequence $\vec{f} = (f^{(1)}, \dots, f^{(T)})$ from $\mathsf{Enc}(\alpha)$ to $\mathsf{Enc}(\beta)$ over $\{0,1\}^{p(n)}$ such that for every string $\gamma \in \{0,1\}^n \setminus \{\alpha,\beta\}$ and every $t \in [T]$,*

$$\min\Big\{\delta\big(f^{(t)}, \mathsf{Enc}(\alpha)\big), \delta\big(f^{(t)}, \mathsf{Enc}(\beta)\big)\Big\} \leqslant \frac{1}{4}, \tag{3.5}$$

$$\delta\big(f^{(t)}, \mathsf{Enc}(\gamma)\big) > \frac{1}{2} - \varepsilon_0 - \mathcal{O}\big(\tfrac{1}{n}\big). \tag{3.6}$$

Proof. For any distinct strings $\alpha \neq \beta \in \{0,1\}^n$, let $\overline{\alpha} \coloneqq \alpha \circ 0^{n\log(Bn)-n}$ and $\overline{\beta} \coloneqq \beta \circ 0^{n\log(Bn)-n}$. Note that $\mathsf{Enc}(\alpha) = (\mathsf{Had} \circ \mathsf{RS}_{\mathbb{F}})(\overline{\alpha})$ and $\mathsf{Enc}(\beta) = (\mathsf{Had} \circ \mathsf{RS}_{\mathbb{F}})(\overline{\beta})$. Think of a string in $\{0,1\}^{(Bn)^2}$ as consisting of Bn *blocks* in $\{0,1\}^{Bn}$. Recall that if $\mathsf{RS}_{\mathbb{F}}(\overline{\alpha})_k \neq \mathsf{RS}_{\mathbb{F}}(\overline{\beta})_k$ for the k^{th} block, then $\mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\alpha})_k)$ and $\mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\beta})_k)$ differ in *exactly* $\frac{Bn}{2}$ bits owing to the relative distance of Had. Consider a reconfiguration sequence $\vec{f}$ from $\mathsf{Enc}(\alpha)$ to $\mathsf{Enc}(\beta)$ obtained by the following procedure.

Reconfiguration sequence $\vec{f}$ from $\mathsf{Enc}(\alpha)$ to $\mathsf{Enc}(\beta)$

1: **for each** $k \in [Bn]$ such that $\mathsf{RS}_{\mathbb{F}}(\overline{\alpha})_k \neq \mathsf{RS}_{\mathbb{F}}(\overline{\beta})_k$ **do**
2: change the k^{th} block of the current string from $\mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\alpha})_k)$ to $\mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\beta})_k)$ by flipping exactly $\frac{Bn}{2}$ bits on which they are different.

Let $f^\circ = (f^\circ_1, \dots, f^\circ_{Bn}) \in (\{0,1\}^{Bn})^{Bn}$ be any intermediate string of $\vec{f}$. We first prove Eq. (3.5) for f°. By construction, Bn blocks of f° can be partitioned into $X \cup Y \cup Z = [Bn]$ such that $f^\circ_k = \mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\alpha})_k)$ for every $k \in X$, $f^\circ_k = \mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\beta})_k)$ for every $k \in Y$, and f°_k is neither $\mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\alpha})_k)$ nor $\mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\beta})_k)$ for every $k \in Z$ (i.e., it is on the way of transition). Since $|Z| \leqslant 1$ and every block f°_k is $\frac{1}{2}$-close to both $\mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\alpha})_k)$ and $\mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\beta})_k)$, we derive

$$\begin{aligned}
\delta\big(f^\circ, \mathsf{Enc}(\alpha)\big) &\leqslant \frac{1}{Bn} \cdot \Big(0 \cdot |X| + \tfrac{1}{2} \cdot |Y| + \tfrac{1}{2} \cdot |Z|\Big) \leqslant \frac{|Y|+|Z|}{2Bn}, \\
\delta\big(f^\circ, \mathsf{Enc}(\beta)\big) &\leqslant \frac{1}{Bn} \cdot \Big(\tfrac{1}{2} \cdot |X| + 0 \cdot |Y| + \tfrac{1}{2} \cdot |Z|\Big) \leqslant \frac{|X|+|Z|}{2Bn},
\end{aligned} \tag{3.7}$$

$$\implies \min\Big\{\delta\big(f^\circ, \mathsf{Enc}(\alpha)\big), \delta\big(f^\circ, \mathsf{Enc}(\beta)\big)\Big\} \leqslant \frac{\min\big\{|X|+|Z|, |Y|+|Z|\big\}}{2Bn} \leqslant \frac{\frac{Bn}{2}}{2Bn} = \frac{1}{4}, \tag{3.8}$$

where we used the fact that $\min\{|X|+|Z|, |Y|+|Z|\} \leqslant \frac{Bn}{2}$.

We then prove Eq. (3.6) for f°. Let $\gamma \in \{0,1\}^n \setminus \{\alpha,\beta\}$ and $\overline{\gamma} \coloneqq \gamma \circ 0^{n\log(Bn)-n}$. Since the relative distance of $\mathsf{RS}_{\mathbb{F}}$ is $1-\varepsilon_0$, we have $\mathsf{RS}_{\mathbb{F}}(\overline{\gamma})_k = \mathsf{RS}_{\mathbb{F}}(\overline{\alpha})_k$ or $\mathsf{RS}_{\mathbb{F}}(\overline{\gamma})_k = \mathsf{RS}_{\mathbb{F}}(\overline{\beta})_k$ for at most $2\varepsilon_0 \cdot Bn$ number of k's in $X \cup Y$. Therefore, $\delta(f^\circ_k, \mathsf{Had}(\mathsf{RS}_{\mathbb{F}}(\overline{\gamma})_k)) = \frac{1}{2}$ for at least $(|X \cup Y| - 2\varepsilon_0 \cdot Bn)$ number of k's in $X \cup Y$. Consequently, we get

$$\delta\big(f^\circ, \mathsf{Enc}(\gamma)\big) \geqslant \frac{1}{Bn} \cdot \frac{1}{2}\Big(|X| + |Y| - 2\varepsilon_0 \cdot Bn\Big) \geqslant \frac{1}{2} - \varepsilon_0 - \mathcal{O}\big(\tfrac{1}{n}\big), \tag{3.9}$$

where we used the fact that $|X| + |Y| \geqslant Bn - 1$, as desired. ◀

▶ Remark 3.6. In the proof of Lemma 3.5, we essentially used the following simple properties of Hadamard and Reed–Solomon codes:

- Since $\delta(\mathsf{Enc}(\alpha), \mathsf{Enc}(\beta)) \geqslant \frac{1}{2}$ for every distinct strings $\alpha \neq \beta$ by the relative distance of Hadamard codes, we can transform $\mathsf{Enc}(\alpha)$ into $\mathsf{Enc}(\beta)$ without getting $\frac{1}{4}$-far from both $\mathsf{Enc}(\alpha)$ and $\mathsf{Enc}(\beta)$.
- Since $\delta(\mathsf{RS}_{\mathbb{F}}(\alpha), \mathsf{RS}_{\mathbb{F}}(\beta)) \geqslant 1 - \varepsilon_0$ for every distinct strings $\alpha \neq \beta$ by the relative distance of Reed–Solomon codes, we can transform $\mathsf{RS}_{\mathbb{F}}(\alpha)$ into $\mathsf{RS}_{\mathbb{F}}(\beta)$ without getting close to any other codeword $\mathsf{RS}_{\mathbb{F}}(\gamma)$.

Reduction

Our reduction from Succinct Reach to (a "robust" version of) Circuit SAT Reconfiguration is described as follows. Given a polynomial-size circuit $C\colon \{0,1\}^{2n} \to \{0,1\}$ as an input of Succinct Reach, let $\mathsf{Enc}\colon \{0,1\}^n \to \{0,1\}^{p(n)}$ be the proposed error-correcting code. Without loss of generality, we can assume that n is a power of 2 and sufficiently large so that Theorem 3.4 holds even if Eq. (3.4) is replaced by "$\rho := \frac{1}{3}$" and Lemma 3.5 holds even if Eq. (3.6) is replaced by "$\delta(f^{(t)}, \mathsf{Enc}(\gamma)) \geqslant \frac{2}{5} > \frac{1}{3}$." Create a new polynomial-size circuit $\Phi\colon \{0,1\}^{2p(n)} \to \{0,1\}$ such that $\Phi(f \circ g) = 1$ for two strings $f, g \in \{0,1\}^{p(n)}$ if and only if the following hold:

$$\max\Big\{\delta\big(f, \mathsf{Enc}(\cdot)\big), \delta\big(g, \mathsf{Enc}(\cdot)\big)\Big\} \leqslant \frac{1}{4}, \tag{3.10}$$

$$\forall \alpha, \beta \in \{0,1\}^n, \quad \left(\max\Big\{\delta\big(f, \mathsf{Enc}(\alpha)\big), \delta\big(g, \mathsf{Enc}(\beta)\big)\Big\} \leqslant \frac{1}{3} \implies C(\alpha, \beta) = 1\right). \tag{3.11}$$

Specifically, Φ can be implemented as follows.

Implementation of Φ

Input: two strings $f, g \in \{0,1\}^{p(n)}$.
1: run $\mathcal{D}_{\mathsf{Enc}}$ on f to generate a string list $\mathscr{L}_f := \big\{\alpha \in \{0,1\}^n \bigm| \delta(f, \mathsf{Enc}(\alpha)) \leqslant \frac{1}{3}\big\}$.
2: run $\mathcal{D}_{\mathsf{Enc}}$ on g to generate a string list $\mathscr{L}_g := \big\{\beta \in \{0,1\}^n \bigm| \delta(g, \mathsf{Enc}(\beta)) \leqslant \frac{1}{3}\big\}$.
3: **if** $\delta(f, \mathsf{Enc}(\alpha)) > \frac{1}{4}$ for every string $\alpha \in \mathscr{L}_f$ **then**
4: ⌊ **return** 0.
5: **if** $\delta(g, \mathsf{Enc}(\beta)) > \frac{1}{4}$ for every string $\beta \in \mathscr{L}_g$ **then**
6: ⌊ **return** 0.
7: **for each** $\alpha \in \mathscr{L}_f$ and $\beta \in \mathscr{L}_g$ **do**
8: ⌊ **if** $C(\alpha \circ \beta) = 0$ **then**
9: ⌊ ⌊ **return** 0.
10: **return** 1.

Define $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}} \in \{0,1\}^{2p(n)}$ as $\sigma^{\mathsf{ini}} := \mathsf{Enc}(0^n) \circ \mathsf{Enc}(0^n)$ and $\sigma^{\mathsf{end}} := \mathsf{Enc}(1^n) \circ \mathsf{Enc}(1^n)$. Observe easily that $\Phi(\sigma^{\mathsf{ini}}) = 1$ and $\Phi(\sigma^{\mathsf{end}}) = 1$, which completes the description of the reduction.

Correctness

We obtain the following "perfect" completeness and "robust" soundness.

▶ **Lemma 3.7.** *The following hold*:

- (*Perfect completeness*) *If C is a* YES *instance, then there exists a reconfiguration sequence $\vec{\sigma} = (\sigma^{(1)}, \dots, \sigma^{(T)})$ from σ^{ini} to σ^{end} such that $\Phi(\sigma^{(t)}) = 1$ for every $t \in [T]$.*

- (*Robust soundness*) *If C is a* No *instance, then every reconfiguration sequence $\vec{\sigma} = (\sigma^{(1)}, \dots, \sigma^{(T)})$ from σ^{ini} to σ^{end} contains some assignment $\sigma^{(t)}$ that is $\frac{1}{48}$-far from every satisfying assignment to Φ.*

Proof. We first prove the perfect completeness. Suppose that C is a Yes instance; i.e., there exists a reconfiguration edge sequence $(\alpha^{(1)} \circ \beta^{(1)}, \dots, \alpha^{(T)} \circ \beta^{(T)})$ from $0^n \circ 0^n$ to $1^n \circ 1^n$ such that $C(\alpha^{(t)} \circ \beta^{(t)}) = 1$ for every $t \in [T]$. Consider a reconfiguration sequence $\vec{\sigma_t}$ from $\mathsf{Enc}(\alpha^{(t)}) \circ \mathsf{Enc}(\beta^{(t)})$ to $\mathsf{Enc}(\alpha^{(t+1)}) \circ \mathsf{Enc}(\beta^{(t+1)})$ for each $t \in [T-1]$ obtained by the following procedure.

Reconfiguration sequence $\vec{\sigma_t}$ from $\mathsf{Enc}(\alpha^{(t)}) \circ \mathsf{Enc}(\beta^{(t)})$ to $\mathsf{Enc}(\alpha^{(t+1)}) \circ \mathsf{Enc}(\beta^{(t+1)})$

1: **if** $\beta^{(t)} = \beta^{(t+1)}$ **and** $\alpha^{(t)} \neq \alpha^{(t+1)}$ **then**
2: let $(f^{(1)}, \dots, f^{(T')})$ be a reconfiguration sequence from $\mathsf{Enc}(\alpha^{(t)})$ to $\mathsf{Enc}(\alpha^{(t+1)})$ obtained by Lemma 3.5.
3: **return** $(f^{(1)} \circ \mathsf{Enc}(\beta^{(t)}), \dots, f^{(T')} \circ \mathsf{Enc}(\beta^{(t)}))$.
4: **else if** $\alpha^{(t)} = \alpha^{(t+1)}$ **and** $\beta^{(t)} \neq \beta^{(t+1)}$ **then**
5: let $(g^{(1)}, \dots, g^{(T')})$ be a reconfiguration sequence from $\mathsf{Enc}(\beta^{(t)})$ to $\mathsf{Enc}(\beta^{(t+1)})$ obtained by Lemma 3.5.
6: **return** $(\mathsf{Enc}(\alpha^{(t)}) \circ g^{(1)}, \dots, \mathsf{Enc}(\alpha^{(t)}) \circ g^{(T')})$.
7: **else** ▷ *$\alpha^{(t)} = \alpha^{(t+1)}$ and $\beta^{(t)} = \beta^{(t+1)}$.*
8: **return** $(\mathsf{Enc}(\alpha^{(t)}) \circ \mathsf{Enc}(\beta^{(t)}))$.

We claim that any intermediate assignment σ° of $\vec{\sigma_t}$ satisfies Φ. Suppose first that $\beta^{(t)} = \beta^{(t+1)}$ but $\alpha^{(t)} \neq \alpha^{(t+1)}$; namely, σ° is of the form $f^\circ \circ \mathsf{Enc}(\beta^{(t)})$ for some string $f^\circ \in \{0,1\}^{p(n)}$. Since $\delta(f^\circ, \mathsf{Enc}(\cdot)) \leqslant \frac{1}{4}$ by Lemma 3.5, σ° satisfies Eq. (3.10). Observe further that $\delta(f^\circ, \mathsf{Enc}(\gamma)) > \frac{1}{3}$ for every string $\gamma \notin \{\alpha^{(t)}, \alpha^{(t+1)}\}$ by Lemma 3.5, $\delta(\mathsf{Enc}(\beta^{(t)}), \mathsf{Enc}(\gamma)) \geqslant \frac{1}{2}$ for every string $\gamma \neq \beta^{(t)}$, and $C(\alpha^{(t)} \circ \beta^{(t)}) = C(\alpha^{(t+1)} \circ \beta^{(t)}) = 1$ by assumption; i.e., σ° satisfies Eq. (3.11), implying that $\Phi(\sigma^\circ) = 1$. Suppose next that $\alpha^{(t)} = \alpha^{(t+1)}$ but $\beta^{(t)} \neq \beta^{(t+1)}$. Similarly to the first case, we can show that $\Phi(\sigma^\circ) = 1$. Suppose finally that $\alpha^{(t)} = \alpha^{(t+1)}$ and $\beta^{(t)} = \beta^{(t+1)}$, which is a trivial case. Consequently, concatenating $\vec{\sigma_t}$ for every $t \in [T-1]$, we obtain a reconfiguration sequence $\vec{\sigma}$ from σ^{ini} to σ^{end} consisting only of satisfying assignments to Φ, which completes the proof the perfect completeness.

We then prove the robust soundness. Suppose that C is a No instance. Let $\vec{\sigma} = (\sigma^{(1)}, \dots, \sigma^{(T)})$ be any reconfiguration sequence from σ^{ini} to σ^{end}, where each $\sigma^{(t)}$ is of the form $f^{(t)} \circ g^{(t)} \in \{0,1\}^{2p(n)}$. Create then a sequence $\vec{\gamma} = (\alpha^{(1)} \circ \beta^{(1)}, \dots, \alpha^{(T)} \circ \beta^{(T)})$ over $\{0,1\}^{2n}$ by "decoding" $\vec{\sigma}$; namely, $\alpha^{(t)}$ and $\beta^{(t)}$ for each t are defined as follows:

$$\alpha^{(t)} \coloneqq \operatorname*{argmin}_{\alpha \in \{0,1\}^n} \delta\big(f^{(t)}, \mathsf{Enc}(\alpha)\big) \quad \text{and} \quad \beta^{(t)} \coloneqq \operatorname*{argmin}_{\beta \in \{0,1\}^n} \delta\big(g^{(t)}, \mathsf{Enc}(\beta)\big), \tag{3.12}$$

where ties are broken according to any prefixed order over $\{0,1\}^n$. Note that $\alpha^{(1)} = \beta^{(1)} = 0^n$ and $\alpha^{(T)} = \beta^{(T)} = 1^n$. Since $\sigma^{(t)}$ and $\sigma^{(t+1)}$ differ in at most one bit, $\alpha^{(t)} = \alpha^{(t+1)}$ or $\beta^{(t)} = \beta^{(t+1)}$ for every t; i.e., $\vec{\gamma}$ is a valid reconfiguration edge sequence from $0^n \circ 0^n$ to $1^n \circ 1^n$. In particular, $\vec{\gamma}$ must contain $\alpha^{(t)} \circ \beta^{(t)}$ such that $C(\alpha^{(t)} \circ \beta^{(t)}) = 0$.

We will demonstrate that $\sigma^{(t)}$ is $\frac{1}{48}$-far from every satisfying assignment to Φ. Let $\sigma^* = f^* \circ g^* \in \{0,1\}^{2p(n)}$ be any satisfying assignment to Φ. There must exist a pair $\alpha^*, \beta^* \in \{0,1\}^n$ such that $\delta(f^*, \mathsf{Enc}(\alpha^*)) \leqslant \frac{1}{4}$, $\delta(g^*, \mathsf{Enc}(\beta^*)) \leqslant \frac{1}{4}$, and $C(\alpha^* \circ \beta^*) = 1$. Deduce that (1) f^* is $\frac{1}{3}$-far from $\mathsf{Enc}(\alpha^{(t)})$ or (2) g^* is $\frac{1}{3}$-far from $\mathsf{Enc}(\beta^{(t)})$, because otherwise we have $\Phi(f^* \circ g^*) = 0$. Suppose first that $\delta(f^*, \mathsf{Enc}(\alpha^{(t)})) > \frac{1}{3}$, implying that $\alpha^* \neq \alpha^{(t)}$.

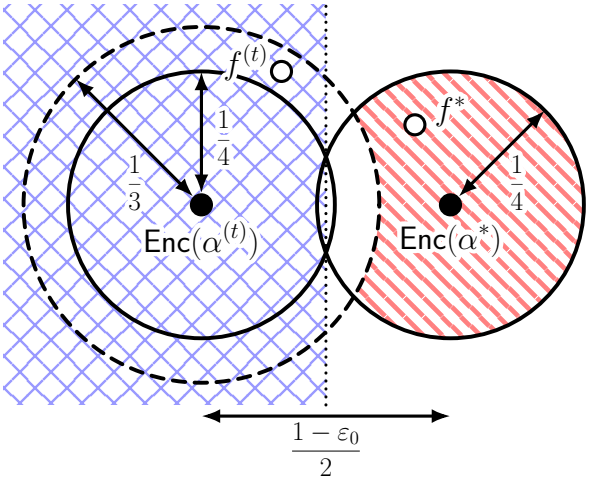

Figure 1 Illustration of the proof of the robust soundness in Lemma 3.7. One one hand, $f^{(t)}$ is closer to $\mathsf{Enc}(\alpha^{(t)})$ than $\mathsf{Enc}(\alpha^*)$ (denoted by blue crosshatch pattern). On the other hand, f^* is $\frac{1}{3}$-far from $\mathsf{Enc}(\alpha^{(t)})$ and $\frac{1}{4}$-close to $\mathsf{Enc}(\alpha^*)$ (denoted by red lines). The two regions must be $\frac{1}{24}$-far from each other.

Putting together, we have the following three inequalities in hand (see Figure 1):

$$\delta\big(f^*, \mathsf{Enc}(\alpha^{(t)})\big) > \frac{1}{3}, \qquad \text{(by assumption)} \tag{3.13}$$

$$\delta\big(f^*, \mathsf{Enc}(\alpha^*)\big) \leqslant \frac{1}{4}, \qquad \text{(by assumption)} \tag{3.14}$$

$$\delta\big(f^{(t)}, \mathsf{Enc}(\alpha^{(t)})\big) \leqslant \delta\big(f^{(t)}, \mathsf{Enc}(\alpha^*)\big). \qquad \text{(by definition of } \alpha^{(t)}) \tag{3.15}$$

Simple calculation using the triangle inequality derives

$$\begin{aligned} \delta\big(f^*, \mathsf{Enc}(\alpha^{(t)})\big) &\leqslant \delta\big(f^*, f^{(t)}\big) + \delta\big(f^{(t)}, \mathsf{Enc}(\alpha^{(t)})\big) \\ &\leqslant \delta\big(f^*, f^{(t)}\big) + \delta\big(f^{(t)}, \mathsf{Enc}(\alpha^*)\big) \\ &\leqslant \delta\big(f^*, f^{(t)}\big) + \delta\big(f^{(t)}, f^*\big) + \delta\big(f^*, \mathsf{Enc}(\alpha^*)\big) \\ &= 2 \cdot \delta\big(f^{(t)}, f^*\big) + \delta\big(f^*, \mathsf{Enc}(\alpha^*)\big), \end{aligned} \tag{3.16}$$

$$\implies \delta\big(f^{(t)}, f^*\big) \geqslant \frac{1}{2} \cdot \Big(\underbrace{\delta\big(f^*, \mathsf{Enc}(\alpha^{(t)})\big)}_{>\frac{1}{3}} - \underbrace{\delta\big(f^*, \mathsf{Enc}(\alpha^*)\big)}_{\leqslant\frac{1}{4}}\Big) > \frac{1}{24}. \tag{3.17}$$

Consequently, $\sigma^{(t)}$ must be $\frac{1}{48}$-far from σ^*. Suppose next that $\delta(g^*, \mathsf{Enc}(\beta^{(t)})) > \frac{1}{3}$. Similarly to the first case, we can show that $\delta(g^{(t)}, g^*) > \frac{1}{24}$, deriving that $\sigma^{(t)}$ is $\frac{1}{48}$-far from σ^*, which completes the proof of the robust soundness. ◀

3.3 Composing Assignment Testers

We build a PCRP system for Circuit SAT Reconfiguration to complete the proof of Theorem 3.1.

Reduction

Our reduction from (a "robust" version of) Circuit SAT Reconfiguration to a PCRP system is described as follows. Given a polynomial-size circuit $\Phi\colon \{0,1\}^{2p(n)} \to \{0,1\}$ and a pair of its satisfying assignments $\sigma^{\mathsf{ini}}, \sigma^{\mathsf{end}} \in \{0,1\}^{p(n)}$ produced by Lemma 3.7, we apply Theorem 2.6 to

obtain an assignment tester $\mathcal{A}$ for Φ with randomness complexity $\mathcal{O}(\log n)$, query complexity $\mathcal{O}(1)$, and rejection rate $\kappa > 0$. The proof length of $\mathcal{A}$, denoted by $\ell(n)$, can be bounded by some polynomial in the size of Φ and thus n. Our PCRP verifier $\mathcal{V}$ is given oracle access to an assignment $\sigma \in \{0,1\}^{2p(n)}$ as implicit input and *twins* of proof $\pi_1, \pi_2 \in \{0,1\}^{\ell(n)}$. Then, $\mathcal{V}$ runs $\mathcal{A}^{\sigma\circ\pi_1}(\Phi)$ and $\mathcal{A}^{\sigma\circ\pi_2}(\Phi)$ independently and accepts if (at least) either of the runs accepted, which is described as follows.

Our PCRP verifier $\mathcal{V}$

Input: a polynomial-size circuit $\Phi\colon \{0,1\}^{2p(n)} \to \{0,1\}$ and an assignment tester $\mathcal{A}$.
Oracle access: an assignment $\sigma \in \{0,1\}^{2p(n)}$ and twins of proof $\pi_1, \pi_2 \in \{0,1\}^{\ell(n)}$.

1: independently run $\mathcal{A}^{\sigma\circ\pi_1}(\Phi)$ and $\mathcal{A}^{\sigma\circ\pi_2}(\Phi)$.
2: **if** either of runs returned 1 **then**
3: **return** 1.
4: **else**
5: **return** 0.

By Theorem 2.6, compute two proofs $\pi^{\mathsf{ini}}, \pi^{\mathsf{end}} \in \{0,1\}^{\ell(n)}$ in polynomial time such that $\mathcal{A}(\Phi)$ accepts $\sigma^{\mathsf{ini}} \circ \pi^{\mathsf{ini}}$ and $\sigma^{\mathsf{end}} \circ \pi^{\mathsf{end}}$ with probability 1. Define two proofs $\Pi^{\mathsf{ini}}, \Pi^{\mathsf{end}} \in \{0,1\}^{2p(n)+2\ell(n)}$ as $\Pi^{\mathsf{ini}} := \sigma^{\mathsf{ini}} \circ \pi^{\mathsf{ini}} \circ \pi^{\mathsf{ini}}$ and $\Pi^{\mathsf{end}} := \sigma^{\mathsf{end}} \circ \pi^{\mathsf{end}} \circ \pi^{\mathsf{end}}$. Observe that $\mathcal{V}$ accepts Π^{ini} and Π^{end} with probability 1, which completes the description of the PCRP system.

Correctness

We show the following perfect completeness and soundness.

▶ **Lemma 3.8.** *The following hold*:

- (*Perfect completeness*) *If C is a* YES *instance, then there exists a reconfiguration sequence $\vec{\Pi}$ from Π^{ini} to Π^{end} such that $\mathcal{V}$ accepts every proof in $\vec{\Pi}$ with probability* 1.
- (*Soundness*) *If C is a* NO *instance, then every reconfiguration sequence $\vec{\Pi}$ from Π^{ini} to Π^{end} contains some proof that is rejected by $\mathcal{V}$ with probability more than $\left(\frac{\kappa}{48}\right)^2$.*

Proof. We first prove the perfect completeness. Suppose that C is a YES instance. By Lemma 3.7, there exists a reconfiguration sequence $\vec{\sigma} = (\sigma^{(1)}, \ldots, \sigma^{(T)})$ from σ^{ini} to σ^{end} such that $\Phi(\sigma^{(t)}) = 1$ for every t. Let $\pi^{(t)}$ be a proof such that $\mathcal{A}(\Phi)$ accepts $\sigma^{(t)} \circ \pi^{(t)}$ with probability 1. In particular, $\pi^{(1)} = \pi^{\mathsf{ini}}$ and $\pi^{(T)} = \pi^{\mathsf{end}}$. Consider a reconfiguration sequence $\vec{\Pi}$ from Π^{ini} to Π^{end} obtained by the following procedure.

Reconfiguration sequence $\vec{\Pi}$ from Π^{ini} to Π^{end}

1: **for each** t from 1 to $T-1$ **do** ▷ *start with* $\sigma^{(t)} \circ \pi^{(t)} \circ \pi^{(t)}$.
2: change the second string from $\pi^{(t)}$ to $\pi^{(t+1)}$. ▷ *obtain* $\sigma^{(t)} \circ \pi^{(t+1)} \circ \pi^{(t)}$.
3: change the first string from $\sigma^{(t)}$ to $\sigma^{(t+1)}$. ▷ *obtain* $\sigma^{(t+1)} \circ \pi^{(t+1)} \circ \pi^{(t)}$.
4: change the third string from $\pi^{(t)}$ to $\pi^{(t+1)}$. ▷ *end with* $\sigma^{(t+1)} \circ \pi^{(t+1)} \circ \pi^{(t+1)}$.

We claim that $\mathcal{V}$ accepts any intermediate proof Π° of $\vec{\Pi}$ with probability 1. By construction, Π° is of the form $\sigma^\circ \circ \pi_1^\circ \circ \pi_2^\circ$ such that $\sigma^\circ = \sigma^{(t)}$ and ($\pi_1^\circ = \pi^{(t)}$ or $\pi_2^\circ = \pi^{(t)}$) for some t. Since $\mathcal{A}(\Phi)$ always accepts $\sigma^{(t)} \circ \pi^{(t)}$, we find $\mathcal{V}$ to accept Π° with probability 1, which completes the proof of the perfect completeness.

We then prove the soundness. Suppose that C is a No instance. Let $\vec{\Pi} = (\Pi^{(1)}, \dots, \Pi^{(T)})$ be any reconfiguration sequence from Π^{ini} to Π^{end}. By Lemma 3.7, $\vec{\Pi}$ contains a proof $\Pi^{(t)} = \sigma^{(t)} \circ \pi_1^{(t)} \circ \pi_2^{(t)}$ such that $\sigma^{(t)}$ is $\frac{1}{48}$-far from every satisfying assignment to Φ. Since $\mathcal{A}(\Phi)$ rejects both $\sigma^{(t)} \circ \pi_1^{(t)}$ and $\sigma^{(t)} \circ \pi_2^{(t)}$ with probability more than $\frac{\kappa}{48}$ by Theorem 2.6, $\mathcal{V}$ rejects $\Pi^{(t)}$ with probability

$$\mathbb{P}\Big[\big(\mathcal{A}(\Phi) \text{ rejects } \sigma^{(t)} \circ \pi_1^{(t)}\big) \text{ and } \big(\mathcal{A}(\Phi) \text{ rejects } \sigma^{(t)} \circ \pi_2^{(t)}\big)\Big] > \Big(\frac{\kappa}{48}\Big)^2, \tag{3.18}$$

which completes the proof of the soundness. ◀

The proof of Theorem 3.1 follows from **PSPACE**-completeness of Succinct Reach (Corollary 3.3) and its PCRP system (Lemma 3.8), where the soundness error can be reduced to $\frac{1}{2}$ by repeating $\mathcal{V}$ a constant number of times.

References

1 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. doi:10.1017/CBO9780511804090.

2 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. doi:10.1145/278298.278306.

3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

4 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM Journal on Computing*, 36(4):889–974, 2006. doi:10.1137/S0097539705446810.

5 Hans L. Bodlaender, Carla Groenland, Jesper Nederlof, and Céline Swennenhuis. Parameterized problems complete for nondeterministic FPT time and logarithmic space. *Information and Computation*, 300:105195, 2024. doi:10.1016/j.ic.2024.105195.

6 Hans L. Bodlaender, Carla Groenland, and Céline M. F. Swennenhuis. Parameterized complexities of dominating and independent set ceconfiguration. In *Proceedings of the International Symposium on Parameterized and Exact Computation (IPEC)*, pages 9:1–9:16, 2021. doi:10.4230/LIPIcs.IPEC.2021.9.

7 Paul Bonsma. The complexity of rerouting shortest paths. *Theoretical Computer Science*, 510:1–12, 2013. doi:10.1016/j.tcs.2013.09.012.

8 Paul Bonsma and Luis Cereceda. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theoretical Computer Science*, 410(50):5215–5226, 2009. doi:10.1016/j.tcs.2009.08.023.

9 Nicolas Bousquet, Amer E. Mouawad, Naomi Nishimura, and Sebastian Siebertz. A survey on the parameterized complexity of reconfiguration problems. *Computer Science Review*, 53:100663, 2024. doi:10.1016/j.cosrev.2024.100663.

10 Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Finding paths between 3-colorings. *Journal of Graph Theory*, 67(1):69–82, 2011. doi:10.1002/jgt.20514.

11 Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.

12 Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM Journal on Computing*, 36(4):975–1024, 2006. doi:10.1137/S0097539705446962.

13 G. David Forney, Jr. *Concatenated Codes*. PhD thesis, Massachusetts Institute of Technology, 1965. URL: http://hdl.handle.net/1721.1/13449.

14 Hana Galperin and Avi Wigderson. Succinct representations of graphs. *Information and Control*, 56(3):183–198, 1983. doi:10.1016/S0019-9958(83)80004-7.

15 Parikshit Gopalan, Phokion G. Kolaitis, Elitza Maneva, and Christos H. Papadimitriou. The connectivity of Boolean satisfiability: Computational and structural dichotomies. *SIAM Journal on Computing*, 38(6):2330–2355, 2009. doi:10.1137/07070440X.

16 Venkatesan Guruswami. *List Decoding of Error-Correcting Codes*. PhD thesis, Massachusetts Institute of Technology, September 2001. URL: http://hdl.handle.net/1721.1/8700.

17 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory, 2019. Draft available at https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/.

18 Arash Haddadan, Takehiro Ito, Amer E. Mouawad, Naomi Nishimura, Hirotaka Ono, Akira Suzuki, and Youcef Tebbal. The complexity of dominating set reconfiguration. *Theoretical Computer Science*, 651:37–49, 2016. doi:10.1016/j.tcs.2016.08.016.

19 Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science*, 343(1-2):72–96, 2005. doi:10.1016/j.tcs.2005.05.008.

20 Robert A. Hearn and Erik D. Demaine. *Games, Puzzles, and Computation*. A K Peters, Ltd., 2009.

21 Shuichi Hirahara and Naoto Ohsaka. Optimal PSPACE-hardness of approximating set cover reconfiguration. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 85:1–85:18, 2024. doi:10.4230/LIPIcs.ICALP.2024.85.

22 Shuichi Hirahara and Naoto Ohsaka. Probabilistically checkable reconfiguration proofs and inapproximability of reconfiguration problems. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 1435–1445, 2024. doi:10.1145/3618260.3649667.

23 Shuichi Hirahara and Naoto Ohsaka. Asymptotically optimal inapproximability of maxmin k-cut reconfiguration. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, 2025. to appear.

24 Duc A. Hoang. Combinatorial Reconfiguration, 2024. URL: https://reconf.wikidot.com/.

25 Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999. doi:10.1007/BF02392825.

26 Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. doi:10.1145/502090.502098.

27 Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12-14):1054–1065, 2011. doi:10.1016/j.tcs.2010.12.005.

28 Marcin Kamiński, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theoretical Computer Science*, 439:9–15, 2012. doi:10.1016/j.tcs.2012.03.004.

29 Karthik C. S. and Pasin Manurangsi. On inapproximability of reconfiguration problems: PSPACE-hardness and some tight NP-hardness results. *Electronic Colloquium on Computational Complexity*, TR24-007, 2023. URL: http://eccc.weizmann.ac.il/report/2024/007.

30 Daniel Lokshtanov and Amer E. Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Transactions on Algorithms*, 15(1):7:1–7:19, 2019. doi:10.1145/3280825.

31 Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, 78(1):274–297, 2017. doi:10.1007/s00453-016-0159-2.

32 Christina M. Mynhardt and Shahla Nasserasr. Reconfiguration of colourings and dominating sets in graphs. In *50 years of Combinatorics, Graph Theory, and Computing*, chapter 10, pages 171–191. Chapman and Hall/CRC, 2019.

33 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018. doi:10.3390/a11040052.

34 Naoto Ohsaka. Gap preserving reductions between reconfiguration problems. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 49:1–49:18, 2023. doi:10.4230/LIPIcs.STACS.2023.49.

35 Naoto Ohsaka. Alphabet reduction for reconfiguration problems. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 113:1–113:17, 2024. doi:10.4230/LIPIcs.ICALP.2024.113.

36 Naoto Ohsaka. Gap amplification for reconfiguration problems. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1345–1366, 2024. doi:10.1137/1.9781611977912.54.

37 Naoto Ohsaka. Tight inapproximability of target set reconfiguration. *Computing Research Repository*, abs/2402.15076, 2024. doi:10.48550/arXiv.2402.15076.

38 Naoto Ohsaka. On approximate reconfigurability of label cover. *Information Processing Letters*, 189:106556, 2025. doi:10.1016/j.ipl.2024.106556.

39 Naoto Ohsaka and Tatsuya Matsuoka. Reconfiguration problems on submodular functions. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, pages 764–774, 2022. doi:10.1145/3488560.3498382.

40 Christos H. Papadimitriou and Mihalis Yannakakis. A note on succinct representations of graphs. *Information and Control*, 71(3):181–185, 1986. doi:10.1016/S0019-9958(86)80009-2.

41 Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960. doi:10.1137/0108018.

42 Jan van den Heuvel. The complexity of change. In *Surveys in Combinatorics 2013*, volume 409, pages 127–160. Cambridge University Press, 2013. doi:10.1017/CBO9781139506748.005.

43 Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *Journal of Computer and System Sciences*, 93:1–10, 2018. doi:10.1016/j.jcss.2017.11.003.