

Unbounded-width CSPs are Untestable in a Sublinear Number of Queries

Yumou Fei*

Abstract

The bounded-degree query model, introduced by Goldreich and Ron (*Algorithmica*, 2002), is a standard framework in graph property testing and sublinear-time algorithms. Many properties studied in this model, such as bipartiteness and 3-colorability of graphs, can be expressed as satisfiability of constraint satisfaction problems (CSPs). We prove that for the entire class of *unbounded-width* CSPs, testing satisfiability requires $\Omega(n)$ queries in the bounded-degree model. This result unifies and generalizes several previous lower bounds. In particular, it applies to all CSPs that are known to be **NP**-hard to solve, including k -colorability of ℓ -uniform hypergraphs for any $k, \ell \geq 2$ with $(k, \ell) \neq (2, 2)$.

Our proof combines the techniques from Bogdanov, Obata, and Trevisan (*FOCS*, 2002), who established the first $\Omega(n)$ query lower bound for CSP testing in the bounded-degree model, with known results from universal algebra.

*Department of EECS, Massachusetts Institute of Technology.

Contents

1	Introduction	3
1.1	Testing CSP Satisfiability	3
1.2	Discussion: Testing vs. Approximation	5
1.2.1	MaxCSP Problems	5
1.2.2	MaxCSP vs. Satisfiability Testing	5
1.2.3	Sublinear Time vs. Polynomial Time	6
1.2.4	Sublinear Time vs. Sublinear Space	7
1.3	Open Problems	7
2	Preliminaries	8
2.1	Relational Structures	8
2.2	Algebraic Structures	10
2.3	Irredundant Relations and Galois Duality	11
3	Main Lemma from Universal Algebra	12
4	Hardness of Linear Equations	14
5	The Reduction Map	16
5.1	The Expander Gadget	16
5.2	Construction in the Core Case	17
5.2.1	Preparatory Materials	17
5.2.2	The Construction Procedure	18
5.3	Analysis of the Construction	19
5.4	Conclusion in the General Case	22
	References	23
A	Definition of Bounded Width	26
B	Allowing Variable Repetition	28
C	Galois Duality Proof	28
D	Random Regular Hypergraphs	30
D.1	Random Regular Hypergraphs are Locally Sparse	30
D.2	Concentration of Value for Random Instances	31
E	Varieties Admitting Affine-type	34
E.1	Varieties	34
E.2	Types of Strictly Simple Algebras	35

1 Introduction

Property testing is a class of algorithmic problems in which the algorithm is allowed to inspect only a small portion of the input. In exchange for this limited access, the goal is relaxed: instead of deciding whether the input exactly satisfies a given property, the tester must merely distinguish inputs that satisfy the property from those that are *far* from satisfying it, under an appropriate notion of distance.

In this paper, we consider the problem of testing satisfiability of constraint satisfaction problems (CSPs). We formally define CSPs as follows. First of all, every CSP considered in this paper is specified by a finite *template*.

Definition 1.1. A *relational structure* is a pair (D, Γ) where D is a finite set, and Γ is a finite collection of relations on D . Each relation $R \in \Gamma$ is a function $R : D^k \rightarrow \{0, 1\}$ for some positive integer k (called the *arity* of R). A fixed relational structure (D, Γ) is also referred to as a *CSP template*.

As a running example, if we let $D = \{0, 1\}$ and take $\Gamma = \{\mathbb{1}_{\{10,01\}}\}$, where $\mathbb{1}_{\{10,01\}}(x, y) = 1$ if and only if $x \neq y$, then the template (D, Γ) could express the bipartiteness property of graphs, as we will soon explain.

Given a CSP template (D, Γ) , we can collect all *instances* of the CSP into an infinite set $\text{CSP}(\Gamma)$.

Definition 1.2. Given a CSP template (D, Γ) , we define $\text{CSP}(\Gamma)$ as the (infinite) collection of all *instances* $I = (V, \mathcal{C})$, where V is a finite set of *variables* and $\mathcal{C} = \{C_1, \dots, C_m\}$ is a multi-set of *constraints* satisfying the following. Each constraint C_i is a pair $(\mathbf{v}, R)$, where $\mathbf{v} = (v_1, \dots, v_k)$ is a tuple of (not necessarily distinct)¹ variables from V , and $R : D^k \rightarrow \{0, 1\}$ is a relation from Γ . The variable set $\{v_1, \dots, v_k\} \subseteq V$ is called the *scope* of C_i .

Now an instance in $\text{CSP}(\{\mathbb{1}_{\{10,01\}}\})$ can be viewed as a graph, with variables as vertices and constraints as edges. The graph is bipartite if and only if the CSP instance is “satisfiable,” as defined in the following definition.

Definition 1.3. An *assignment* for an instance $I = (V, \mathcal{C}) \in \text{CSP}(\Gamma)$ is a map $\tau : V \rightarrow D$. We say τ *satisfies* a constraint $((v_1, \dots, v_k), R)$ if $R(\tau(v_1), \dots, \tau(v_k)) = 1$. The *value* of an assignment τ , denoted by $\text{val}_I(\tau)$, is the fraction of constraints in $\mathcal{C}$ satisfied by τ . Note that $\text{val}_I(\tau) \in [0, 1]$. If $\text{val}_I(\tau) = 1$, we say that τ *satisfies* the instance I , or τ is a *satisfying assignment* of I . The *value* of the instance I , denoted by val_I , is the maximum possible value any assignment may achieve.

The CSP defined by the bipartiteness template $(\{0, 1\}, \{\mathbb{1}_{\{10,01\}}\})$ will be denoted by 2COL. Similarly, the property of vertex 3-colorability in graphs can be expressed by as a constraint satisfaction problem denoted by 3COL.

1.1 Testing CSP Satisfiability

We study the property testing of CSP satisfiability in the *bounded-degree query model*, defined in [Definition 1.5](#) below. In this model, the input CSP instance has a fixed variable set, and the degree of each variable (i.e. the number of constraints involving the variable) in the instance is assumed to be uniformly bounded by a constant.

¹Some authors allow repetition of variables in a constraint. In our setting, one can also “artificially” allow repetitions by adding more relations to the relational structure. See [Section 2.1](#) for details.

Definition 1.4. Let (D, Γ) be a CSP template and let V be a set of variables. We write $\text{CSP}(\Gamma, V) \subseteq \text{CSP}(\Gamma)$ for the collection of all $\text{CSP}(\Gamma)$ -instances whose variable set is V .

Definition 1.5. Fix a CSP template (D, Γ) . In the $\text{BD}(d, n)$ model for testing satisfiability, the tester is given oracle access to a CSP instance $I = ([n], \mathcal{C}) \in \text{CSP}(\Gamma, [n])$ with maximum degree at most d . The goal of an ε -tester, where $\varepsilon > 0$ is a fixed constant, is to achieve the following:

- (1) If I is satisfiable, the tester must accept I with probability at least $2/3$.
- (2) If at least εdn constraints must be removed from $\mathcal{C}$ to make I satisfiable, the tester must reject I with probability at least $2/3$.

The tester accesses the instance I through *oracle queries*: when the tester queries a variable $v \in [n]$, the oracle reveals an arbitrary one of the previously unseen constraints involving v , unless all constraints involving v have been revealed (in which case the oracle returns $\perp$).

The bounded-degree query model was first introduced by [GR02], in the context of graph property testing. One of the earliest results in this model is that testing bipartiteness (i.e. testing satisfiability of 2COL instances) requires at least $\Omega(\sqrt{n})$ queries [GR02] (for some degree bound d and error $\varepsilon > 0$) and can be done with at most $\tilde{O}(\sqrt{n})$ queries [GR99] (for any degree bound d and any error $\varepsilon > 0$). In contrast, Bogdanov, Obata, and Trevisan [BOT02] showed that testing vertex 3-colorability requires $\Omega(n)$ queries.²

The study of testing CSP satisfiability in the bounded-degree model was already formally initiated in the early work of Bogdanov, Obata, and Trevisan [BOT02], who showed that, beyond the specific case of 3COL, several other CSPs (such as 3SAT and 3LIN) require $\Omega(n)$ queries to test in this model. A more systematic investigation was later carried out by Yoshida [Yos11], who proved that any CSP *robustly solved by BasicLP* (the basic linear program) has a satisfiability tester that makes a constant number of queries. Typical examples of such CSPs include 1SAT and 3HornSAT. For the satisfiability testing of any CSP that is not *solved by BasicLP*, [Yos11] further showed that at least $\Omega(\sqrt{n})$ queries are necessary. Examples of such CSPs include 2SAT and 2COL.³

The main result of this paper strengthens this picture by proving that, among these BasicLP-unsolvable CSPs, the subclass of *unbounded-width* CSPs requires $\Omega(n)$ queries to test.

Theorem 1.6. *For any CSP template (D, Γ) of unbounded width, there exist constants $\varepsilon \in (0, 1)$ and $d \in \mathbb{N}$ such that any ε -tester for satisfiability of $\text{CSP}(\Gamma)$ instances in the $\text{BD}(d, n)$ model must make $\Omega_{d, \varepsilon}(n)$ queries.*

The notion of bounded width (formally defined in Appendix A) originates from [FV98] and has been extensively studied in the universal-algebraic approach to CSPs. Bounded-width CSPs can be solved in polynomial time (see Proposition A.2), so the class of unbounded-width CSPs includes all CSPs whose satisfiability is **NP**-hard to decide (unless $\mathbf{P} = \mathbf{NP}$, in which case all CSPs are trivially **NP**-hard). Consequently, Theorem 1.6 implies that all CSPs known to be **NP**-hard are also unconditionally (and maximally) hard to test in the bounded-degree model. This encompasses, for example, the property of k -colorability in ℓ -uniform hypergraphs for all $k, \ell \geq 2$ with $(k, \ell) \neq (2, 2)$.

²These seminal works, along with a large number of subsequent studies (such as [GR11, BSS10, YI10, YI12, CMOS19], to name a few), also investigated a wide range of other graph properties within the bounded-degree model, many of which are not constraint satisfaction problems.

³Robust solvability by BasicLP is a stronger condition than solvability by BasicLP, so [Yos11] left open the possibility of CSPs with superconstant but $o(\sqrt{n})$ query complexity. However, a recent unpublished note by Brady [Bra22] proves that all CSPs solvable by BasicLP are also robustly solvable by it, thereby turning the result of [Yos11] into a full dichotomy theorem.

(see [Remark A.5](#)). Our result thus unifies and extends several prior lower bounds from works such as [\[BOT02, YI10, ACC25\]](#).⁴

1.2 Discussion: Testing vs. Approximation

A property tester can be viewed as an algorithm that approximates the distance of an input to satisfying a property — albeit in a very coarse sense, as it only distinguishes zero distance from sufficiently large distance. Nevertheless, many property testers (for example, the constant-query tester of [\[Yos11\]](#) for certain CSPs) operate by explicitly estimating this distance and checking whether it is close to zero. On the negative side, a hardness result for property testing (such as our [Theorem 1.6](#)) can often be interpreted as a hardness of approximation result. This connection to approximation algorithms was, in fact, one of the original motivations in [\[GGR98\]](#), where graph property testing was first introduced.

In this subsection, we examine and discuss about our main result [Theorem 1.6](#) from the perspective of sublinear-time approximation algorithms.

1.2.1 MaxCSP Problems

A natural approximation problem associated with CSPs is that of estimating the value of a CSP instance (as defined in [Definition 1.3](#)).⁵ This estimation task can be conveniently formulated in the following “gap version.”

Definition 1.7. For a fixed CSP template (D, Γ) , a completeness parameter $c \in (0, 1]$ and a soundness parameter $s \in [0, c)$, the problem $\text{MaxCSP}(\Gamma)[c, s]$ is the promise problem where given an instance $I \in \text{CSP}(\Gamma)$, the (randomized) algorithm should distinguish between the following two cases:

- (1) Yes case: if $\text{val}_I \geq c$, then the algorithm should accept with probability at least $2/3$.
- (2) No case: if $\text{val}_I \leq s$, then the algorithm should reject with probability at least $2/3$.

The complexity of MaxCSP problems has been extensively studied in the context of polynomial-time algorithms, and a central question is to determine for which tuples (Γ, c, s) the problem $\text{MaxCSP}(\Gamma)[c, s]$ can be solved in polynomial time. Notably, Raghavendra [\[Rag08\]](#) almost⁶ characterizes all such tuples, conditioned on the Unique Games Conjecture of [\[Kho02\]](#).

1.2.2 MaxCSP vs. Satisfiability Testing

Whether property testing of CSP satisfiability has implications for MaxCSP problems depends on the notion of “distance to satisfiability” used by the property tester. In the d -bounded-degree model, having distance ε to satisfiability means that at least εdn constraints must be removed to make the instance satisfiable. In contrast, an instance with value $1 - \varepsilon$ in the MaxCSP sense requires removing at least an ε fraction of its constraints. Thus, testing CSP satisfiability in the bounded-degree model is analogous to solving $\text{MaxCSP}(\Gamma)[1, 1 - \varepsilon]$, provided the CSP instance has $\Theta(n)$ constraints.

⁴The recent work of [\[ACC25\]](#) proves that testing k -colorability in k -uniform hypergraphs requires $\Omega(n)$ queries, for any $k \geq 3$.

⁵Ideally, an approximation algorithm would also produce an assignment achieving this value, i.e. one that satisfies as many constraints as possible. However, since we focus on sublinear-time algorithms in this paper, we cannot expect them to explicitly output a complete assignment.

⁶Except for the perfect-completeness case (i.e. the case $c = 1$), which remains a mystery.

However, a CSP instance in the d -bounded-degree model $\text{BD}(d, n)$ may have far fewer than $\Theta(n)$ constraints. In fact, if an instance contains only $O(1)$ constraints, it would be unreasonable to expect any sublinear-time algorithm to approximate its value, since with high probability the queried variables would not appear in any constraint. To address this, we introduce a modified model $\text{BD}^*(d, \alpha, n)$, where $\alpha > 0$ is a fixed constant ensuring that all CSP instances considered contain at least αn constraints.

Under this refinement, [Theorem 1.6](#) can be equivalently stated as follows:

Theorem 1.8. *For any CSP template (D, Γ) of unbounded width, there exist constants $\varepsilon, \alpha \in (0, 1)$ and $d \in \mathbb{N}$ such that any algorithm for $\text{MaxCSP}(\Gamma)[1, 1 - \varepsilon]$ in the $\text{BD}^*(d, \alpha, n)$ model must make $\Omega(n)$ queries.*

It is easy to see that [Theorem 1.8](#) implies [Theorem 1.6](#). Due to the convenience provided by this rephrased version, in the rest of the paper we will work with [Theorem 1.8](#) instead of [Theorem 1.6](#).

1.2.3 Sublinear Time vs. Polynomial Time

As discussed in [Section 1.2.2](#), because of the specific notion of “distance to satisfiability” used in the bounded-degree model, satisfiability testing in this setting most naturally corresponds to MaxCSP problems on instances with $\Theta(n)$ constraints. In contrast, the *dense graph model* (the more extensively studied model in graph property testing) uses another notion of distance, making it better aligned with MaxCSP problems on “dense” instances.

We argue that exactly due to this difference in instance regimes, the bounded-degree model provides a stronger connection to the study of polynomial-time algorithms for MaxCSPs, at least in the respect of worst-case approximation ratios. First, MaxCSP problems on general instances can often be reduced to bounded-degree instances (see, e.g., [\[Tre01\]](#)), so bounded-degree instances already capture much of the hardness in the problems. Second, MaxCSP problems tend to be easier on dense instances (see, e.g., [\[AKK95\]](#)). Indeed, [\[AS03\]](#) shows that every CSP admits a constant-query satisfiability tester in the dense graph model. In contrast, in the bounded-degree model, satisfiability testing already exhibits a wide spectrum of complexities across CSP templates, with query complexity ranging from constant, to $\tilde{\Theta}(\sqrt{n})$, up to $\Theta(n)$.

We next discuss some results about MaxCSP in the polynomial-time setting that are most relevant to this work.

In the context of polynomial-time algorithms, for certain CSP templates (D, Γ) , the *perfect-completeness* problem $\text{MaxCSP}(\Gamma)[1, 1 - \varepsilon]$ can exhibit a markedly different level of complexity from the (potentially harder) *imperfect-completeness* problem $\text{MaxCSP}(\Gamma)[1 - \varepsilon', 1 - \varepsilon]$, even when $\varepsilon' \rightarrow 0$ and ε is fixed. For example, in the case of 3LIN (an unbounded-width CSP), the former problem admits a polynomial-time algorithm via Gaussian elimination, whereas the latter is **NP**-hard (due to [\[Hås01\]](#)). More generally, Dalmau and Krokhin [\[DK13\]](#) proved that the imperfect-completeness problem is **NP**-hard for all unbounded-width CSP templates.

Theorem 1.9 ([\[DK13\]](#)). *For any CSP template (D, Γ) of unbounded width, there exists a constant $\varepsilon \in (0, 1)$ such that for every constant $\varepsilon' \in (0, \varepsilon)$, the problem $\text{MaxCSP}(\Gamma)[1 - \varepsilon', 1 - \varepsilon]$ is **NP**-hard.*

In terms of conclusions, our result is not directly comparable with [Theorem 1.9](#), for multiple reasons. First, [Theorem 1.9](#) is a hardness result for the imperfect completeness problem, while our [Theorem 1.8](#) shows hardness for the (potentially easier) perfect completeness problem. Second, [Theorem 1.9](#) is proved against polynomial-time algorithms, while [Theorem 1.8](#) is against sublinear-query algorithms. Although **NP**-hardness is likely much stronger than a linear query lower bound, the latter does have the benefit of being unconditional. It is also not clear a priori whether some

useful algorithms could have sublinear query complexity but super-polynomial time complexity, so **NP**-hardness usually does not immediately imply query lower bounds in sublinear-time models, even when assuming $\mathbf{P} \neq \mathbf{NP}$.

In terms of techniques, however, the proof of [Theorem 1.8](#) runs in direct parallel to that of [Theorem 1.9](#) in [\[DK13\]](#). While [\[DK13\]](#) constructs a reduction from the **NP**-hardness of approximating 3LIN instances (established by [\[Hås01\]](#)), our proof of [Theorem 1.8](#) builds an analogous reduction from the linear-query lower bound for 3LIN instances (proved in [\[BOT02\]](#)). Unlike in the **NP** setting, where a reduction need only run in polynomial time, our bounded-degree query setting requires addressing several additional subtleties. In particular, certain parts of the proof (for example, [Section 5.1](#)) rely on a careful combination of techniques from both [\[BOT02\]](#) and [\[DK13\]](#).

Remark 1.10. The result of [\[DK13\]](#) ([Theorem 1.9](#)) was later shown to be tight by Barto and Kozik [\[BK16\]](#). Specifically, they proved that for every CSP template (D, Γ) of bounded width and every constant $\varepsilon \in (0, 1)$, there exists a constant $\varepsilon' \in (0, \varepsilon)$ such that the problem $\text{MaxCSP}(\Gamma)[1 - \varepsilon', 1 - \varepsilon]$ belongs to **P**. Together with the result of [\[DK13\]](#), this establishes that bounded width exactly characterizes the class of *robustly solvable* CSPs, as conjectured earlier by Guruswami and Zhou [\[GZ12\]](#).

1.2.4 Sublinear Time vs. Sublinear Space

The past decades have seen a growing body of work on MaxCSP problems in *sublinear-space* algorithmic models (such as [\[KKS14, KK19, CGSV24, CGS+22, SSSV25, FMW25a\]](#), to name a few), particularly in the contexts of sketching and streaming algorithms. Recent studies [\[FMW25b, FMW25a\]](#) reveal a phenomenological connection between the *multi-pass streaming* model and the bounded-degree query model: the class of MaxCSP problems that are easy in the bounded-degree model (admitting algorithms with constantly many queries) almost⁷ coincides with those that are easy in the multi-pass streaming model (admitting algorithms with constantly many passes and polylogarithmic space). We view this connection as further motivation for studying MaxCSP and other approximation problems in the bounded-degree query model.

In light of this connection, a natural question arises: does the hardness result of [Theorem 1.8](#) also extend to the multi-pass streaming model?

Question 1.11. *Is it true that for every CSP template (D, Γ) of unbounded width, there exists a constant $\varepsilon > 0$ such that $\text{MaxCSP}(\Gamma)[1, 1 - \varepsilon]$ requires $\Omega(n)$ space in the multi-pass streaming model?*

We remark that, as discussed in [\[FMW25a\]](#), query lower bounds in sublinear time models are usually easier to establish than space lower bounds, and thus [Theorem 1.8](#) can be seen as a preliminary step towards answering [Question 1.11](#).

1.3 Open Problems

Perhaps the most obvious open problem left by this work is whether the result of [Theorem 1.8](#) is tight, analogous to the case of [Theorem 1.9](#) (see [Remark 1.10](#)).

Question 1.12. *Do sublinear-query algorithms exist for satisfiability testing of all bounded-width CSPs?*

In their original work, Bogdanov, Obata and Trevisan [\[BOT02\]](#) asked whether, given a CSP template (D, Γ) , one can determine the optimal hardness gap ε for which $\text{MaxCSP}(\Gamma)[1, 1 - \varepsilon]$

⁷Except for the perfect-completeness case, which remains a mystery for both models.

requires $\Omega(n)$ queries. This question remains wide open (even for “simple” templates such as 3COL). Here, we pose the following more ambitious version.

Question 1.13. *Determine the query complexity of $\text{MaxCSP}(\Gamma)[c, s]$ in the bounded-degree model, for all predicate family Γ and parameters c and s .*

As discussed in [Section 1.2.4](#), this question may also have strong relevance to sublinear-space computational models. Moreover, resolving [Question 1.13](#) unconditionally (without assuming $\mathbf{P} \neq \mathbf{NP}$ or the Unique Games Conjecture) could possibly shed new light on the corresponding question for polynomial-time algorithms.

2 Preliminaries

As mentioned in [Section 1.2.3](#), the proof of [Theorem 1.8](#) is by constructing a reduction from “linear equation” instances. As is the case in [\[BOT02\]](#), the reduction is centered around the fact that the linear equations can be “simulated” by some constant-size gadgets built from constraints in the target template (D, Γ) . The following general definition captures such gadget constructions.

Definition 2.1. Given a relational structure (D, Γ) and a relation $R : D^k \rightarrow \{0, 1\}$, we say that R is *generated* by Γ if there exists an instance $I = (V, \mathcal{C}) \in \text{CSP}(\Gamma)$ such that the following holds:

- (1) The variable set V is $\{v_1, \dots, v_k, u_1, \dots, u_\ell\}$ for some integer $\ell \geq 0$.
- (2) For any tuple $(x_1, \dots, x_k) \in D^k$, we have $R(x_1, \dots, x_k) = 1$ if and only if there exists a satisfying assignment $\tau : V \rightarrow D$ such that $\tau(v_i) = x_i$ for each $i \in [k]$.

The (possibly infinite) collection of all relations generated by Γ is denoted by $\langle \Gamma \rangle^*$.

Remark 2.2. In the setting of [Definition 2.1](#), we will say that the instance I “simulates” the relation R on the variables $v_1, \dots, v_k$. These variables will sometimes be referred to as “primary variables” later in this paper, while the other variables $u_1, \dots, u_\ell$ in I will be called “secondary variables.”

Over the past decades, the development of the powerful framework of universal algebra has enabled a systematic study of the power and limitations of gadget constructions for CSPs. In contrast to the case-by-case constructions used in earlier works such as [\[BOT02, YI10, ACC25\]](#), we draw on established results in universal algebra to fully exploit the strength of gadget constructions. The universal-algebraic result ([Theorem 3.2](#)) used in the proof of [Theorem 1.8](#) is highly nontrivial (though by now perhaps standard) and serves as the main workhorse that allows us to unify the existing $\Omega(n)$ query lower bounds. The remainder of this section is devoted to introducing the necessary background in universal algebra before formally stating the lemma in the next section.

2.1 Relational Structures

We begin by introducing some basic terminology for studying relational structures.

Recall from [Definition 1.2](#) that we do not allow repetition of variables in individual constraints of CSP instances. For example, if E3SAT is the collection of relations $R : \{0, 1\}^3 \rightarrow \{0, 1\}$ such that $|R^{-1}(1)| = 7$, then $\text{CSP}(\text{E3SAT})$ is the collection of all CNF formulas where each clause contains exactly 3 literals. In order to take into account CNF formulas with some clauses containing less than 3 literals (as is standard in the 3SAT problem), we enlarge the relation set E3SAT into 3SAT

— the collection of relations $R : \{0, 1\}^k \rightarrow \{0, 1\}$ such that $|R^{-1}(1)| = 2^k - 1$ and $k \in \{1, 2, 3\}$. Now CSP(3SAT) expresses the standard version of the 3SAT problem.

Given a general relation set Γ , in order to allow variable repetition in CSP(Γ) instances, we can perform the following operation on Γ .

Definition 2.3. For any relation $R : D^k \rightarrow \{0, 1\}$ on a finite domain D , we let $\overline{\{R\}}$ be the collection of relations $R' : D^\ell \rightarrow \{0, 1\}$ such that there exists a surjective map $\pi : [k] \rightarrow [\ell]$ and $R'(x_1, \dots, x_\ell) = R(x_{\pi(1)}, \dots, x_{\pi(k)})$ for all $x_1, \dots, x_\ell \in D$. For any relational structure (D, Γ) , we let $\bar{\Gamma}$ be the union $\bigcup_{R \in \Gamma} \overline{\{R\}}$. A relational structure (D, Γ) is said to be *repetition-closed*⁸ if $\Gamma = \bar{\Gamma}$.

By a standard reduction, in our main theorem (Theorem 1.8) it suffices to consider relational structures that are repetition-closed. Formally, it will be proved in Appendix B that the following statement implies Theorem 1.8.

Theorem 2.4. *For any repetition-closed CSP template (D, Γ) of unbounded width, there exist constants $\varepsilon, \alpha \in (0, 1)$ and $d \in \mathbb{N}$ such that any algorithm for MaxCSP(Γ)[$1, 1 - \varepsilon$] in the $\text{BD}^*(d, \alpha, n)$ model must make $\Omega(n)$ queries.*

We next turn to study maps between relational structures.

Definition 2.5. Let (D_1, Γ_1) and (D_2, Γ_2) be relational structures. Suppose there exists a bijection between Γ_1 and Γ_2 such that paired relations have the same arity. A map $\varphi : D_1 \rightarrow D_2$ is called a *relational homomorphism* if for every $R_1 \in \Gamma_1$ and $R_2 \in \Gamma_2$ paired under this bijection with arity k , and for all $x_1, \dots, x_k \in D_1$,

$$R_1(x_1, \dots, x_k) = 1 \implies R_2(\varphi(x_1), \dots, \varphi(x_k)) = 1.$$

Throughout the rest of the paper, we consider maps between relational structures only when the relations of the source and target structures are bijectively matched, and corresponding relations have the same arity. Such pairs of structures are said to share the same *signature*.

Definition 2.6. Let (D, Γ) be a relational structure. A map $\varphi : D \rightarrow D$ is called a *relational endomorphism* if it is a relational homomorphism from (D, Γ) to itself (under the obvious identification of relations). A relational endomorphism is called a *relational automorphism* if it is bijective and its inverse is also a relational endomorphism.⁹ A relational structure is called a *core* if all of its relational endomorphisms are relational automorphisms.

The next task is to show that every relational structure can be “reduced” to a core.

Definition 2.7. Two relational structures (D_1, Γ_1) and (D_2, Γ_2) are said to be *homomorphically equivalent* if there exist two relational homomorphisms $\varphi_1 : D_1 \rightarrow D_2$ and $\varphi_2 : D_2 \rightarrow D_1$ (under the same identification between Γ_1 and Γ_2).

Proposition 2.8. Every relational structure (D, Γ) is homomorphically equivalent to a core. If (D, Γ) is repetition-closed, then (D, Γ) is homomorphically equivalent to a repetition-closed core.

Proof. Let $D_1 \subseteq D$ be a smallest subset for which there exists a relational homomorphism φ from (D, Γ) to (D_1, Γ_1) , where Γ_1 is obtained by restricting the relations in Γ to D_1 . Since the natural

⁸This is not standard terminology.

⁹The second condition is actually unnecessary because we only deal with finite structures. In the finite setting, the inverse of a bijective relational endomorphism is automatically an endomorphism.

embedding $\iota : D_1 \hookrightarrow D$ is itself a relational homomorphism, we conclude that (D, Γ) and (D_1, Γ_1) are homomorphically equivalent.

We claim that (D_1, Γ_1) is a core. Otherwise, there exists a relational endomorphism φ_1 on (D_1, Γ_1) that is not bijective. In particular, its image is a proper subset $D_2 \subsetneq D_1$. But then the composition $\varphi_1 \circ \varphi$ defines a homomorphism from D to D_2 , contradicting the minimality of D_1 . Hence (D_1, Γ_1) must be a core.

Now assume (D, Γ) is repetition closed. Any relation in Γ_1 of arity k is the restriction of some relation $R : D^k \rightarrow \{0, 1\}$ to D_1 . Therefore, for any surjection $\pi : [k] \rightarrow [\ell]$, there exists $R' \in \Gamma$ such that $R'(x_1, \dots, x_\ell) = R(x_{\pi(1)}, \dots, x_{\pi(k)})$ for all $x_1, \dots, x_\ell \in D$, and in particular $R'_{|D_1}(x_1, \dots, x_\ell) = R_{|D_1}(x_{\pi(1)}, \dots, x_{\pi(k)})$ for all $x_1, \dots, x_\ell \in D_1$. As $R'_{|D_1} \in \Gamma_1$, we conclude that (D_1, Γ_1) is also repetition closed. $\square$

2.2 Algebraic Structures

Universal algebra enters the picture through the notion of *algebraic structures*.

Definition 2.9. An *algebraic structure* is a pair $(D, \mathcal{F})$ where D is a finite set and $\mathcal{F}$ is a (possibly infinite) collection of operations on D . Each operation $f \in \mathcal{F}$ is a function $f : D^k \rightarrow D$ for some positive integer k (the *arity* of f).

Although at first glance algebraic structures may seem to run parallel to relational structures, the two are in fact deeply interconnected through the following key definitions.

Definition 2.10. Let $f : D^k \rightarrow D$ be an operation, and let $R : D^\ell \rightarrow \{0, 1\}$ be a relation on D . We say that f *preserves* R if for every matrix $(x_{i,j}) \in D^{\ell \times k}$,

$$R(x_{1,j}, \dots, x_{\ell,j}) = 1 \text{ for all } j \in [k]$$

implies

$$R(f(x_{1,1}, \dots, x_{1,k}), \dots, f(x_{\ell,1}, \dots, x_{\ell,k})) = 1.$$

Definition 2.11. An operation f is a *polymorphism* of a relational structure (D, Γ) if it preserves every relation $R \in \Gamma$. The (infinite) set of all polymorphisms of (D, Γ) is denoted by $\text{Pol}(\Gamma)$.

Note that a unary polymorphism is exactly a relational endomorphism, as defined in [Definition 2.5](#).

The following definition gives algebraic structures their own homomorphism maps. It is important not to confuse algebraic homomorphisms with relational homomorphisms.

Definition 2.12. Let $(D_1, \mathcal{F}_1)$ and $(D_2, \mathcal{F}_2)$ be algebraic structures. Suppose there exists a bijection between $\mathcal{F}_1$ and $\mathcal{F}_2$ such that paired functions have the same arity. A map $\varphi : D_1 \rightarrow D_2$ is called an *algebraic homomorphism* if for every $f_1 \in \mathcal{F}_1$ and $f_2 \in \mathcal{F}_2$ paired under this bijection with arity k , and for all $x_1, \dots, x_k \in D_1$,

$$\varphi(f_1(x_1, \dots, x_k)) = f_2(\varphi(x_1), \dots, \varphi(x_k)).$$

Definition 2.13. Let $(D_1, \mathcal{F}_1)$ and $(D_2, \mathcal{F}_2)$ be algebraic structures.

- (1) We say that $(D_1, \mathcal{F}_1)$ is a subalgebra of $(D_2, \mathcal{F}_2)$ if there exists an injective algebraic homomorphism $\varphi : D_1 \rightarrow D_2$.
- (2) We say that $(D_2, \mathcal{F}_2)$ is a homomorphic image of $(D_1, \mathcal{F}_1)$ if there exists a surjective algebraic homomorphism $\varphi : D_1 \rightarrow D_2$.

It turns out that a special class of algebraic structures, namely the *idempotent* ones, provides significant convenience in various contexts. Since idempotent structures play an important role in formulating the main universal-algebraic lemma (Lemma 3.4), we give the definition below.

Definition 2.14. An operation $f : D^k \rightarrow D$ is *idempotent* if $f(x, x, \dots, x) = x$ holds for all $x \in D$. We call algebraic structure $(D, \mathcal{F})$ idempotent if every operation $f \in \mathcal{F}$ is idempotent.

2.3 Irredundant Relations and Galois Duality

Given a relational structure (D, Γ) , let $\langle \Gamma \rangle$ denote the (possibly infinite) set of relations that are preserved by $\text{Pol}(\Gamma)$. It is not hard to see that every relation generated by Γ in the sense of Definition 2.1 is preserved by $\text{Pol}(\Gamma)$.¹⁰ Using the notation of Definition 2.1, this implies $\langle \Gamma \rangle^* \subseteq \langle \Gamma \rangle$.

A fundamental result of Geiger [Gei68] shows that if the notion of “relations generated by Γ ” in Definition 2.1 is suitably relaxed, then the resulting set $\langle \Gamma \rangle^*$ actually coincides with $\langle \Gamma \rangle$. This result captures a deep “duality” between algebraic and relational structures (an instance of the more general concept of a *Galois connection*), which we do not elaborate on here. Instead, following Dalmau and Krokhn [DK13], we state a variant of Geiger’s theorem asserting that any *irredundant* relation in $\langle \Gamma \rangle$ is contained in $\langle \Gamma \rangle^*$ (without relaxing Definition 2.1).

Definition 2.15. Let $R : D^k \rightarrow \{0, 1\}$ be a relation on a finite set D . We say that R is *irredundant* if for any distinct indices $i, j \in [k]$, there exists a tuple $(x_1, \dots, x_k) \in D^k$ such that $R(x_1, \dots, x_k) = 1$ and $x_i \neq x_j$.

Theorem 2.16 ([Gei68]). *Let (D, Γ) be a repetition-closed relational structure, and let $R : D^k \rightarrow \{0, 1\}$ be an irredundant relation on D . If every polymorphism of (D, Γ) preserves R , then $R \in \langle \Gamma \rangle^*$.*

For the sake of completeness, a proof of Theorem 2.16 is given in Appendix C.

As an application of Theorem 2.16, we next define an “Endomorphism” relation for any given relational structure (D, Γ) and prove that it belongs to $\langle \Gamma \rangle^*$.

Definition 2.17. For any relational structure (D, Γ) , we define a relation $\text{End}_\Gamma : D^D \rightarrow \{0, 1\}$ as follows. For any tuple $y = (y_x)_{x \in D} \in D^D$, we let $\text{End}_\Gamma(y) = 1$ if and only if the map $x \mapsto y_x$ is a relational endomorphism on (D, Γ) .

Proposition 2.18. For any repetition-closed relational structure (D, Γ) , we have $\text{End}_\Gamma \in \langle \Gamma \rangle^*$.

Proof. It is easy to see that End_Γ is irredundant, since the tuple $y \in D^D$ defined by $y_x = x$ for all $x \in D$ satisfies the relation End_Γ . It suffices to show that End_Γ is preserved by all polymorphisms of Γ . Once we have that, Theorem 2.16 implies that $\text{End}_\Gamma \in \langle \Gamma \rangle^*$.

Suppose $f \in \text{Pol}(\Gamma)$ is a k -ary polymorphism. For any k tuples $y^{(1)}, \dots, y^{(k)} \in D^D$ satisfying End_Γ , there are endomorphisms $\varphi_1, \dots, \varphi_k$ of (D, Γ) such that $\varphi_i(x) = y_x^{(i)}$ for all $i \in [k]$ and $x \in D$. It now suffices to verify that the tuple

$$\left(f(\varphi_1(x), \dots, \varphi_k(x)) \right)_{x \in D} \in D^D$$

satisfies End_Γ , or in other words, the map $x \mapsto f(\varphi_1(x), \dots, \varphi_k(x))$ is an relational endomorphism of (D, Γ) . But this is a special case of the general observation that for any relational structure, the collection of its polymorphisms is closed under composition.¹¹ $\square$

¹⁰This observation is not used elsewhere in this paper.

¹¹In general, a k -ary polymorphism f composed with an m -ary polymorphism g yields a km -ary polymorphism $f \circ g$ defined by $f \circ g(x_{1,1}, \dots, x_{1,m}, \dots, x_{k,1}, \dots, x_{k,m}) = f(g(x_{1,1}, \dots, x_{1,m}), \dots, g(x_{k,1}, \dots, x_{k,m}))$

3 Main Lemma from Universal Algebra

In this section, we state the main universal-algebraic result used in the proof of [Theorem 2.4](#). Informally, the result asserts that any CSP template of unbounded width can simulate relations defined by linear equations over abelian groups, in the sense of [Definition 2.1](#). The formal statement, however, is much more nuanced.

We begin by defining 3SUM_G , the set of relations that specify the sum of exactly three variables. Throughout the rest of the paper, Abelian groups denoted by G are assumed to be finite and have at least two elements.

Definition 3.1. Let G be an Abelian group. For any element $b \in G$, we let $S_b^G : G^3 \rightarrow \{0, 1\}$ be the relation defined by

$$S_b^G(x_1, x_2, x_3) = 1 \quad \text{if and only if} \quad x_1 + x_2 + x_3 = b.$$

We use 3SUM_G to denote the relation set $\{S_b^G \mid b \in G\}$.

We are now ready to state the following main result from universal algebra.

Theorem 3.2 ([BK14]). *Let (D, Γ) be a relational structure of unbounded width. Assume further that (D, Γ) is a core and that all operations in $\text{Pol}(\Gamma)$ are idempotent. Then there exists an Abelian group G and a family of operations $\mathcal{F} \subseteq \text{Pol}(3\text{SUM}_G)$ such that $(G, \mathcal{F})$ is a homomorphic image of a subalgebra of the polymorphism algebra $(D, \text{Pol}(\Gamma))$.*

The statement of [Theorem 3.2](#) is primarily derived from [BK14, Conjecture 4.3], with additional ingredients drawn from other works, such as [Val09]. Further details concerning [Theorem 3.2](#) will be provided in [Appendix E](#).

The remaining goal of this section is to translate [Theorem 3.2](#) into the more explicit and directly applicable statement of [Lemma 3.4](#). One aspect of [Theorem 3.2](#) that requires “translation” is the idempotency condition on $\text{Pol}(\Gamma)$. Given a relational structure, its polymorphism algebra can be made idempotent by augmenting the relation set with all constant relations, defined as follows.

Definition 3.3. Let D be a finite set. We denote by CONST_D the set of all unary constant relations on D ; that is, relations R of the form

$$R(x) = 1 \quad \text{if and only if} \quad x = b,$$

for some $b \in D$.

We are now ready to state and prove the following translated version of [Theorem 3.2](#).

Lemma 3.4. *Let (D, Γ) be a repetition-closed relational structure of unbounded width. Then there exist a subset $D' \subseteq D$, an Abelian group G , and a surjective map $\varphi : D' \rightarrow G$ such that any ternary relation $R : D^3 \rightarrow \{0, 1\}$ of the form*

$$R(x_1, x_2, x_3) = 1 \quad \text{if and only if} \quad x_1, x_2, x_3 \in D' \text{ and } R'(\varphi(x_1), \varphi(x_2), \varphi(x_3)) = 1, \quad (3.1)$$

for some $R' \in 3\text{SUM}_G$, belongs to the collection $\langle \Gamma \cup \text{CONST}_D \rangle^*$.

Proof. We divide the proof into the following steps.

Step 1: construction of φ . Because of the presence of constant relations, every relational endomorphism of the structure $(D, \Gamma \cup \text{CONST}_D)$ is the identity map. By [Definition 2.6](#), this implies that $(D, \Gamma \cup \text{CONST}_D)$ is a core. Furthermore, by [Proposition A.4](#), $(D, \Gamma \cup \text{CONST}_D)$ also has unbounded width. Since every polymorphism of $(D, \Gamma \cup \text{CONST}_D)$ preserves the constant relations and is therefore idempotent, we can apply [Theorem 3.2](#) to obtain a subalgebra $(D_1, \mathcal{F}_1)$ of $(D, \text{Pol}(\Gamma \cup \text{CONST}_D))$ and a homomorphic image $(G, \mathcal{F}_2)$ of $(D_1, \mathcal{F}_1)$ such that $\mathcal{F}_2 \subseteq \text{Pol}(3\text{SUM}_G)$, as illustrated in the following diagram.

$$\begin{array}{ccc} & & (D, \text{Pol}(\Gamma \cup \text{CONST}_D)) \\ & \nearrow \varphi_1 & \\ (D_1, \mathcal{F}_1) & & \\ & \searrow \varphi_2 & \\ & & (G, \mathcal{F}_2) \end{array}$$

Let $\varphi_1 : D_1 \rightarrow D$ be the injective algebraic homomorphism from $(D_1, \mathcal{F}_1)$ to $(D, \text{Pol}(\Gamma \cup \text{CONST}_D))$, and let $\varphi_2 : D_1 \rightarrow G$ be the surjective algebraic homomorphism from $(D_1, \mathcal{F}_1)$ to $(G, \mathcal{F}_2)$. Let $D' = \varphi_1(D_1) \subseteq D$ be the image of φ_1 . Then we may define a surjective map $\varphi : D' \rightarrow G$ by letting $\varphi(x) = \varphi_2(\varphi_1^{-1}(x))$ for any $x \in D'$.

Step 2: properties of φ . We claim that any polymorphism $f : D^k \rightarrow D$ of the relational structure $(D, \Gamma \cup \text{CONST}_D)$ also preserves any relation R of the form [\(3.1\)](#). Since φ_1, φ_2 are algebraic homomorphisms, by [Definition 2.12](#) we know that some $f_1 \in \mathcal{F}_1$ is implicitly paired with f and some $f_2 \in \mathcal{F}_2$ is in turn paired with f_1 .

Let $(x_{i,j}) \in D^{3 \times k}$ be a matrix of elements such that $R(x_{1,j}, x_{2,j}, x_{3,j}) = 1$ for all $j \in [k]$. Therefore, by [\(3.1\)](#) it must be the case that $x_{\ell,j} \in D'$ for all $\ell \in [3]$ and $j \in [k]$. We let $y_{\ell,j} = \varphi_1^{-1}(x_{\ell,j}) \in D_1$. Now we have

$$\begin{aligned} & R(f(x_{1,1}, \dots, x_{1,k}), f(x_{2,1}, \dots, x_{2,k}), f(x_{3,1}, \dots, x_{3,k})) \\ &= R(f(\varphi_1(y_{1,1}), \dots, \varphi_1(y_{1,k})), f(\varphi_1(y_{2,1}), \dots, \varphi_1(y_{2,k})), f(\varphi_1(y_{3,1}), \dots, \varphi_1(y_{3,k}))) \\ &= R'(\varphi_2 \circ \varphi_1^{-1}(f(\varphi_1(y_{1,1}), \dots, \varphi_1(y_{1,k}))), \varphi_2 \circ \varphi_1^{-1}(f(\varphi_1(y_{2,1}), \dots, \varphi_1(y_{2,k}))), \\ &\quad \varphi_2 \circ \varphi_1^{-1}(f(\varphi_1(y_{3,1}), \dots, \varphi_1(y_{3,k})))) \\ &= R'(\varphi_2(f_1(y_{1,1}, \dots, y_{1,k})), \varphi_2(f_1(y_{2,1}, \dots, y_{2,k})), \varphi_2(f_1(y_{3,1}, \dots, y_{3,k}))) \\ &= R'(\varphi_2(\varphi_2(y_{1,1}), \dots, \varphi_2(y_{1,k})), \varphi_2(\varphi_2(y_{2,1}), \dots, \varphi_2(y_{2,k})), \varphi_2(\varphi_2(y_{3,1}), \dots, \varphi_2(y_{3,k}))) \\ &= R'(\varphi_2(\varphi(x_{1,1}), \dots, \varphi(x_{1,k})), \varphi_2(\varphi(x_{2,1}), \dots, \varphi(x_{2,k})), \varphi_2(\varphi(x_{3,1}), \dots, \varphi(x_{3,k}))) \end{aligned} \quad (3.2)$$

Since $\mathcal{F}_2 \subseteq \text{Pol}(3\text{SUM}_G)$ and $f_2 \in \mathcal{F}_2$, it follows from the assumption $R' \in 3\text{SUM}_G$ that f_2 preserves R' . Combining this with the observation that

$$R'(\varphi(x_{1,j}), \varphi(x_{2,j}), \varphi(x_{3,j})) = R(x_{1,j}, x_{2,j}, x_{3,j}) = 1 \quad \text{for all } j \in [k],$$

we deduce that [\(3.2\)](#) evaluates to 1. Therefore, we conclude that f preserves R .

Step 3: applying Galois duality. Since every relation $R' \in 3\text{SUM}_G$ is irredundant in the sense of [Definition 2.15](#), it is easy to see that any relation $R : D^3 \rightarrow \{0, 1\}$ defined according to [\(3.1\)](#) is also irredundant. Due to [Theorem 2.16](#) and the claim proved in Step 2 of the proof, it follows that $R \in \langle \Gamma \cup \text{CONST}_D \rangle^*$. $\square$

4 Hardness of Linear Equations

As mentioned earlier, we prove [Theorem 2.4](#) by constructing a reduction from the following hardness result for the “linear equation” template $(\mathbb{G}, 3\text{SUM}_{\mathbb{G}})$.

Theorem 4.1 ([BOT02]). *For any Abelian group $\mathbb{G}$ and any $\varepsilon > 0$, there exists a constant $d > 0$ such that $\text{MaxCSP}(3\text{SUM}_{\mathbb{G}})[1, \frac{1}{|\mathbb{G}|} + \varepsilon]$ requires $\Omega(n)$ queries in the $\text{BD}^*(d, \frac{d}{3}, 3n)$ model.¹²*

However, unlike [DK13], which applies the result of [Hås01] as a black box to prove [Theorem 1.9](#), we cannot directly use [Theorem 4.1](#) due to the subtleties of our bounded-degree query model. In this section, we unpack the proof of [Theorem 4.1](#) and extract the components that can be explicitly leveraged in proving [Theorem 2.4](#). For the sake of convenience, several aspects of the proof in [BOT02] are slightly modified in the presentation below.

The proof of [Theorem 4.1](#) proceeds by defining two probability distributions over $\text{CSP}(3\text{SUM}_{\mathbb{G}})$ instances, referred to as the YES and NO distributions. Instances drawn from the YES distribution typically have value 1, while those from the NO distribution typically have value at most $\frac{1}{|\mathbb{G}|} + \varepsilon$. The goal is to show that any sublinear-query algorithm cannot reliably distinguish between instances sampled from these two distributions.

A combinatorial structure used in both the YES and NO distributions is hypergraph perfect matching, formally defined as follows.

Definition 4.2. Let n be a positive integer, and let D be a finite set of cardinality $k \geq 2$. Consider the complete k -partite k -uniform hypergraph with vertex set $[n] \times D$, partitioned according to the second coordinate. We write $\mathcal{M}_{D,n}$ for the collection of all perfect matchings of this hypergraph (a *perfect matching* is a set of n hyperedges, no two of which share a common vertex). In other words, a matching $M \in \mathcal{M}_{D,n}$ can be specified by k permutations $\pi_x : [n] \rightarrow [n]$, for $x \in D$, where M is the edge set

$$M = \left\{ \left((\pi_x(i), x) \right)_{x \in D} \in \prod_{x \in D} ([n] \times \{x\}) \mid i \in [n] \right\}$$

A perfect matching $M \in \mathcal{M}_{D,n}$ serves as a guide for placing constraints on variables. Viewing the vertex set $[n] \times D$ as a collection of variables, each hyperedge of M corresponds to a tuple of variables on which a constraint can be imposed. In the setting of [Theorem 4.1](#), we take $D = [3]$, and on each hyperedge (v_1, v_2, v_3) we place a constraint from $3\text{SUM}_{\mathbb{G}}$.

The following definition specify the YES and NO distributions used to prove [Theorem 4.1](#).

Definition 4.3. For any positive integers n, d and any Abelian group $\mathbb{G}$, we define $\mathcal{D}_{\text{pair}}^{\mathbb{G}, n, d}$ to be the distribution over pairs of $\text{CSP}(3\text{SUM}_{\mathbb{G}}, [n] \times [3])$ instances obtained by sampling from the following procedure:

1. Sample an assignment $\tau : [n] \times [3] \rightarrow \mathbb{G}$ uniformly at random.
2. Initialize empty instances $I_{\text{yes}}, I_{\text{no}}$. Repeat d times:
 - Sample uniformly at random a 3-partite perfect matching $M \in \mathcal{M}_{[3], n}$.
 - For each hyperedge $(v_1, v_2, v_3) \in M$:
 - Add to I_{yes} the constraint $((v_1, v_2, v_3), S_b^{\mathbb{G}})$, where $b = \tau(v_1) + \tau(v_2) + \tau(v_3)$.

¹²The paper [BOT02] only proved [Theorem 4.1](#) in the case $\mathbb{G} = \mathbb{Z}/2\mathbb{Z}$, but the proof can easily be adapted to work for general Abelian groups.

– Sample b' uniformly from $\mathbf{G}$, and add to I_{no} the constraint $((v_1, v_2, v_3), S_{b'}^{\mathbf{G}})$.

3. Output the pair $(I_{\text{yes}}, I_{\text{no}})$.

The marginal distributions of $\mathcal{D}_{\text{pair}}^{\mathbf{G}, n, d}$ on the first and second components are denoted by $\mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}$ and $\mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}$, respectively.

By construction, the distributions $\mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}$ and $\mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}$ satisfy the properties stated in the following proposition.

Proposition 4.4. In every instance I in the support of $\mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}$ or $\mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}$, each variable $v \in [n] \times [3]$ is involved in exactly d constraints. Furthermore, every instance I in the support of $\mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}$ is satisfiable.

The next lemma shows that instances sampled from the NO distribution typically have value at most $\frac{1}{|\mathbf{G}|} + \varepsilon$, as expected.

Lemma 4.5 ([BOT02]). *For any Abelian group $\mathbf{G}$ and any $\varepsilon > 0$, there exist a constant integer $d \geq 1$ such that*

$$\mathbb{P}_{I \sim \mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}} \left[\text{val}_I \geq \frac{1}{|\mathbf{G}|} + \varepsilon \right] = o(1),$$

where $o(1)$ denotes a term tending to 0 as $n \rightarrow \infty$.

It remains to show that $\mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}$ and $\mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}$ are indistinguishable to sublinear-query algorithms. The idea is that, although a random instance from the YES distribution is globally very different from a random instance from the NO distribution, they are locally very similar. Specifically, when one inspects only a sublinear-sized portion of the instances, the two distributions are actually almost indistinguishable. The following definition helps formalize this intuition.

Definition 4.6. Let (D, Γ) be a CSP template and let V be a set of variables. We write $\text{CSP}(\Gamma, V) \subseteq \text{CSP}(\Gamma)$ for the collection of all $\text{CSP}(\Gamma)$ -instances whose variable set is exactly V . For any subset $U \subseteq V$, we define the *sub-instance map*

$$\text{Sub}_U : \text{CSP}(\Gamma, V) \longrightarrow \text{CSP}(\Gamma, U)$$

to be the operation that retains only those constraints whose scopes are contained in U .

The indistinguishability between $\mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}$ and $\mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}$ is formalized in the next lemma.

Lemma 4.7 ([BOT02]). *For any Abelian group $\mathbf{G}$ and any positive integer d , there exist a constant $\delta \in (0, 1)$ such that the following holds. For any positive integer n , there exists an event $\mathcal{E}$ with $\mathcal{D}_{\text{pair}}^{\mathbf{G}, n, d}(\mathcal{E}) \geq 1 - o(1)$ such that for any $U \subseteq [n] \times [3]$ with $|U| \leq \delta n$ and any subset $\Lambda \subseteq \text{CSP}(\Gamma, U)$, we have*

$$\mathbb{P}_{(I_{\text{yes}}, I_{\text{no}}) \sim \mathcal{D}_{\text{pair}}^{\mathbf{G}, n, d}} \left[\text{Sub}_U[I_{\text{yes}}] \in \Lambda \mid \mathcal{E} \right] = \mathbb{P}_{(I_{\text{yes}}, I_{\text{no}}) \sim \mathcal{D}_{\text{pair}}^{\mathbf{G}, n, d}} \left[\text{Sub}_U[I_{\text{no}}] \in \Lambda \mid \mathcal{E} \right]$$

It is not hard to deduce Theorem 4.1 from Proposition 4.4 and Lemmas 4.5 and 4.7. For the sake of completeness, proofs of Lemmas 4.5 and 4.7 will be sketched in Appendix D.

5 The Reduction Map

We now arrive at the main technical part of this paper. In this section, we construct the YES and NO distributions for [Theorem 2.4](#) using the distributions $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ introduced in [Section 4](#). The construction for the case where (D, Γ) is a core is presented in [Sections 5.1](#) and [5.2](#) and analyzed in [Section 5.3](#). Finally, in [Section 5.4](#), we complete the proof of [Theorem 2.4](#) by extending the argument to the general case.

5.1 The Expander Gadget

Recall that our universal-algebraic lemma ([Lemma 3.4](#)) guarantees the simulation of “linear equation constraints” only by $\Gamma \cup \text{CONST}_D$, rather than by Γ alone. Consequently, we must first find a way to simulate the constant relations using only the relations in Γ . Note that constant relations may not be generated by Γ in the sense of [Definition 2.1](#), so different techniques are required. Interestingly, the original work of [\[BOT02\]](#) encountered the same difficulty when proving the hardness of testing graph 3-colorability.¹³ To address this issue, [\[BOT02\]](#) uses an “expander gadget” to fix the assignments of certain variables — a technique that also plays a crucial role in our proof.

In contrast to [\[BOT02\]](#), where ordinary expander graphs suffice, our more general setting requires the use of *random regular hypergraphs*, defined formally below.

Definition 5.1. Let n, ℓ be positive integers, and let D be a finite set of cardinality $k \geq 2$. Let $\mathcal{H}_{D,n,\ell}$ denote the distribution of the random ℓ -regular k -partite hypergraph¹⁴ on the vertex set $[n] \times D$, whose edge multiset is the union of ℓ independent perfect matchings chosen uniformly at random from $\mathcal{M}_{D,n}$.

Similarly to [Definition 4.3](#), the hyperedges of the perfect matchings sampled in [Definition 5.1](#) are exactly the variable tuples on which we impose constraints.

Definition 5.2. Let D be a finite set of cardinality $k \geq 2$, and let $R : D^D \rightarrow \{0, 1\}$ be a relation. Any d -regular k -partite hypergraph H in the support of $\mathcal{H}_{D,n,\ell}$ uniquely defines a CSP($\{R\}, [n] \times D$)-instance $\mathcal{I}[R, H]$, by viewing each hyperedge $\mathbf{v} = (v_x)_{x \in D}$ as a constraint $(\mathbf{v}, R)$.

Ideally, we would like to use a relation $R : D^D \rightarrow \{0, 1\}$ that evaluates to 1 on a tuple $y = (y_x)_{x \in D} \in D^D$ only when $y_x = x$ for all $x \in D$. In this case, the instance $\mathcal{I}[R, H]$ would admit a unique satisfying assignment that assigns to each variable $(i, x) \in [n] \times D$ its second coordinate, yielding a gadget that effectively simulates constant relations across many variables. However, for many templates (D, Γ) , such a strict relation cannot be generated in the sense of [Definition 2.1](#). Instead, we settle for weaker relations: it suffices to use a relation $R : D^D \rightarrow \{0, 1\}$ whose satisfying assignments are permutations of D . Such relations are formally defined below.

Definition 5.3. Let D be a finite set. A relation $R : D^D \rightarrow \{0, 1\}$ is called *sub-unique* if any tuple $y = (y_x)_{x \in D} \in D^D$ such that $R(y) = 1$ has pairwise distinct components, i.e. $y_x \neq y_{x'}$ for any distinct elements $x, x' \in D$.

Intuitively speaking, when $R : D^D \rightarrow \{0, 1\}$ is sub-unique and the hypergraph H has a good “expansion” property, any assignment that nearly satisfies $\mathcal{I}[R, H]$ must assign to almost all variables $(i, x) \in [n] \times D$ the value $\pi(x)$, for some permutation $\pi : D \rightarrow D$. The next lemma asserts

¹³We now view this need for constant relations in [\[BOT02\]](#) as a special case of the general phenomenon described in [Lemma 3.4](#), since 3-colorability is an example of an unbounded-width CSP.

¹⁴Multi-edges are allowed.

that a random regular hypergraph H has this desired “expansion” property with high probability. Therefore, the instance $\mathcal{I}[R, H]$ can serve as a gadget that effectively simulates constant relations across many variables, up to only a global permutation.

Lemma 5.4. *Let D be a finite set with at least 2 elements, and let $R : D^D \rightarrow \{0, 1\}$ be a sub-unique relation. For any $\varepsilon \in (0, \frac{1}{10|D|}]$, there exists a constant integer $\ell \geq 1$ such that with probability $1 - \exp(-\Omega(n))$, a random hypergraph $H \sim \mathcal{H}_{D,n,\ell}$ satisfies the following. For any assignment $\tau : [n] \times D \rightarrow D$ to the instance $\mathcal{I}[R, H]$ with $\text{val}_{\mathcal{I}[R,H]}(\tau) \geq 1 - \varepsilon$, there exists a tuple $y \in D^D$ such that $R(y) = 1$ and*

$$\mathbb{P}_{(i,x) \in [n] \times D} [\tau((i, x)) \neq y_x] \leq 2\varepsilon. \quad (5.1)$$

A proof of Lemma 5.4 is given in Appendix D.

5.2 Construction in the Core Case

Recall that the expander gadget constructed in Section 5.1 relies on the availability of a sub-unique relation. However, some CSP templates cannot generate any sub-unique relation (in the sense of Definition 2.1). Fortunately, when the template (D, Γ) is a core (see Definition 2.6), the “Endomorphism” relation provides a natural sub-unique relation that can be generated from Γ , as established in Proposition 2.18.¹⁵

In this subsection, we construct the YES and NO distributions for Theorem 2.4 in the case the template is a core. Throughout this subsection, we fix a repetition-closed relational structure (D, Γ) , and assume that it is a core and has unbounded width. In particular, this implies $|D| \geq 2$.

5.2.1 Preparatory Materials

We begin by gathering several preparatory materials that we are going to use in the construction. A checklist of these materials is presented below.

- (1) By applying Lemma 3.4, we obtain a subset $D' \subseteq D$, an Abelian group G , and a surjective map $\varphi : D' \rightarrow \mathsf{G}$. For any $b \in \mathsf{G}$, we let $R_b : D^3 \rightarrow \{0, 1\}$ be the relation defined by

$$R_b(x_1, x_2, x_3) = 1 \quad \text{if and only if} \quad x_1, x_2, x_3 \in D' \text{ and } S_b^{\mathsf{G}}(\varphi(x_1), \varphi(x_2), \varphi(x_3)) = 1.$$

- (2) The conclusion of Lemma 3.4 ensures that $R_b \in \langle \Gamma \cup \text{CONST}_D \rangle^*$ for all $b \in \mathsf{G}$. Let r_1 be a positive integer such that for each $b \in \mathsf{G}$, the relation R_b is simulated by an instance $\mathcal{I}_b \in \text{CSP}(\Gamma \cup \text{CONST}_D)$ with at most r_1 secondary variables and at most r_1 constraints (see Definition 2.1). Let $\mathcal{I}_b^{\natural} \in \text{CSP}(\Gamma)$ be the instance obtained by removing all constant constraints from $\mathcal{I}_b$.
- (3) We then apply Lemma 4.5 and obtain a positive integer d such that a random $\text{CSP}(3\text{SUM}_{\mathsf{G}})$ -instance from $\mathcal{D}_{\text{no}}^{\mathsf{G},n,d}$ has value at most $2/3$ with high probability.
- (4) By Proposition 2.18, for some positive integer r_2 , the relation End_{Γ} is simulated by an instance $\mathcal{I}_{\text{End}} \in \text{CSP}(\Gamma)$ with at most r_2 secondary variables and at most r_2 constraints, in the sense of Definition 2.1.
- (5) Apply Lemma 5.4 with $\varepsilon = 1/(10r_1|D|)$ and $R = \text{End}_{\Gamma}$,¹⁶ which yields a positive integer ℓ and a hypergraph $H \in \text{supp}(\mathcal{H}_{D,r_1n,\ell})$ satisfying the property stated in Lemma 5.4.

¹⁵The idea of using the Endomorphism relation is due to [DK13].

¹⁶The relation $\text{End}_{\Gamma} : D^D \rightarrow \{0, 1\}$ is sub-unique because (D, Γ) is a core (see Definition 2.6).

5.2.2 The Construction Procedure

Given a $\text{CSP}(3\text{SUM}_G, [n] \times [3])$ -instance $I \in \text{supp}(\mathcal{D}_{\text{yes}}^{G,n,d}) \cup \text{supp}(\mathcal{D}_{\text{no}}^{G,n,d})$, we transform it into an instance $\mathfrak{T}[I] \in \text{CSP}(\Gamma)$ as follows.

Step 1: variable sets. The variable set of $\mathfrak{T}[I]$ is partitioned into 4 parts:

- The set of *original variables* $V_{\text{orig}} = [n] \times [3]$.
- The set of *constant variables* $V_{\text{const}} = [n] \times [r_1] \times D$.
- The first set of *auxiliary variables* $V_{\text{aux}}^{(1)} = [d] \times [n] \times [r_1]$.
- The second set of *auxiliary variables* $V_{\text{aux}}^{(2)} = [\ell] \times [n] \times [r_1] \times [r_2]$.

Step 2: replacing constraints of I ? Initially, I is a d -regular $\text{CSP}(3\text{SUM}_G)$ -instance on $V_{\text{orig}} = [n] \times [3]$. For each original constraint $((v_1, v_2, v_3), S_b^G)$ in I , we would like to replace it with a copy of the $\text{CSP}(\Gamma \cup \text{CONST}_D)$ -instance $\mathcal{I}_b$, which simulates R_b . This copied instance cannot stand alone but must be incorporated as a part of $\mathfrak{T}[I]$. In Step 3 below, we describe how to assign its variables to V_{const} and $V_{\text{aux}}^{(1)}$.

Step 3: variable accommodation. Suppose $C = ((v_1, v_2, v_3), S_b^G)$ is the i -th constraint incident to v_1 in I . By assumption, $\mathcal{I}_b$ has at most r_1 secondary variables and r_1 constraints. Suppose $v_1 = (j, 1)$, where $j \in [n]$. We allocate:

$$\{j\} \times [r_1] \times D \subseteq V_{\text{const}} \quad \text{and} \quad \{i\} \times \{j\} \times [r_1] \subseteq V_{\text{aux}}^{(1)}$$

for these secondary variables.

Variables in $\mathcal{I}_b$ subject to constant constraints¹⁷ are identified with variables in $\{j\} \times [r_1] \times D \subseteq V_{\text{const}}$, where the choices in the third component correspond to which constants the variables are fixed to (the choices in the second component may be arbitrary). Other variables of $\mathcal{I}_b$ (neither primary nor constant-constrained) are represented by variables in $\{i\} \times \{j\} \times [r_1] \subseteq V_{\text{aux}}^{(1)}$.

Having specified the variable accommodations, we can now place a copy of $\mathcal{I}_b^{\sharp}$ in $\mathfrak{T}[I]$, in replacement of the constraint $C = ((v_1, v_2, v_3), S_b^G)$ in I . Note that here we put in $\mathfrak{T}[I]$ only the non-constant constraints of $\mathcal{I}_b$; the constant constraints will be “simulated” by the addition of an expander gadget in Steps 4 and 5. The set of constraints in this copy of $\mathcal{I}_b^{\sharp}$, as a subset of the constraint set of the eventual transformed instance $\mathfrak{T}[I]$, will be denoted by $\mathcal{Q}_C$.

Step 4: expander gadget. The constant-variable set $V_{\text{const}} = [n] \times [r_1] \times D$ can be viewed as a $|D|$ -partite vertex set, with each part containing $r_1 n$ vertices (partitioned according to the third component). We may therefore use the expander gadget $H \in \text{supp}(\mathcal{H}_{D, r_1 n, \ell})$ (prepared in Section 5.2.1), taking V_{const} as its vertex set. For each hyperedge $\mathbf{v} = (v_x)_{x \in D}$ of H , we would like to add an End_{Γ} -constraint $(\mathbf{v}, \text{End}_{\Gamma})$ to $\mathfrak{T}[I]$ (see Definition 2.17). In the language of Definition 5.2, this corresponds to placing a copy of $\mathcal{I}[\text{End}_{\Gamma}, H]$ on V_{const} . However, since the relation End_{Γ} may not belong to Γ itself, we must use the gadget instance $\mathcal{I}_{\text{End}}$ to simulate End_{Γ} ; this may introduce new variables, which must also be accommodated.

¹⁷These do not include the primary variables v_1, v_2, v_3 , since $\mathcal{I}_b$ simulates R on these variables and R does not fix any of them.

Step 5: variable accommodation (again). Fix some $z \in D$ as a “special element.” Suppose $\mathbf{v} = (v_x)_{x \in D}$ is the i -th hyperedge incident to $v_z = (j, t, z)$ in H . We create a copy of $\mathcal{I}_{End}$ simulating End_Γ on $\mathbf{v}$, and allocate

$$\{i\} \times \{j\} \times \{t\} \times [r_2] \subseteq V_{aux}^{(2)}$$

for the secondary variables in this copy of $\mathcal{I}_{End}$. The set of constraints in this copy of $\mathcal{I}_{End}$, as a subset of the constraint set of the eventual transformed instance $\mathfrak{T}[I]$, will be denoted by $\mathcal{Q}_{\mathbf{v}}$.

We repeat the process for every hyperedge $\mathbf{v}$ of H , thereby effectively placing a copy of $\mathcal{I}[End_\Gamma, H]$ on V_{const} .

Conclusion. We have now completed the construction of $\mathfrak{T}[I]$. The constraint set of the final instance $\mathfrak{T}[I]$ is the disjoint union of $\mathcal{Q}_C$ ’s and $\mathcal{Q}_{\mathbf{v}}$ ’s, where C ranges in the constraint set of the original $\text{CSP}(3\text{SUM}_G)$ -instance I and $\mathbf{v}$ ranges in the hyperedge set of the expander gadget H . There are dn constraints in I and $r_1 \ell n$ hyperedges in H ; each $\mathcal{Q}_C$ contains at most r_1 constraints and each $\mathcal{Q}_{\mathbf{v}}$ contains at most r_2 constraints. Consequently, the total number of constraints in $\mathfrak{T}[I]$ is at most

$$r_1 \cdot dn + r_2 \cdot r_1 \ell n = (d + \ell r_2) r_1 n,$$

and at least $dn + \ell r_1 n$. To obtain degree bounds¹⁸ on variables, we deal with the 4 parts of variable set separately:

- Every variable in V_{orig} is involved in at most d of the sets $\mathcal{Q}_C$ ’s, and so has degree at most dr_1 .
- Every variable $(j, t, x) \in V_{const}$ is involved in at most d of the sets $\mathcal{Q}_C$ ’s (namely, for those C that involve the variable $(j, 1) \in [n] \times [3]$) and at most ℓ of the sets $\mathcal{Q}_{\mathbf{v}}$ ’s. Therefore, the variable (j, t, x) has degree at most $dr_1 + \ell r_2$.
- Every variable in $V_{aux}^{(1)}$ is involved in at most 1 of the sets $\mathcal{Q}_C$ ’s, and so has degree at most r_1 .
- Every variable in $V_{aux}^{(2)}$ is involved in at most 1 of the sets $\mathcal{Q}_{\mathbf{v}}$ ’s, and so has degree at most r_2 .

Therefore, every variable in the instance $\mathfrak{T}[I]$ has degree at most $dr_1 + \ell r_2$.

5.3 Analysis of the Construction

Having completed the construction of the reduction map, we next prove its completeness and soundness.

Lemma 5.5 (Completeness). *For any $\text{CSP}(3\text{SUM}_G, [n] \times [3])$ -instance I in the support of $\mathcal{D}_{yes}^{G, n, d}$, the transformed instance $\mathfrak{T}[I] \in \text{CSP}(\Gamma)$ is satisfiable.*

Proof. By [Definition 4.3](#), there exists an assignment $\tau : [n] \times [3] \rightarrow G$ that satisfies I . We provide an assignment

$$\tilde{\tau} : V_{orig} \cup V_{const} \cup V_{aux}^{(1)} \cup V_{aux}^{(2)} \longrightarrow D$$

that satisfies $\mathfrak{T}[I]$, as follows.

- (1) For any original variable $v \in V_{orig} = [n] \times [3]$, we pick $\tilde{\tau}(v)$ to be an arbitrary element in the inverse image $\varphi^{-1}(\tau(v))$. Such an element exists because φ is surjective.

¹⁸The degree of a variable is the number of constraints it participates in.

- (2) For any constant variable $(j, t, x) \in V_{\text{const}} = [n] \times [r_1] \times D$, we assign $\tilde{\tau}((j, t, x)) = x$.
- (3) For each original constraint $C = ((v_1, v_2, v_3), S_b^G)$ in I , we know that there is an assignment to the instance $\mathcal{I}_b$ (prepared in [Section 5.2.1](#)) that is consistent with τ . We use this assignment to define $\tilde{\tau}(v)$ for every $v \in V_{\text{aux}}^{(1)}$ involved in $\mathcal{Q}_C$.

Any remaining variables in $V_{\text{aux}}^{(1)}$ not involved in any $\mathcal{Q}_C$ may be assigned arbitrarily by $\tilde{\tau}$.

- (4) Since the CSP(Γ)-instance $\mathcal{I}_{\text{End}}$ (prepared in [Section 5.2.1](#)) simulates the endomorphism relation of Γ , and since the identity map $\text{id} : D \rightarrow D$ is an endomorphism, there is an assignment to $\mathcal{I}_{\text{End}}$ that restricts to the identity assignment on the primary variables. For each hyperedge $\mathbf{v}$ of H , we use this assignment to define $\tilde{\tau}(v)$ for every $v \in V_{\text{aux}}^{(2)}$ involved in $\mathcal{Q}_{\mathbf{v}}$.

Any remaining variables in $V_{\text{aux}}^{(2)}$ not involved in any $\mathcal{Q}_{\mathbf{v}}$ may be assigned arbitrarily by $\tilde{\tau}$.

Items (1) and (3) above together ensure that $\tilde{\tau}$ satisfies all constraints in the sets $\mathcal{Q}_C$'s. Due to item (2), $\tilde{\tau}$ restricts to the identity assignment on any hyperedge $\mathbf{v}$ of H , and item (4) then ensures that $\tilde{\tau}$ also satisfies all constraints in the sets $\mathcal{Q}_{\mathbf{v}}$'s. Therefore, $\tilde{\tau}$ satisfies all constraints in $\mathfrak{T}[I]$. $\square$

Lemma 5.6 (Soundness). *For a random instance $\text{CSP}(3\text{SUM}_G, [n] \times [3])$ -instance $I \sim \mathcal{D}_{\text{no}}^{G, n, d}$, the transformed instance $\mathfrak{T}[I] \in \text{CSP}(\Gamma)$ has value at most $1 - (10(d + \ell r_2)r_1|D|)^{-1}$ with probability at least $1 - o(1)$. Here, $o(1)$ denotes a term tending to 0 as $n \rightarrow \infty$.*

Proof. As guaranteed in [Section 5.2.1](#), the instance I has value at most $2/3$ with probability $1 - o(1)$. We claim that whenever $\text{val}_I \leq 2/3$, any assignment must violate at least $n/(10|D|)$ constraints in $\mathfrak{T}[I]$. Since the total number of constraints in $\mathfrak{T}[I]$ is at most $(d + \ell r_2)r_1n$, this implies that

$$\text{val}_{\mathfrak{T}[I]} \leq 1 - \frac{n/(10|D|)}{(d + \ell r_2)r_1n} = 1 - \frac{1}{10(d + \ell r_2)r_1|D|},$$

as desired.

Assume on the contrary that $\text{val}_I \leq 2/3$ and there is an assignment

$$\tilde{\tau} : V_{\text{orig}} \cup V_{\text{const}} \cup V_{\text{aux}}^{(1)} \cup V_{\text{aux}}^{(2)} \longrightarrow D$$

violating less than $n/(10|D|)$ constraints of $\mathfrak{T}[I]$. Since H has $\ell r_1 n$ hyperedges $\mathbf{v}$, at most $1/(10r_1|D|)$ fraction of the sets $\mathcal{Q}_{\mathbf{v}}$'s contain constraints violated by $\tilde{\tau}$. Therefore, $\tilde{\tau}$ violates at most $1/(10r_1|D|)$ fraction of the constraints $(\mathbf{v}, \text{End}_{\Gamma})$ simulated by $\mathcal{Q}_{\mathbf{v}}$'s. In other words, we have

$$\text{val}_{\mathcal{I}(\text{End}_{\Gamma}, H)}(\tilde{\tau}|_{V_{\text{const}}}) \geq 1 - \frac{1}{10r_1|D|}.$$

Since H satisfies the property stated in [Lemma 5.4](#) (with respect to the sub-unique relation End_{Γ}), it follows from the conclusion of [Lemma 5.4](#) that there exists a relational endomorphism $\psi : D \rightarrow D$ of (D, Γ) such that

$$\mathbb{P}_{(j, t, x) \in V_{\text{const}}} [\tilde{\tau}((j, t, x)) \neq \psi(x)] \leq \frac{1}{5r_1|D|}. \quad (5.2)$$

As (D, Γ) is a core, ψ^{-1} is also a relational endomorphism. Thus the assignment

$$\psi^{-1} \circ \tilde{\tau} : V_{\text{orig}} \cup V_{\text{const}} \cup V_{\text{aux}}^{(1)} \cup V_{\text{aux}}^{(2)} \longrightarrow D$$

satisfies the same subset of constraints in $\mathfrak{T}[I]$ as $\tilde{\tau}$ does. In particular, it violates at most $n/(10|D|)$ of these constraints.

Now consider the assignment $\tau : [n] \times [3] \rightarrow \mathbf{G}$ defined as follows. For any $v \in V_{\text{orig}} = [n] \times [3]$, let

$$\tau(v) = \begin{cases} \varphi(\psi^{-1}(\tilde{\tau}(v))), & \text{if } \psi^{-1}(\tilde{\tau}(v)) \in D', \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to see that any original constraint $C = ((v_1, v_2, v_3), S_b^{\mathbf{G}})$ of I is violated by τ only if either of the following happens:

- (1) The assignment $\psi^{-1} \circ \tilde{\tau}$ violates one of the constraints in $\mathcal{Q}_C$.
- (2) For some constant variable $v = (j, t, x) \in V_{\text{const}}$ involved in $\mathcal{Q}_C$, it happens that $\psi^{-1}(\tilde{\tau}(v)) \neq x$.

The second scenario above happens for at most $|V_{\text{const}}|/(5r_1|D|) = n/5$ constant variables $v \in V_{\text{const}}$, due to (5.2). Since every constant variable in V_{const} is involved in at most d of the sets $\mathcal{Q}_C$'s, it follows that the second scenario happens for at most $dn/5$ constraints of I .

The first scenario happens for at most $n/(10|D|)$ constraints of I , since no two $\mathcal{Q}_C$'s share constraints in $\mathfrak{T}[I]$, and at most $n/(10|D|)$ constraints in $\mathfrak{T}[I]$ are violated by $\psi^{-1} \circ \tilde{\tau}$.

In total, at most

$$\frac{dn}{5} + \frac{n}{10|D|} < \frac{dn}{3}$$

constraints of I are violated by τ , contradicting the assumption $\text{val}_I \leq 2/3$. $\square$

The following lemma proves that the indistinguishability between $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$, as established in Lemma 4.7, is “preserved” by the reduction map $\mathfrak{T}[\cdot]$.

Lemma 5.7 (Indistinguishability). *Let $V = V_{\text{orig}} \cup V_{\text{const}} \cup V_{\text{aux}}^{(1)} \cup V_{\text{aux}}^{(2)}$. There exists a constant $\gamma > 0$, depending only on (D, Γ) but not on n , such that any algorithm $\mathcal{A}$ computing a function*

$$\mathcal{A} : \text{CSP}(\Gamma, V) \longrightarrow \{0, 1\}$$

using at most γn variable-queries satisfies

$$\left| \mathbb{P}_{I \sim \mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}} [\mathcal{A}(\mathfrak{T}[I]) = 1] - \mathbb{P}_{I \sim \mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}} [\mathcal{A}(\mathfrak{T}[I]) = 1] \right| \leq o(1),$$

where $o(1)$ denotes a term tending to 0 as $n \rightarrow \infty$.

Proof. Let $\mathcal{A}$ be such an algorithm that makes at most γn variable-queries. We claim that there exists an algorithm $\mathcal{A}'$ computing a function

$$\mathcal{A}' : \text{supp}(\mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}) \cup \text{supp}(\mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}) \longrightarrow \{0, 1\}$$

using at most γn variable-queries, such that

$$\mathcal{A}(\mathfrak{T}[I]) = \mathcal{A}'(I) \quad \text{for all } I \in \text{supp}(\mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}) \cup \text{supp}(\mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}). \quad (5.3)$$

In other words, given an instance I from $\mathcal{D}_{\text{yes}}$ or $\mathcal{D}_{\text{no}}$, the algorithm $\mathcal{A}'$ effectively simulates the execution of $\mathcal{A}$ on $\mathfrak{T}[I]$.

Consider the types of queries that $\mathcal{A}$ may make on input $\mathfrak{T}[I]$:

- A query to a variable in $V_{\text{aux}}^{(2)}$ reveals no information about I .
- A query to $(i, j, t) \in V_{\text{aux}}^{(1)}$ reveals no more information than a query to $(j, 1) \in V_{\text{orig}}$ in I .
- A query to $(j, t, x) \in V_{\text{const}}$ reveals no more information than a query to $(j, 1) \in V_{\text{orig}}$ in I .
- A query to $v \in V_{\text{orig}}$ reveals no more information than a query to the same variable v in I .

Thus every query to $\mathfrak{T}[I]$ performed by $\mathcal{A}$ can be replaced by a query to I by $\mathcal{A}'$, proving the claim.

Now let $\delta \in (0, 1)$ and $\mathcal{E}$ be the parameter and the event from [Lemma 4.7](#), respectively. Conditioned on the event that the pair of instances $(I_{\text{yes}}, I_{\text{no}})$ sampled from $\mathcal{D}_{\text{pair}}^{\mathbf{G}, n, d}$ falls in $\mathcal{E}$, the distributions of I_{yes} and I_{no} coincide when restricted to any δn variables, according to [Lemma 4.7](#). We set $\gamma = \delta/(3d + 1)$. Since any algorithm $\mathcal{A}'$ with γn queries to I can observe constraints involving at most $\gamma n(1 + 3d) = \delta n$ variables in V_{orig} , it follows that

$$\mathbb{P}_{(I_{\text{yes}}, I_{\text{no}}) \sim \mathcal{D}_{\text{pair}}^{\mathbf{G}, n, d}} [\mathcal{A}'(I_{\text{yes}}) = 1 \mid \mathcal{E}] = \mathbb{P}_{(I_{\text{yes}}, I_{\text{no}}) \sim \mathcal{D}_{\text{pair}}^{\mathbf{G}, n, d}} [\mathcal{A}'(I_{\text{no}}) = 1 \mid \mathcal{E}].$$

Since $\mathcal{D}_{\text{pair}}^{\mathbf{G}, n, d}(\mathcal{E}) \geq 1 - o(1)$, the fact that $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ are the marginals of $\mathcal{D}_{\text{pair}}^{\mathbf{G}, n, d}$ implies

$$\left| \mathbb{P}_{I \sim \mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}} [\mathcal{A}'(I) = 1] - \mathbb{P}_{I \sim \mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}} [\mathcal{A}'(I) = 1] \right| \leq o(1).$$

Combined with [\(5.3\)](#), this yields the desired conclusion. $\square$

5.4 Conclusion in the General Case

We are now ready to conclude the proof of [Theorem 2.4](#).

Proof of Theorem 2.4. We first consider the case where (D, Γ) is a core. In [Section 5.2](#), we obtained a constant $d \in \mathbb{N}$, an Abelian group $\mathbf{G}$, and a reduction map

$$\mathfrak{T} : \text{supp}(\mathcal{D}_{\text{yes}}^{\mathbf{G}, n, d}) \cup \text{supp}(\mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}) \longrightarrow \text{CSP}(\Gamma).$$

The map $\mathfrak{T}$ produces instances over the variable set

$$V_{\text{orig}} \cup V_{\text{const}} \cup V_{\text{aux}}^{(1)} \cup V_{\text{aux}}^{(2)},$$

whose total size is $(3 + |D|r_1 + dr_1 + \ell r_1 r_2)n$. As noted at the end of [Section 5.2.2](#), the instances produced by $\mathfrak{T}$ contain at least $(d + \ell r_1)n$ constraints, and each variable has degree at most $dr_1 + \ell r_2$. Define the constants

$$\beta = 3 + |D|r_1 + dr_1 + \ell r_1 r_2, \quad \alpha = \frac{d + \ell r_1}{\beta}, \quad d' = dr_1 + \ell r_2.$$

It follows that all instances produced by $\mathfrak{T}$ are valid in the $\text{BD}^*(d', \alpha, \beta n)$ model. By combining [Lemmas 5.5](#) to [5.7](#), there exists a constant $\gamma > 0$ such that, for sufficiently large n , the problem $\text{MaxCSP}(\Gamma)[1, 1 - \varepsilon]$ requires at least γn queries in the $\text{BD}^*(d', \alpha, \beta n)$ model, where

$$\varepsilon := \frac{1}{10(d + \ell r_2)r_1|D|}.$$

We now turn to the case where (D, Γ) is not a core. By [Proposition 2.8](#), (D, Γ) is homomorphically equivalent to a repetition-closed core (D', Γ') . We claim that $\text{MaxCSP}(\Gamma')[1, 1 - \varepsilon]$ is essentially equivalent to $\text{MaxCSP}(\Gamma)[1, 1 - \varepsilon]$.

Let $\varphi_1 : D \rightarrow D'$ and $\varphi_2 : D' \rightarrow D$ be the corresponding relational homomorphisms. Since Γ and Γ' are bijectively matched and corresponding relations have the same arity, any $\text{CSP}(\Gamma)$ instance $I = (V, \mathcal{C})$ can be viewed interchangeably as a $\text{CSP}(\Gamma')$ instance $I' = (V, \mathcal{C}')$, and vice versa. Furthermore, for any assignment $\tau : V \rightarrow D$ to I , we have

$$\text{val}_I(\tau) = \text{val}_{I'}(\varphi_1 \circ \tau),$$

by the definition of relational homomorphism. Conversely, for any assignment $\tau' : V \rightarrow D'$, we have

$$\text{val}_I(\varphi_2 \circ \tau') = \text{val}_{I'}(\tau').$$

We therefore have $\text{val}_I = \text{val}_{I'}$.

By the above discussion, the identification between $\text{CSP}(\Gamma)$ and $\text{CSP}(\Gamma')$ yields trivial reductions between $\text{MaxCSP}(\Gamma')[1, 1 - \varepsilon]$ and $\text{MaxCSP}(\Gamma)[1, 1 - \varepsilon]$. Since (D', Γ') also has unbounded width (by [Proposition A.3](#)), the hardness of $\text{MaxCSP}(\Gamma)[1, 1 - \varepsilon]$ follows from the established hardness of $\text{MaxCSP}(\Gamma')[1, 1 - \varepsilon]$. $\square$

Acknowledgements

The author would like to thank Dor Minzer, Ronitt Rubinfeld, Shuo Wang and Standa Živný for many stimulating discussions during the development of this work. Special thanks to Ronitt Rubinfeld, for her helpful suggestions in improving the presentation of the paper.

References

- [ACC25] Hugo Aaronson, Gaia Carenini, and Atreyi Chanda. Property testing in bounded degree hypergraphs. *arXiv preprint arXiv:2502.18382*, 2025.
- [AKK95] Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 284–293, 1995.
- [AS03] Noga Alon and Asaf Shapira. Testing satisfiability. *Journal of Algorithms*, 47(2):87–103, 2003.
- [Bar14] Libor Barto. The collapse of the bounded width hierarchy. *Journal of Logic and Computation*, 26(3):923–943, 2014.
- [BK14] Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *Journal of the ACM (JACM)*, 61(1):1–19, 2014.
- [BK16] Libor Barto and Marcin Kozik. Robustly solvable constraint satisfaction problems. *SIAM Journal on Computing*, 45(4):1646–1669, 2016.
- [BOT02] Andrej Bogdanov, Kenji Obata, and Luca Trevisan. A lower bound for testing 3-colorability in bounded-degree graphs. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 93–102. IEEE, 2002.

- [Bra22] Zarathustra Brady. Notes on csps and polymorphisms. *arXiv preprint arXiv:2210.07383*, 2022.
- [BSS10] Itai Benjamini, Oded Schramm, and Asaf Shapira. Every minor-closed property of sparse graphs is testable. *Advances in Mathematics*, 223:2200–2218, 2010.
- [CGS⁺22] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, Ameya Velinger, and Santhoshini Velusamy. Linear space streaming lower bounds for approximating csps. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 275–288, 2022.
- [CGSV24] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. Sketching approximability of all finite csps. *Journal of the ACM*, 71(2):1–74, 2024.
- [CMOS19] Artur Czumaj, Morteza Monemizadeh, Krzysztof Onak, and Christian Sohler. Planar graphs: Random walks and bipartiteness testing. *Random Structures & Algorithms*, 55(1):104–124, 2019.
- [DK13] Víctor Dalmau and Andrei Krokhin. Robust satisfiability for csps: Hardness and algorithmic results. *ACM Transactions on Computation Theory (TOCT)*, 5(4):1–25, 2013.
- [FMW25a] Yumou Fei, Dor Minzer, and Shuo Wang. A dichotomy theorem for multi-pass streaming csps. *arXiv preprint arXiv:2509.11399*, 2025.
- [FMW25b] Yumou Fei, Dor Minzer, and Shuo Wang. Multi-pass streaming lower bounds for approximating max-cut. *arXiv preprint arXiv:2503.23404*, 2025.
- [FV98] Tomás Feder and Moshe Y Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.
- [Gei68] David Geiger. Closed systems of functions and predicates. *Pacific journal of mathematics*, 27(1):95–100, 1968.
- [GGR98] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.
- [GR99] Oded Goldreich and Dana Ron. A sublinear bipartiteness tester for bounded degree graphs. *Combinatorica*, 19(3):335–373, 1999.
- [GR02] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002.
- [GR11] Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. In *Studies in Complexity and Cryptography: Miscellanea on the Interplay between Randomness and Computation*, pages 68–75. Springer, 2011.
- [GZ12] Venkatesan Guruswami and Yuan Zhou. Tight bounds on the approximability of almost-satisfiable horn sat and exact hitting set. *Theory of Computing*, 8(1):239–267, 2012.
- [Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.

- [HM88] David Hobby and Ralph McKenzie. *The structure of finite algebras*, volume 76. American Mathematical Society, 1988.
- [Kho02] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 767–775, 2002.
- [KK19] Michael Kapralov and Dmitry Krachun. An optimal space lower bound for approximating max-cut. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 277–288, 2019.
- [KKS14] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating max-cut. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1263–1282. SIAM, 2014.
- [LZ07] Benoit Larose and László Zádori. Bounded width problems and algebras. *Algebra universalis*, 56(3):439–466, 2007.
- [Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 245–254, 2008.
- [SSSV25] Raghuvansh R Saxena, Noah G Singer, Madhu Sudan, and Santhoshini Velusamy. Streaming algorithms via local algorithms for maximum directed cut. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3392–3408. SIAM, 2025.
- [Sze92] Ágnes Szendrei. A survey on strictly simple algebras and minimal varieties. In *Research and Exposition in Mathematics*, pages 209–239. Heldermann Verlag, Berlin, 1992.
- [Tre01] Luca Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 453–461, 2001.
- [Val09] Matthew A Valeriote. A subalgebra intersection property for congruence distributive varieties. *Canadian Journal of Mathematics*, 61(2):451–464, 2009.
- [vH14] Ramon van Handel. Probability in high dimension (lecture notes), 2014.
- [Wor99] Nicholas Charles Wormald. Models of random regular graphs. In *Surveys in Combinatorics, 1999*, pages 239–298. 1999.
- [YI10] Yuichi Yoshida and Hiro Ito. Query-number preserving reductions and linear lower bounds for testing. *IEICE transactions on information and systems*, 93(2):233–240, 2010.
- [YI12] Yuichi Yoshida and Hiro Ito. Property testing on k-vertex-connectivity of graphs. *Algorithmica*, 62(3):701–712, 2012.
- [Yos11] Yuichi Yoshida. Optimal constant-time approximation algorithms and (unconditional) inapproximability results for every bounded-degree csp. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 665–674, 2011.

A Definition of Bounded Width

In this section, we provide the definition and basic properties of the “bounded width” notion for CSPs. The class of bounded-width CSP templates is defined by the correctness of an algorithm, which we call the width- (k, ℓ) algorithm, in deciding satisfiability of instances.

Definition A.1 ([FV98]). Let k, ℓ be integers such that $\ell \geq k \geq 1$. A CSP template (D, Γ) is said to have width (k, ℓ) if for all instances $I \in \text{CSP}(\Gamma)$, [Algorithm 1](#) correctly decides whether I is satisfiable. A template (D, Γ) is said to have bounded width if it has width (k, ℓ) for some integers k, ℓ ; otherwise, we say (D, Γ) has unbounded width.

Algorithm 1: Width- (k, ℓ) Algorithm

Input : a $\text{CSP}(\Gamma)$ instance $I = (V, (C_1, \dots, C_m))$
Output: satisfiable or unsatisfiable

```

1 for each subset  $U \subseteq V$  of size  $\ell$  do
2   Initialize a set of partial assignments  $S_U \leftarrow D^U$ 
3   for each pair  $(\tau, i)$  of partial assignment  $\tau \in D^U$  and constraint index  $i \in [m]$  do
4     if  $U$  contains the scope of  $C_i$  and  $\tau$  does not satisfy  $C_i$  then
5        $S_U \leftarrow S_U \setminus \{\tau\}$  // remove inconsistencies with original constraints

                                     // finish initialization; start main procedure
6 repeat
7   for each pair of subsets  $U_1, U_2 \subseteq V$  such that  $|U_1| = |U_2| = \ell$  do
8     for each (if any) subset  $W \subseteq U_1 \cap U_2$  such that  $|W| = k$  do
9       for each  $\tau \in S_{U_1}$  do
10        if  $\tau|_W \neq \sigma|_W$  for all  $\sigma \in S_{U_2}$  then
11           $S_{U_1} \leftarrow S_{U_1} \setminus \{\tau\}$  // remove inconsistencies in any  $k$  variables
12 until the sets  $S_U$  stop changing
13 Output unsatisfiable if  $S_U = \emptyset$  for some  $U$ ; otherwise output satisfiable

```

Note that [Algorithm 1](#) is always correct if it outputs **unsatisfiable**, so the only mistake is when the input instance is unsatisfiable but it outputs **satisfiable**. Also, since the running time of [Algorithm 1](#) is clearly polynomial in $|V|$ and m , we immediately have the following property:

Proposition A.2. For a CSP template (D, Γ) , we let Γ -SAT be the problem of deciding satisfiability of instances from $\text{CSP}(\Gamma)$. If (D, Γ) has bounded width, then Γ -SAT $\in \mathbf{P}$.

The following basic property of bounded-width templates is used in [Section 5.4](#).

Proposition A.3. Let (D_1, Γ_1) and (D_2, Γ_2) be two relational structures that are homomorphically equivalent to each other. Then (D_1, Γ_1) has bounded width if and only if (D_2, Γ_2) has bounded width.

Proof. Suppose $\varphi_1: D_1 \rightarrow D_2$ and $\varphi_2: D_2 \rightarrow D_1$ are homomorphisms. Assume that (D_2, Γ_2) has width (k, ℓ) . To show that (D_1, Γ_1) also has width (k, ℓ) , it suffices to prove that for any instance $I \in \text{CSP}(\Gamma_1)$, if [Algorithm 1](#) outputs **satisfiable** on I , then I is indeed satisfiable.

For local assignments we introduce the following notation:

- (1) For each subset $U \subseteq V$ and each collection of local assignments $S \subseteq D_1^U$, set

$$\varphi_1^U(S) := \{\varphi_1 \circ \tau \mid \tau \in S\} \subseteq D_2^U.$$

- (2) For any $W, U \subseteq V$ with $W \subseteq U$, and each set of local assignments $S \subseteq D_1^U$, define the restriction

$$\text{Res}_{U,W}(S) := \{\tau|_W \mid \tau \in S\}.$$

By slight abuse of notation we use the same symbol $\text{Res}_{U,W}(S)$ when $S \subseteq D_2^U$.

Let $\{S_U\}_{U \subseteq V, |U|=\ell}$ denote the family of local assignment sets produced by [Algorithm 1](#) at the end of its execution on I . Since we have assumed that [Algorithm 1](#) outputs **satisfiable** on I , it follows that $S_U \neq \emptyset$ for every U . By [Line 11](#) of [Algorithm 1](#), we also have

$$\text{Res}_{U_1,W}(S_{U_1}) = \text{Res}_{U_2,W}(S_{U_2})$$

for any k -element set $W \subseteq V$ and any ℓ -element sets $U_1, U_2 \subseteq V$ containing W . Furthermore, by [Line 5](#), for any ℓ -element set $U \subseteq V$, every assignment in S_U satisfies all constraints of I whose scopes are contained in U .

Construct an instance $I' \in \text{CSP}(\Gamma_2)$ by replacing each constraint of I with the corresponding constraint from Γ_2 . Since φ_1 is a relational homomorphism, [Definition 2.5](#) implies that for any ℓ -element set $U \subseteq V$, each assignment $\tau' \in \varphi_1^U(S_U)$ satisfies all constraints of I' whose scopes are contained in U . Moreover, for any k -element set $W \subseteq V$ and ℓ -element sets $U_1, U_2 \subseteq V$ containing W , we have

$$\text{Res}_{U_1,W}(\varphi_1^{U_1}(S_{U_1})) = \varphi_1^W(\text{Res}_{U_1,W}(S_{U_1})) = \varphi_1^W(\text{Res}_{U_2,W}(S_{U_2})) = \text{Res}_{U_2,W}(\varphi_1^{U_2}(S_{U_2})).$$

Thus the (nonempty) family of local assignment sets $\{\varphi_1^U(S_U)\}_{U \subseteq V, |U|=\ell}$ persists to the end of [Algorithm 1](#) when run on I' . Since (D_2, Γ_2) has width (k, ℓ) by assumption, I' is satisfiable. Finally, for any global assignment $\tau': V \rightarrow D_2$ satisfying I' , the composed assignment $\varphi_2 \circ \tau': V \rightarrow D_1$ satisfies I because φ_2 is also a relational homomorphism. Hence I is satisfiable, completing the proof. $\square$

The following simple observation is used in the proof of [Lemma 3.4](#).

Proposition A.4. Let (D, Γ_1) and (D, Γ_2) be two relational structures on a same domain. If (D, Γ_1) has unbounded width and $\Gamma_1 \subseteq \Gamma_2$, then (D, Γ_2) also has unbounded width.

Proof. This is simply because any $\text{CSP}(\Gamma_1)$ instance is also a $\text{CSP}(\Gamma_2)$ instance. If [Algorithm 1](#) makes error on some instance $I \in \text{CSP}(\Gamma_1)$, then it also makes error on the same instance $I \in \text{CSP}(\Gamma_2)$. $\square$

Remark A.5. A decidable characterization of bounded-width CSP templates was given by Barto [[Bar14](#), Corollary 8.5]. Using Barto's criterion, it is easy to verify that for any $k, \ell \geq 2$ with $(k, \ell) \neq (2, 2)$, the CSP template expressing k -colorability of ℓ -uniform hypergraphs does not have bounded width.

B Allowing Variable Repetition

Proof of Theorem 1.8 assuming Theorem 2.4. Given any relational structure (D, Γ) , we will define a reduction map $\mathfrak{R} : \text{CSP}(\bar{\Gamma}) \rightarrow \text{CSP}(\Gamma)$. Let k be the maximum arity of any relation in Γ . Given an instance $I = (V, \mathcal{C}) \in \text{CSP}(\bar{\Gamma})$, we construct $\mathfrak{R}[I]$ as follows.

- (1) Replace every variable $v \in V$ by $m := k|D|$ copies $v^{(1)}, \dots, v^{(m)}$. The new variable set is denoted by $V \times [m]$.
- (2) Each constraint $C \in \mathcal{C}$ can be viewed as a requirement

$$R(\tau(v_1), \dots, \tau(v_r)) = 1$$

for assignments $\tau : V \rightarrow D$, where $R \in \Gamma$ and $v_1, \dots, v_r$ are *not necessarily distinct* variables in V . In place of this constraint, we add to $\mathfrak{R}[I]$ every constraint of the form

$$\left((v_1^{(i_1)}, \dots, v_r^{(i_r)}), R \right)$$

such that the chosen variables $v_1^{(i_1)}, \dots, v_r^{(i_r)}$ are pairwise distinct.

The reduction map has the following desired properties.

- (1) Note that every constraint of I corresponds to at least 1 and at most m^k constraints of $\mathfrak{R}[I]$. If I has maximum degree at most d , then $\mathfrak{R}[I]$ has maximum degree at most $m^k d$.
- (2) It is clear that if I is satisfiable then $\mathfrak{R}[I]$ is also satisfiable.
- (3) If $\text{val}_I \leq 1 - \varepsilon$, then we claim that $\text{val}_{\mathfrak{R}[I]} \leq 1 - m^{-k} \varepsilon$. Given any assignment $\tilde{\tau} : V \times [m] \rightarrow D$ and any $v \in V$, by the pigeonhole principle some element of D appears in $\tau(v^{(1)}), \dots, \tau(v^{(m)})$ at least k times. We define $\tau(v)$ to be any such element. For any constraint $C \in \mathcal{C}$ that this assignment $\tau : V \rightarrow D$ violates, at least one of the (at most m^k) constraints in $\mathfrak{R}[I]$ corresponding to C is violated by $\tilde{\tau}$. Therefore, the assumption that $\text{val}_I(\tau) \leq 1 - \varepsilon$ implies $\text{val}_{\mathfrak{R}[I]}(\tilde{\tau}) \leq 1 - m^{-k} \varepsilon$.
- (4) Finally, it is easy to see that every query to a variable $v^{(i)}$ in $\mathfrak{R}[I]$ reveals no more information than a query to v in I .

Now suppose Theorem 2.4 holds. Then for any unbounded-width CSP template (D, Γ) , since $(D, \bar{\Gamma})$ is repetition-closed and also has unbounded width (due to Proposition A.4), there exists $\varepsilon > 0$ such that $\text{MaxCSP}(\bar{\Gamma})[1, 1 - \varepsilon]$ requires $\Omega(n)$ query in the $\text{BD}^*(d, \alpha, n)$ model. Due to the reduction above, this implies that $\text{MaxCSP}(\Gamma)[1, 1 - m^{-k} \varepsilon]$ requires $\Omega(n)$ queries in the $\text{BD}^*(m^k d, \alpha, mn)$ model, where k is the largest arity of any relation in Γ and $m := k|D|$. $\square$

C Galois Duality Proof

Proof of Theorem 2.16. The proof proceeds in the following steps.

Step 1: the variable pool. Recall from [Definition 2.1](#) that to show $R \in \langle \Gamma \rangle^*$ we must build a $\text{CSP}(\Gamma)$ instance that “simulates” R . Such an instance contains k primary variables $v_1, \dots, v_k$ on which R is simulated, together with a number of secondary variables. Here, however, we first introduce a single *pool* of variables containing both the primary and secondary variables we need. Only afterwards will we specify which are the primary variables.

Let $R^{-1}(1) = \{x = (x_1, \dots, x_k) \in D^k \mid R(x) = 1\}$ and put $m = |R^{-1}(1)|$. We create $|D|^m$ variables u_y , indexed by all vectors $y \in D^m$. Initially, the instance I on variables $(u_y)_{y \in D^m}$ has no constraints; the next step explains how constraints are added.

Step 2: encoding “polymorphism”. The key idea is that the condition of being a polymorphism of Γ can be written syntactically as a CSP instance. An m -ary operation $f: D^m \rightarrow D$ corresponds to an assignment

$$\tau : \{u_y \mid y \in D^m\} \rightarrow D,$$

interpreting $\tau(u_y) = f(y)$. Recall that for f to qualify as a polymorphism of (D, Γ) , it needs to pass all tests given by a relation $R' \in \Gamma$ and a matrix of elements $(x_{i,j}) \in D^{\ell \times m}$, where ℓ is the arity of R' , such that each column of the matrix satisfies R' . In the test, one checks the output of f on each *row* of the matrix and verifies if the combined outputs as an element of D^ℓ satisfies R' . We now describe a procedure producing a CSP instance that enforces these tests.

- For each test $(R', (x_{i,j})) \in \Gamma \times D^{\ell \times m}$:
 1. For each row index $i \in [\ell]$, define $y^{(i)} \in D^m$ by $y_j^{(i)} = x_{i,j}$ for all $j \in [m]$.
 2. Add to I the constraint

$$((u_{y^{(1)}}, \dots, u_{y^{(\ell)}}), R').$$

By construction, an assignment τ satisfies I iff the corresponding operation f is a polymorphism of Γ .

Step 3: selecting the primary variables. Let $w^{(1)}, \dots, w^{(m)}$ be an enumeration of elements of $R^{-1}(1)$. For each $i \in [k]$ define $z^{(i)} \in D^m$ by $z_j^{(i)} = w_i^{(j)}$ for $j \in [m]$. The condition that R is irredundant now easily translates to (desired) property that the vectors $z^{(i)}$ are distinct. We designate as the primary variables

$$(u_{z^{(1)}}, \dots, u_{z^{(k)}}).$$

We have now completed the construction of I .

Step 4: correctness. For any $x = (x_1, \dots, x_k) \in R^{-1}(1)$ we have $x = w^{(j)}$ for some $j \in [m]$. The assignment $\tau(u_y) = y_j$ (corresponding to the projection polymorphism $\pi_j: D^m \rightarrow D$) then satisfies I and yields $\tau(u_{z^{(i)}}) = x_i$ for all $i \in [k]$.

Conversely, if τ satisfies I , the induced operation $f: D^m \rightarrow D$ is a polymorphism of (D, Γ) and hence preserves R . Since each column of the matrix $(w_i^{(j)})_{i \in [k], j \in [m]} \in D^{k \times m}$ satisfies R , the tuple of row outputs

$$(f(z^{(1)}), \dots, f(z^{(k)})) = (\tau(u_{z^{(1)}}), \dots, \tau(u_{z^{(k)}})) \in D^k$$

also satisfies R . Thus the instance $I \in \text{CSP}(\Gamma)$ correctly simulates R on the designated primary variables in the sense of [Definition 2.1](#), which proves $R \in \langle \Gamma \rangle^*$. $\square$

D Random Regular Hypergraphs

In this section, we outline the proofs of [Lemmas 4.5](#), [4.7](#) and [5.4](#), all of which revolve around random regular hypergraphs. The first two lemmas, [Lemmas 4.5](#) and [4.7](#), are only minor adaptations of results from [\[BOT02\]](#), and our proofs closely follow those in that work. In contrast, [Lemma 5.4](#) is new to this paper, though its proof uses arguments that are largely analogous to those used in the proof of [Lemma 4.5](#).

D.1 Random Regular Hypergraphs are Locally Sparse

We first sketch a proof of [Lemma 4.7](#).

Recall from [Definition 4.2](#) that $\mathcal{M}_{[3],n}$ denotes the set of all 3-partite perfect matchings on the vertex set $[n] \times [3]$. For any matching $M \in \mathcal{M}_{[3],n}$ and any vertex subset $U \subseteq [n] \times [3]$, we let $M[U]$ denote the set of hyperedges $(v_1, v_2, v_3) \in M$ such that $v_1, v_2, v_3 \in U$. The following lemma is a slightly modified version of [\[BOT02, Lemma 6\]](#).

Lemma D.1 ([\[BOT02\]](#)). *For any fixed positive integer d , there exists a constant $\delta \in (0, 1)$ such that if $M^{(1)}, \dots, M^{(d)}$ are perfect matchings sampled independent and uniformly from $\mathcal{M}_{[3],n}$, then the condition*

$$\sum_{i=1}^d |M^{(i)}[U]| < \frac{2}{3}|U|, \quad \text{for all } U \subseteq [n] \times [3] \text{ with } |U| \leq \delta n \quad (\text{D.1})$$

is satisfied with probability $1 - o(1)$, where $o(1)$ is a term tending to 0 as $n \rightarrow \infty$.

Proof. For a fixed $U \subseteq [n] \times [3]$ and any $i \in [3]$, let $U_i = \{(j, i) \in U \mid j \in [n]\}$ be the set of vertices in U that belong to the i -th part in the 3-way partition of $[n] \times [3]$. Let W be the set of pairs $(j, i) \in U_1 \times [d]$ such that some hyperedge (v_1, v_2, v_3) in $M^{(i)}$ contains the vertex $v_1 = (j, 1)$ and satisfies $v_1, v_2, v_3 \in U$. We therefore have

$$|W| = \sum_{i=1}^d |M^{(i)}[U]|.$$

For any given subset $W_0 \subseteq U_1 \times [d]$, it is not hard to see that

$$\mathbb{P}_{M^{(1)}, \dots, M^{(d)} \in \mathcal{M}_{[3],n}} [W \supseteq W_0] \leq \left(\frac{|U_2| \times |U_3|}{n^2} \right)^{|W_0|}.$$

Therefore we have

$$\begin{aligned} \mathbb{P}_{M^{(1)}, \dots, M^{(d)} \in \mathcal{M}_{[3],n}} \left[|W| \geq \frac{2}{3}|U| \right] &\leq \sum_{\substack{W_0 \subseteq U_1 \times [d] \\ |W_0| = \lceil 2|U|/3 \rceil}} \mathbb{P}_{M^{(1)}, \dots, M^{(d)} \in \mathcal{M}_{[3],n}} [W \supseteq W_0] \\ &\leq \binom{d|U_1|}{\lceil 2|U|/3 \rceil} \left(\frac{|U_2| \times |U_3|}{n^2} \right)^{\lceil 2|U|/3 \rceil} \leq \left(\frac{5d|U|^2}{n^2} \right)^{\lceil 2|U|/3 \rceil}. \end{aligned}$$

Pick $\delta = 1/(50d^2)$. By taking a union bound over all $U \subseteq [n] \times [3]$ with $|U| \leq \delta n$, we have

$$\mathbb{P} [\text{condition (D.1) fails}] \leq \sum_{t=1}^{\lfloor \delta n \rfloor} \binom{3n}{t} \left(\frac{5dt^2}{n^2} \right)^{2t/3} \leq \sum_{t=1}^{\lfloor \delta n \rfloor} \left(\frac{t}{n} \right)^{t/3} \leq o(1). \quad \square$$

As stated in [BOT02], the conclusion of [Lemma D.1](#) still holds if the constant $2/3$ in [\(D.1\)](#) is replaced by any constant greater than $1/2$. The reason for choosing $2/3$ in [\(D.1\)](#) is that it is the largest constant for which the following observation remains true.

Proposition D.2. Suppose $M^{(1)}, \dots, M^{(d)}$ are perfect matchings in $\mathcal{M}_{[3],n}$ that satisfy [\(D.1\)](#). Then, for any subset $U \subseteq [n] \times [3]$ with $|U| \leq \delta n$, the following hold:

- (1) The edge sets $M^{(i)}$, for $i \in [d]$, are pairwise disjoint.
- (2) There exists a total order on the union $\bigcup_{i=1}^d M^{(i)}$ such that every hyperedge in this union is incident to at least one vertex that does not appear in any hyperedge preceding it in the order.

Proof. The first item holds because no hyperedge can appear in more than one of the $M^{(i)}$'s; otherwise, there would exist a set U of three vertices such that $\sum_{i=1}^d |M^{(i)}[U]| \geq 2$, violating condition [\(D.1\)](#).

We now prove item (2). Let $U_0 \subseteq U$ denote the set of vertices incident to at least one hyperedge in the union $E := \bigcup_{i=1}^d M^{(i)}$. Applying condition [\(D.1\)](#) to U_0 implies that a uniformly random vertex from U_0 is, in expectation, incident to fewer than two hyperedges in E . Hence, there exists at least one vertex in U_0 that is incident to exactly one hyperedge in E .

We remove this hyperedge from E and update U_0 to be the set of vertices incident to at least one hyperedge in the remaining set E . Repeating this process iteratively, we eventually remove all hyperedges from E . This procedure defines a total order on the edge set $\bigcup_{i=1}^d M^{(i)}$, where edges removed later are considered smaller in the order. The desired conclusion then follows directly from the construction. $\square$

We now use [Lemma D.1](#) and [Proposition D.2](#) to prove [Lemma 4.7](#).

Proof Sketch of Lemma 4.7. Recall from [Definition 4.3](#) that for any sample $(I_{\text{yes}}, I_{\text{no}}) \sim \mathcal{D}_{\text{pair}}^{\mathbf{G}, n, d}$, the same collection of d perfect matchings $M^{(1)}, \dots, M^{(d)} \in \mathcal{M}_{[3],n}$ underlies both I_{yes} and I_{no} . Applying [Lemma D.1](#), we obtain a constant $\delta \in (0, 1)$, and let $\mathcal{E}$ denote the event that the condition [\(D.1\)](#) holds for $M^{(1)}, \dots, M^{(d)}$.

We now condition on a choice of $M^{(1)}, \dots, M^{(d)}$ satisfying [\(D.1\)](#). Recall that for each hyperedge $\mathbf{v} = (v_1, v_2, v_3) \in \bigcup_{i=1}^d M^{(i)}$, the instance I_{yes} contains a constraint $(\mathbf{v}, S_b^{\mathbf{G}})$ and I_{no} contains a constraint $(\mathbf{v}, S_{b'}^{\mathbf{G}})$, for possibly distinct “right-hand side” coefficients $b, b' \in \mathbf{G}$. The right-hand sides of the equations in I_{no} are chosen independently, whereas those in I_{yes} are correlated through an underlying random assignment $\tau : [n] \times [3] \rightarrow \mathbf{G}$.

However, it follows from [Proposition D.2](#) that for any variable set $U \subseteq [n] \times [3]$ of size at most δn , the equations placed by I_{yes} within U have right-hand sides that remain jointly independent. Consequently, conditioned on $M^{(1)}, \dots, M^{(d)}$, the restricted random instances $\text{Sub}_U[I_{\text{yes}}]$ and $\text{Sub}_U[I_{\text{no}}]$ are identically distributed, as desired. $\square$

D.2 Concentration of Value for Random Instances

The proofs of [Lemmas 4.5](#) and [5.4](#) both hinge on the following concentration inequality, known as the Azuma-Hoeffding inequality.

Proposition D.3 ([vH14, Corollary 3.9]). Let $\{\mathcal{F}_t\}_{0 \leq t \leq n}$ be a filtration of σ -algebras on a probability space, and let X be a real-valued random variable. If there exists a constant $c > 0$ such that $|\mathbb{E}[X|\mathcal{F}_t] - \mathbb{E}[X|\mathcal{F}_{t-1}]| \leq c$ almost surely for all $t \in [n]$, then

$$\mathbb{P}\left[\mathbb{E}[X|\mathcal{F}_n] - \mathbb{E}[X|\mathcal{F}_0] \geq \Delta\right] \leq \exp\left(-\frac{\Delta^2}{2nc^2}\right).$$

Proof Sketch of Lemma 4.5. Recall from Definition 4.3 that, in process of generating the instance $I \sim \mathcal{D}_{\text{no}}^{\mathbf{G}, n, d}$, the total of dn constraints are placed in d rounds. In each round, for each $j \in [n]$, exactly one constraint involving $(j, 1)$ is added. This naturally divides the stochastic process of generating I into dn time steps, where exactly one constraint is added to I in each step. Let $\mathcal{F}_t$ be the σ -algebra corresponding to the information after t time steps are completed, for each $t \in \{0, 1, \dots, dn\}$. For each fixed assignment $\tau : [n] \times [3] \rightarrow \mathbf{G}$, we define the random variable $X_\tau := \text{val}_I(\tau)$ and apply Proposition D.3 to X_τ and the filtration $(\mathcal{F}_t)_{0 \leq t \leq dn}$. Using a “switching” argument (e.g., see [Wor99, Lemma 2.19]), it is not hard to see that

$$|\mathbb{E}[X_\tau | \mathcal{F}_t] - \mathbb{E}[X_\tau | \mathcal{F}_{t-1}]| \leq \frac{2}{dn},$$

for any $t \in [dn]$. Furthermore, we have $\mathbb{E}[X_\tau | \mathcal{F}_{dn}] = X_\tau = \text{val}_I(\tau)$ and $\mathbb{E}[X | \mathcal{F}_0] = \mathbb{E}[X_\tau] = |\mathbf{G}|^{-1}$. Therefore, the conclusion of Proposition D.3 implies

$$\mathbb{P}_I \left[\text{val}_I(\tau) \geq \frac{1}{|\mathbf{G}|} + \varepsilon \right] \leq \exp \left(-\frac{\varepsilon^2 dn}{8} \right).$$

A union bound over all τ then implies

$$\mathbb{P}_I \left[\text{val}_I \geq \frac{1}{|\mathbf{G}|} + \varepsilon \right] \leq |\mathbf{G}|^{3n} \exp \left(-\frac{\varepsilon^2 dn}{8} \right) \leq o(1),$$

if we set $d \geq 100\varepsilon^{-2} \ln |\mathbf{G}|$. □

We now turn to prove Lemma 5.4. The idea of the proof is similar to the above proof of Lemma 4.5, but more preparation is needed before starting the proof.

Definition D.4. For any nonempty finite set D , we define a relation $\text{Perm}_D : D^D \rightarrow \{0, 1\}$ as follows. For any tuple $y = (y_x)_{x \in D} \in D^D$, we let $\text{Perm}_D(y) = 1$ if and only if the map $x \mapsto y_x$ is a permutation of D .

Note that Perm_D is the “maximal” sub-unique relation on D , as $R(y) \leq \text{Perm}_D(y)$ for any sub-unique relation $y : D^D \rightarrow \{0, 1\}$.

Definition D.5. For any assignment $\tau : [n] \times D \rightarrow D$, we define its *Perm-value* to be

$$\text{Perm-val}(\tau) := \mathbb{E}_{\mathbf{v}=(v_x)_{x \in D}} \left[\text{Perm}_D((\tau(v_x))_{x \in D}) \right],$$

where $\mathbf{v} = (v_x)_{x \in D}$ is a random hyperedge chosen by sampling each v_x from $[n] \times \{x\}$ independently and uniformly at random.

Lemma D.6. For any assignment $\tau : [n] \times D \rightarrow D$ such that $\text{Perm-val}(\tau) \geq 1 - \frac{1}{5|D|}$, there exists a permutation $\pi : D \rightarrow D$ such that

$$\mathbb{P}_{i \in [n]} [\tau((i, x)) = \pi(x)] \geq \text{Perm-val}(\tau) \tag{D.2}$$

holds for all $x \in D$.

Proof. We first show the existence of a map $\pi : D \rightarrow D$ satisfying (D.2) for all $x \in D$, and subsequently prove that the map π thus obtained is a permutation.

For any $x, z \in D$, let $F(x, z)$ denote the fraction of indices $i \in [n]$ such that $\tau((i, x)) = z$. Fix an element $t \in D$ and consider a hyperedge $\mathbf{v} = (v_x)_{x \in D}$. We decompose it into v_t and $\mathbf{v}_{\setminus \{t\}} = (v_x)_{x \in D \setminus \{t\}}$. For the map $x \mapsto \tau(v_x)$ to be a permutation of D , two conditions must hold:

- (1) The values $\tau(v_x)$ for $x \in D \setminus \{t\}$ are pairwise distinct.
- (2) The value $\tau(v_t)$ is the unique remaining element of D .

With τ fixed and $\mathbf{v}$ chosen uniformly at random, conditioning on the requirement (1), the probability that (2) also holds is at most $\max_{z \in D} F(t, z)$. Consequently, the overall probability that $(\tau(v_x))_{x \in D}$ forms a permutation (i.e. $\text{Perm-val}(\tau)$) is at most $\max_{z \in D} F(t, z)$. Define $\pi(t)$ as the element of D attaining this maximum, i.e.

$$F(t, \pi(t)) = \max_{z \in D} F(t, z).$$

By the discussion above, for any $t \in D$ we have

$$\mathbb{P}_{i \in [n]} [\tau((i, t)) = \pi(t)] = \max_{z \in D} F(t, z) \geq \text{Perm-val}(\tau).$$

It remains to show that π is a permutation. For a uniformly random hyperedge $\mathbf{v}$, we have

$$\mathbb{P}_{\mathbf{v}=(v_x)_{x \in D}} [\tau(v_x) = \pi(x), \quad \forall x \in D] \geq \prod_{x \in D} F(x, \pi(x)) \geq \text{Perm-val}(\tau)^{|D|} \geq \left(1 - \frac{1}{5|D|}\right)^{|D|}.$$

If π were not a permutation, then

$$\text{Perm-val}(\tau) = \mathbb{E}_{\mathbf{v}=(v_x)_{x \in D}} [\text{Perm}_D((\tau(v_x))_{x \in D})] \leq 1 - \left(1 - \frac{1}{5|D|}\right)^{|D|} < 1 - \frac{1}{10|D|},$$

contradicting the assumption. Hence, π must be a permutation. $\square$

We are now ready to prove [Lemma 5.4](#).

Proof of Lemma 5.4. We call an assignment $\tau : [n] \times D \rightarrow D$ “bad” (with respect to H) if $\text{val}_{\mathcal{I}[R, H]}(\tau) \geq 1 - \varepsilon$ but

$$\mathbb{P}_{(i, x) \in [n] \times D} [\tau((i, x)) \neq y_x] > 2\varepsilon \tag{D.3}$$

for any tuple $y \in D^D$ with $R(y) = 1$. For any fixed assignment $\tau : [n] \times D \rightarrow D$, we will analyze the probability that it is bad with respect to a random hypergraph $H \sim \mathcal{H}_{D, n, \ell}$.

Case 1: $\text{Perm-val}(\tau) \geq 1 - 2\varepsilon$. Since $2\varepsilon \leq \frac{1}{5|D|}$, we may apply [Lemma D.6](#) and obtain a permutation $\pi : D \rightarrow D$ such that

$$\mathbb{P}_{i \in [n]} [\tau((i, x)) = \pi(x)] \geq 1 - 2\varepsilon$$

holds for any $x \in D$. If $R((\pi(x))_{x \in D}) = 1$, then this contradicts the assumption that [\(D.3\)](#) holds for all $y \in D^D$ with $R(y) = 1$. Therefore, we must have $R((\pi(x))_{x \in D}) = 0$.

Now if $\mathbf{v} = (v_x)_{x \in D}$ is a random hyperedge chosen by sampling each v_x from $[n] \times \{x\}$ independently and uniformly at random, then

$$\mathbb{E}_{\mathbf{v}=(v_x)_{x \in D}} [R((\tau(v_x))_{x \in D})] \leq 1 - \mathbb{P}_{\mathbf{v}=(v_x)_{x \in D}} [\tau(v_x) = \pi(x), \quad \forall x \in D] \leq 1 - (1 - 2\varepsilon)^{|D|} \leq 1 - 2\varepsilon.$$

Using the martingale argument in the proof of [Lemma 4.5](#), it follows that

$$\mathbb{P}_H [\text{val}_{\mathcal{I}[R, H]}(\tau) \geq 1 - \varepsilon] \leq \exp\left(-\frac{\varepsilon^2 \ell n}{8}\right).$$

Therefore, the probability that τ is bad with respect to a random H is at most $\exp(-\varepsilon^2 \ell n / 8)$.

Case 2: $\text{Perm-val}(\tau) < 1 - 2\varepsilon$. Since $R(y) \leq \text{Perm}_D(y)$ for all $y \in D^D$, by [Definition D.5](#) we have

$$\mathbb{P}_{\mathbf{v}=(v_x)_{x \in D}} [R(\tau(v_x)_{x \in D})] \leq \mathbb{P}_{\mathbf{v}=(v_x)_{x \in D}} [\text{Perm}_D(\tau(v_x)_{x \in D})] = \text{Perm-val}(\tau) < 1 - 2\varepsilon.$$

As in case 1, it follows again that τ is bad with respect to H with probability at most $\exp(-\varepsilon^2 \ell n / 8)$.

In conclusion, any fixed assignment $\tau : [n] \times D \rightarrow D$ is bad with respect to H with probability at most $\exp(-\varepsilon^2 \ell n / 8)$. By a union bound over all assignments τ , we have

$$\mathbb{P}_H [\text{Some assignment } \tau \text{ is bad with respect to } H] \leq |D|^{n|D|} \exp\left(-\frac{\varepsilon^2 \ell n}{8}\right) = \exp(-\Omega(n)),$$

as long as $\ell \geq 100\varepsilon^{-2}|D| \ln |D|$. □

E Varieties Admitting Affine-type

In this section, we briefly discuss the background of [Theorem 3.2](#).

E.1 Varieties

Recall from [Definition 2.12](#) that the notion of an algebraic homomorphism is well-defined only when the operations of the source and target structures are placed in a bijective correspondence, and each pair of corresponding operations has the same arity. Such pairs of algebraic structures are said to share the same *signature*.

In universal algebra, it is often convenient to study entire classes of algebraic structures that share a common signature. For such a class, one can represent all operation sets uniformly by a single (possibly infinite) collection of *operation symbols*. Each operation symbol has a fixed finite arity but may represent different concrete operations in different structures. Even so, it is meaningful to consider *identities* among these operation symbols on a purely syntactic level. For instance, if $\mathcal{S}$ is a set of operation symbols and $s_1, s_2 \in \mathcal{S}$, then

$$s_1(x_1, s_2(x_2, x_3)) = s_2(s_1(x_1, x_2), x_3) \quad \forall x_1, x_2, x_3$$

is an example of an identity, which may or may not hold in a particular algebraic structure interpreting the symbols in $\mathcal{S}$.

Definition E.1. Let $\mathcal{S}$ be a (possibly infinite) set of operation symbols, each with a finite arity. A *term* is any finite expression built from variables $x_1, x_2, \dots$ and the operation symbols in $\mathcal{S}$. If t_1 and t_2 are terms, a syntactic statement of the form “ $t_1 = t_2$ for all $x_1, x_2, \dots$ ” is called an *identity*. Given a (possibly infinite) set of identities $\mathcal{T}$, the *variety* $\mathcal{V}(\mathcal{S}; \mathcal{T})$ is the class of all algebraic structures interpreting $\mathcal{S}$ such that every identity in $\mathcal{T}$ holds.

Definition E.2. For any algebraic structure $\mathbb{A}$, we define the variety generated by $\mathbb{A}$ to be $\mathcal{V}(\mathcal{S}; \mathcal{T})$, where $\mathcal{S}$ is the set of operation symbols interpreted by $\mathbb{A}$ and $\mathcal{T}$ is the set of identities in $\mathcal{S}$ that hold for $\mathbb{A}$. The variety generated by $\mathbb{A}$ is also denoted by $\mathcal{V}(\mathbb{A})$.

For example, it is not hard to see that any subalgebra or homomorphic image (defined in [Definition 2.13](#)) of $\mathbb{A}$ is contained in $\mathcal{V}(\mathbb{A})$.

E.2 Types of Strictly Simple Algebras

Following conventions in universal algebra, we refer to an algebraic structure simply as an “algebra.” An algebra of k elements is said to be *simple* if $k \geq 2$ and every homomorphic image of it has either 1 element or k elements. It is further said to be *strictly simple* if every subalgebra of it has either 1 element or k elements. Tame Congruence Theory, developed by Hobby and McKenzie [HM88], classified strictly simple algebras into five type: (1) the unary type, (2) the affine type, (3) the Boolean type, (4) the lattice type and (5) the semilattice type. Later, Szendrei [Sze92] provided a characterization of strictly simple *idempotent* algebras of all five types. We record the following immediate corollary of [Sze92, Theorem 6.1]; readers interested in further details are referred to [HM88, Sze92].

Proposition E.3 ([Sze92]). *If $\mathbb{A}$ is a strictly simple idempotent algebra of unary type or affine type, then there exists an Abelian group structure on the underlying set of $\mathbb{A}$ such that for any operation f in $\mathbb{A}$ of arity k , the equation*

$$f(x_1, \dots, x_k) - f(y_1, \dots, y_k) + f(z_1, \dots, z_k) = f(x_1 - y_1 + z_1, \dots, x_k - y_k + z_k)$$

holds for all $x_1, \dots, x_k, y_1, \dots, y_k, z_1, \dots, z_k \in \mathbb{A}$.

Using the idempotency condition, it is easy to deduce the following corollary.

Corollary E.4. *Any strictly simple idempotent algebra of unary type or affine type has an Abelian group G as its underlying set, and all of its operations belong to $\text{Pol}(3\text{SUM}_G)$.*

In general, a finite algebra (not necessarily strictly simple) is associated with a subset of the five possible types {unary, affine, Boolean, lattice, semilattice}. A variety $\mathcal{V}$ is said to *admit* a type if that type appears in the type set of at least one algebra belonging to $\mathcal{V}$. The following theorem by Valeriote [Val09] is a crucial component of [Theorem 3.2](#).

Theorem E.5 ([Val09, Proposition 3.1]). *Let $\mathbb{A}$ be a finite idempotent algebra. If the variety $\mathcal{V}(\mathbb{A})$ admits either the unary type or the affine type, then some homomorphic image of a subalgebra of $\mathbb{A}$ is a strictly simple algebra of unary or affine type.*

Note that homomorphic images and subalgebras of an idempotent algebra are also automatically idempotent.

The final ingredient in [Theorem 3.2](#) is the following characterization of bounded-width CSP templates in terms of the variety generated by its polymorphism algebra. It was conjectured by [LZ07] and proved by [BK14].

Theorem E.6 ([BK14, Conjecture 4.3]). *Let (D, Γ) be a relational structure that is a core. Then (D, Γ) has unbounded width if and only if the variety generated by the algebra $(D, \text{Pol}(\Gamma))$ admits the unary type or the affine type.*

It is then clear that [Theorem 3.2](#) follows from the combination of [Corollary E.4](#) and [Theorems E.5](#) and [E.6](#).