Definable Inapproximability: New Challenges for Duplicator

Albert Atserias[1]
Departament de Ciències de la Computació, Universitat Politècnica de Catalunya, Barcelona, Catalonia, Spain
atserias@cs.upc.edu
https://orcid.org/0000-0002-3732-1989

Anuj Dawar[2]
Department of Computer Science and Technology, University of Cambridge, UK
anuj.dawar@cl.cam.ac.uk
https://orcid.org/0000-0003-4014-8248

Abstract

We consider the hardness of approximation of optimization problems from the point of view of definability. For many NP-hard optimization problems it is known that, unless P = NP, no polynomial-time algorithm can give an approximate solution guaranteed to be within a fixed constant factor of the optimum. We show, in several such instances and without any complexity theoretic assumption, that no algorithm that is expressible in fixed-point logic with counting (FPC) can compute an approximate solution. Since important algorithmic techniques for approximation algorithms (such as linear or semidefinite programming) are expressible in FPC, this yields lower bounds on what can be achieved by such methods. The results are established by showing lower bounds on the number of variables required in first-order logic with counting to separate instances with a high optimum from those with a low optimum for fixed-size instances.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic, Theory of computation → Finite Model Theory

Keywords and phrases Descriptive Compleixty, Hardness of Approximation, MAX SAT, Vertex Cover, Fixed-point logic with counting

Digital Object Identifier 10.4230/LIPIcs.CSL.2018.7

Related Version A full version of this paper is available at [8], `https://arxiv.org/abs/1806.11307`.

Acknowledgements The research reported here was initiated at the Simons Institute for the Theory of Computing during the programme on Logical Structures in Computation in autumn 2016.

1 Introduction

Twenty years ago, the PCP theorem [4] transformed the landscape of complexity theory. It showed that if $\mathrm{P} \neq \mathrm{NP}$ then not only is it impossible to efficiently solve NP-hard problems exactly but for some of them it is also impossible to approximate the solution to within a

[1] Partially funded by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement ERC-2014-CoG 648276 (AUTAR) and MICCIN grant TIN2016-76573-C2-1P (TASSAT3).
[2] Supported in part by a Fellowship of the Alan Turing Institute.

© Albert Atserias and Anuj Dawar;
licensed under Creative Commons License CC-BY

constant factor. Consider for instance the problem MAX 3SAT. Here we are given a Boolean formula in 3CNF and we are asked to determine m^*, the maximum number of clauses that can be simultaneously satisfied by an assignment of Boolean values to its variables. It is a consequence of the PCP theorem that there is a constant $c < 1$ such that, assuming $\mathrm{P} \neq \mathrm{NP}$, no polynomial-time algorithm can be guaranteed to produce an assignment that satisfies at least cm^* clauses, or indeed determine the value of m^* up to a factor of c. The proof of the PCP theorem introduced sophisticated new techniques into complexity theory such as the *probabilistically checkable proofs* that gave the theorem its name. Over the years, stronger results were proved, improving the constant c and, by reductions, proving inapproximability results for a host of other NP-hard problems.

A structural theory of hardness of approximation was introduced by Papadimitriou and Yannakakis [23] who defined the class MAX SNP of approximation problems, with a definition rooted in descriptive complexity theory. They showed that for every problem in this class, there is a constant d such that a polynomial-time algorithm can find approximate solutions within a factor d of the optimum. At the same time, for all problems that are MAX SNP-hard, under approximation-preserving reductions defined by [23], there is a constant c such that no polynomial-time algorithm can approximate solutions within a factor c. This makes it a challenge, for each MAX SNP-complete problem, to determine the exact approximation ratio that is achievable by an efficient algorithm. In some cases, this has been pinned down exactly. For instance, for MAX 3SAT we know that there is a polynomial-time algorithm that will produce an assignment satisfying 7/8 of the clauses in any formula but, unless $\mathrm{P} = \mathrm{NP}$, there is no polynomial-time algorithm that is guaranteed to produce a solution within $7/8 + \epsilon$ of the optimal, for any $\epsilon > 0$ [16]. Another interesting case is MAX 3XOR, where we are given a formula which is the conjunction of clauses, each of which is the XOR of three literals. Here, satisfiability is decidable in polynomial time as the problem is essentially that of solving a system of linear equations over the two-element field. However, determining, for an unsatisfiable system, how many of its clauses can be simultaneously satisfied is MAX SNP-hard, and the exact approximation ratio that is achievable efficiently is known: unless $\mathrm{P} = \mathrm{NP}$, no polynomial-time algorithm can achieve an approximation ratio bounded above 1/2 [16].

To give a problem of another flavour, consider *minimum vertex cover*, the problem of finding, in a graph G, a minimum set S of vertices such that every edge is incident on a vertex in S. Let $\mathrm{VC}(G)$ denote the size of a minimum size vertex cover in G. There are algorithms that are guaranteed to find a vertex cover no larger than $2\mathrm{VC}(G)$ (this being a minimization problem, the approximation ratio is expressed as a number $c \geq 1$). It has been proved, by means of rather sophisticated reductions starting at the PCP theorem, that, unless $\mathrm{P} = \mathrm{NP}$, no polynomial-time algorithm can achieve a ratio better than 1.36 [14]. Very recent results announced in [20] improve this lower bound to $\sqrt{2}$. It is conjectured that indeed no such algorithm could achieve a ratio of $2 - \epsilon$ for arbitrarily small $\epsilon > 0$ but, as of our current knowledge, the right threshold constant could be somewhere between $\sqrt{2}$ and 2.

We approach these questions on the hardness of approximability from the point of view of definability. Our aim is to show that the tools of descriptive complexity can be brought to bear in showing lower bounds on the definability of approximations and that these definability lower bounds have consequences on understanding commonly used techniques in approximation algorithms.

A reference logic in descriptive complexity is fixed-point logic with counting, FPC. The class of problems definable in this logic form a proper subclass of the complexity class P. However, FPC is very expressive and many natural problems in P are expressible in this logic.

For instance, any polynomial-time decidable problem on a proper-minor closed class of graphs is expressible in FPC [15]. Also, problems that can be formulated as linear programming or semidefinite programming problems are in FPC [2, 9, 13]. At the same time, for many problems we are able to prove categorically, i.e., without complexity theoretic assumptions, that they are not definable in FPC. Among these are NP-complete problems like 3SAT, graph 3-colourability and Hamiltonicity (see [11]). We can also prove that certain problems in P are not in FPC, such as 3XOR.

A particularly interesting class of problems are the optimization problems known as MAX CSP or constraint maximization problems, where we are given a collection of constraints and the problem is to find the maximum number of constraints that can be simultaneously satisfied. When it comes to finding exact solutions, definability in FPC turns out to be an excellent guide to the tractability of such problems. It is known that each such problem is either in P *and* definable in FPC or it is NP-complete and provably *not* definable in FPC [12]. We would like to extend such results also to the *approximability* of such problems. This paper develops the methodology for doing so.

For MAX 3SAT, we prove, without any complexity theoretic assumption, that no algorithm expressible in FPC can achieve an approximation ratio of $7/8 + \epsilon$. The question seems ill-posed at first sight as FPC is a formalism for defining problems rather than expressing algorithms. We return to the precise formulation shortly, but first note that there is a sense in which FPC can express, say the ellipsoid method for solving linear programs [2]. This is the basis for showing that many commonly used algorithmic techniques for approximation problems, such as semidefinite programming relaxations, are also expressible in FPC. Thus, on the one hand, reductions from MAX SNP-hard problems show inapproximability by *any* polynomial-time algorithm, assuming $\mathrm{P} \neq \mathrm{NP}$. On the other hand, our results show, without the assumption, inapproximability by the most commonly used polynomial-time methods.

Undefinability of a class of structures $\mathcal{C}$ in FPC is typically established by showing that structures in $\mathcal{C}$ cannot be distinguished from structures not in $\mathcal{C}$ in C^k – first-order logic with counting and just k variables – for any fixed k. In the terminology of [13], $\mathcal{C}$ has unbounded *counting width*. On the other hand, hardness of approximation for a maximization problem is typically established by showing that every class that includes all instances with an optimum m^* and excludes all instances with an optimum less than cm^*, is NP-hard. Our method combines these two. We aim to show that any class separating instances with an optimum m^* from instances with an optimum less than cm^* has unbounded counting width. In general, we not only show that counting width is unbounded, but establish stronger bounds on how it grows with the size of instances, as such bounds are directly tied to lower bounds on semidefinite programming hierarchies [13]. This methodology poses new challenges for Spoiler-Duplicator games in finite model theory. Such games are typically played on pairs of structures that are minimally different. In the new setting, we need to show Duplicator winning strategies in games on pairs of structures that differ substantially, on some numeric parameters.

The PCP theorem is the *fons et origo* of results on hardness of approximation. It established the first provably NP-hard constant gap between the fully satisfiable instances of MAX 3SAT, i.e., those in which all clauses can be satisfied, and the less satisfiable ones, those where no more than $1 - \epsilon_0$ can be satisfied, for some explicit $\epsilon_0 > 0$. The gap between 1 and $1 - \epsilon_0$ was then amplified and also transferred to other problems by means of reductions. For us, the starting point is the problem MAX 3XOR. We are able to establish a definability gap between the satisfiable instances of this and instances in which little more than $3/4$ of the clauses can be satisified. The methods for establishing this *initial gap* are very different from that for the PCP theorem. We construct a k-locally satisfiable instance of MAX 3XOR which, by a random construction is at the same time highly unsatisfiable. We can then combine this with a construction adapted from [6] to obtain a gap that defeats any fixed counting width. With such an initial gap in hand, we can then amplify the gap and transfer it to other problems by means of reductions, just as in classical inapproximability. Our reductions have to preserve FPC definability and we mostly rely on first-order definable reductions. Indeed, many of the reductions used in the classical theory of approximability turn out to be first-order reductions but this requires close examination and proof.

By expressing the *long-code reductions* from [16] in first-order logic and composing them with our initial gap, we show optimal hardness for MAX 3SAT and MAX 3XOR. For the first, we show that FPC cannot achieve an approximation ratio of $7/8 + \epsilon$, even on satisfiable instances, and for the second it cannot achieve an approximation ratio of $1/2 + \epsilon$. These match known algorithmic lower bounds and are provably tight. For the vertex cover problem, direct reductions from these show that FPC cannot give an approximation better than 7/6. This can be improved, using the reduction of [14] to 1.36 and the details of this may be found in the full version of this paper [8]. It is possible that this could be improved to $\sqrt{2}$ using the recent breakthrough of [20] but we leave this to future work.

2 Preliminaries

We use $\mathbb{F}_2$ to denote the 2-element field. For any positive integer n, let $[n] := \{1, \dots, n\}$.

Logics and games. We assume familiarity with first-order logic FO. All our vocabularies are finite and relational, and all structures are finite. For a structure $\mathbb{A}$, we write A to denote its universe. We refer to fixed-point logic with counting FPC but the definition is not required for the technical development in this paper. Here, it suffices to consider the bounded variable fragments of first-order logic.

For a fixed positive integer k, we write L^k to denote the fragment of first-order logic in which every formula has at most k variables, free or bound. We also write $\exists L^{k,+}$ for the *existential positive* fragment of L^k. This consists of those formulas of L^k formed using only the positive Boolean connectives $\wedge$ and $\vee$, and existential quantification. FOC is the extension of first-order logic with *counting quantifiers*. For each natural number i, we have a quantifier $\exists^i$ where $\mathbb{A} \models \exists^i x\, \phi$ if, and only if, there are at least i distinct elements $a \in A$ such that $\mathbb{A} \models \phi[a/x]$. While the extension of first-order logic with counting quantifiers is no more expressive than FO itself, the presence of these quantifiers does affect the number of variables that are necessary to express a query. Let C^k denote the k-variable fragment of FOC in which no more than k variables appear, free or bound.

For two structures $\mathbb{A}$ and $\mathbb{B}$, we write $\mathbb{A} \equiv_{C^k} \mathbb{B}$ to denote that they are not distinguished by any sentence of C^k. All that we need to know about FPC is that for every formula ϕ of FPC there is a k such that if $\mathbb{A} \equiv_{C^k} \mathbb{B}$ then $\mathbb{A} \models \phi$ if, and only if, $\mathbb{B} \models \phi$. We also write $\mathbb{A} \Rrightarrow_k \mathbb{B}$ to denote that every sentence of $\exists L^{k,+}$ that is true in $\mathbb{A}$ is also true in $\mathbb{B}$. While $\equiv_{C^k}$ is an equivalence relation, $\Rrightarrow_k$ is reflexive and transitive but not symmetric. These relations have well established characterizations in terms of two-player pebble games. The relation $\Rrightarrow_k$ is characterized by the *existential k-pebble game* [21] and $\equiv_{C^k}$ by the k-pebble bijective game [17]. Rather than review the definitions here, we refer the reader to the sources.

For undirected graphs, the relation $\equiv_{C^2}$ has a simple combinatorial characterization in terms of *vertex refinement* (see [19]). For any graph G, there is a coarsest partition $C_1, \dots, C_m$ of the vertices of G such that for each $1 \leq i, j \leq m$ there exists δ_{ij} such that each

$v \in C_i$ has exactly δ_{ij} neighbours in C_j. Let H be another graph and $D_1, \dots D_{m'}$ be the corresponding partition of H with constants γ_{ij}. Then $G \equiv_{C^2} H$ if, and only if, $m = m'$ and there is a permutation $h \in \mathrm{Sym}_m$ such that $|C_i| = |D_{h(i)}|$ and $\delta_{ij} = \gamma_{h(i)h(j)}$ for all i and j.

Let $\mathcal{C}$ be a class of structures and for any $n \in \mathbb{N}$, let $\mathcal{C}_n$ denote the structures in $\mathcal{C}$ with at most n elements. The *counting width* of $\mathcal{C}$ [13] is the function $k : \mathbb{N} \to \mathbb{N}$ where $k(n)$ is the smallest value such that for any $\mathbb{A} \in \mathcal{C}_n$ and any $\mathbb{B} \notin \mathcal{C}$, we have $\mathbb{A} \not\equiv_{C^{k(n)}} \mathbb{B}$. Note that $k(n) \leq n$. Because $\mathbb{A} \not\equiv_{C^1} \mathbb{B}$ whenever $\mathbb{A}$ and $\mathbb{B}$ have different numbers of elements, $k(n)$ is also the smallest value such that $\mathcal{C}_n$ is a union of $\equiv_{C^{k(n)}}$-classes. In particular, it follows that the counting width of $\mathcal{C}$ is the same as that of its complement. For $k : \mathbb{N} \to \mathbb{N}$, we say that two disjoint classes $\mathcal{C}$ and $\mathcal{D}$ are *C^k-separable* if whenever $\mathbb{A} \in \mathcal{C}_n$ and $\mathbb{B} \in \mathcal{D}_n$, then we have $\mathbb{A} \not\equiv_{C^{k(n)}} \mathbb{B}$. Equivalently $\mathcal{C}$ and $\mathcal{D}$ are *C^k-separable* if there is a class $\mathcal{E}$ of counting width at most k such that $\mathcal{C} \subseteq \mathcal{E}$ and $\mathcal{D} \subseteq \overline{\mathcal{E}}$.

Interpretations. Consider two signatures σ and τ. A *d-ary* FO*-interpretation of τ in σ* is a sequence of first-order formulas in vocabulary σ consisting of: (i) a formula $\delta(\overline{x})$; (ii) a formula $\varepsilon(\overline{x}, \overline{y})$; (iii) for each relation symbol $R \in \tau$ of arity k, a formula $\phi_R(\overline{x}_1, \dots, \overline{x}_k)$; and (iv) for each constant symbol $c \in \tau$, a formula $\gamma_c(\overline{x})$, where each $\overline{x}$, $\overline{y}$ or $\overline{x}_i$ is a d-tuple of variables. We call d the *dimension* of the interpretation. If $d = 1$, we say that the interpretaion is *linear*. We say that an interpretation Θ associates a τ-structure $\mathbb{B}$ to a σ-structure $\mathbb{A}$ if there is a map h from $\{\overline{a} \in A^d \mid \mathbb{A} \models \delta[\overline{a}]\}$ to the universe B of $\mathbb{B}$ such that: (i) h is surjective onto B; (ii) $h(\overline{a}_1) = h(\overline{a}_2)$ if, and only if, $\mathbb{A} \models \varepsilon[\overline{a}_1, \overline{a}_2]$; (iii) $R^{\mathbb{B}}(h(\overline{a}_1), \dots, h(\overline{a}_k))$ if, and only if, $\mathbb{A} \models \phi_R[\overline{a}_1, \dots, \overline{a}_k]$; and (iv) $h(\overline{a}) = c^{\mathbb{B}}$ if, and only if, $\mathbb{A} \models \gamma_c[\overline{a}]$. Note that an interpretation Θ associates a τ-structure with $\mathbb{A}$ only if ε defines an equivalence relation on A^d that is a congruence with respect to the relations defined by the formulae ϕ_R and γ_c. In such cases, however, $\mathbb{B}$ is uniquely defined up to isomorphism and we write $\Theta(\mathbb{A}) = \mathbb{B}$. It is also worth noting that the size of $\mathbb{B}$ is at most n^d, if $\mathbb{A}$ is of size n. But, it may in fact be smaller. We call an interpretation p-bounded, for a polynomial p, if $|\mathbb{B}| \leq p(|\mathbb{A}|)$, and say the interpretation is *linearly bounded* if p is linear. Every linear interpretation is linearly bounded, but the converse is not necessarily the case.

For a class of structures $\mathcal{C}$ and an interpretation Θ, we write $\Theta(\mathcal{C})$ to denote the class $\{\Theta(\mathbb{A}) \mid \mathbb{A} \in \mathcal{C}\}$. We mainly use interpretations to define reductions between classes of structures. These allow us to transfer bounds on separability, by the following lemma, which is established by simply composing formulas. The details may be found in Appendix A.

▶ **Lemma 2.1.** *Let Θ be a p-bounded interpretation of dimension d and let t be the maximum number of variables appearing in any formula of Θ. If $\mathcal{C}$ and $\mathcal{D}$ are two disjoint classes of structures such that $\Theta(\mathcal{C})$ and $\Theta(\mathcal{D})$ are C^k-separable, then $\mathcal{C}$ and $\mathcal{D}$ are $C^{dk(p(n))+t}$-separable.*

When we wish to define a reduction from a class $\mathcal{C}$ by a first-order interpretation, it suffices to give an interpretation Θ for all structures in $\mathcal{C}$ with at least two elements (or, indeed, at least k elements for any fixed k). This is because we can define an arbitrary map on a finite set of structures by a first-order formula, so we just need to take the disjunction of Θ with the formula that defines the required interpretation on the structures with one element. With this in mind, we define the method of *finite expansions* which gives us interpretations Θ that take a structure $\mathbb{A}$ with universe A to a structure with a universe consisting of l labelled disjoint copies of S for some definable subset S of A. Note that Θ would not, in general, be linear, but it is linearly bounded.

So, fix a value l, and let t be the least integer such that $l \leq 2^t$. In a structure $\mathbb{A}$ with at least two elements, we say that a $t+1$-tuple of elements $(a_1, \dots, a_{t+1})$ *codes* an integer

$i \in [2^t]$ if $b_1 \cdots b_t$ is the binary representation of $i-1$ and $b_j = 1$ if, and only if, $a_{j+1} \neq a_1$. For each i, we can clearly define a formula $\gamma_i(\overline{y})$ with $t+1$ free variables that defines those tuples that code i. Now, for any formula $\sigma(x)$, let $\delta(x, \overline{y})$ be the formula $\sigma(x) \wedge \bigvee_{i \leq l} \gamma_i(\overline{y})$ and let $\epsilon(x_1, \overline{y}_1, x_2, \overline{y}_2)$ be the formula $x_1 = x_2 \wedge \bigvee_i \gamma_i(\overline{y}_1) \wedge \gamma_i(\overline{y}_2)$. In other words, δ picks out those $t+2$ tuples $(s, \overline{a})$ where s satisfies σ and $\overline{a}$ codes an integer in $[l]$, and ϵ identifies distinct tuples which have the same s and the same integer l. An interpretation using these can be seen to yield a structure with l disjoint copies of the set of elements of $\mathbb{A}$ satisfying σ.

3 The Basic Gap Construction

The problems 3SAT and 3XOR both ask to decide if a formula consisting of the conjunction of Boolean constraints each on exactly three Boolean variables is satisfiable. In 3SAT the constraints are disjunctions of literals on three distinct variables. In 3XOR the constraints are parities of three distinct variables. Both problems are known to have unbounded counting width [6]: the class of satisfiable instances cannot be separated in C^k, for bounded k, from the class of unsatisfiable ones. Our aim is to show that this result can be strengthened to show that the class of satisfiable instances is not C^k-separable (for constant or, indeed, moderately growing values of k) from the class of instances that are *highly unsatisfiable*, meaning that no assignment to the variables can satisfy more than a fraction s of the constraints for some fixed $s \in (0, 1)$. In this section, we give a basic construction for 3XOR, based on that in [6], that establishes this for any $s > 3/4$, with a lower bound on the value of k that is linear in the number of variables in the system.

3.1 Systems of constraints

Let Γ be a finite set of relations over a finite domain D, also called a *constraint language*. Let $I = \{c_1, \dots, c_m\}$ be a collection (multi-set) of constraints, each of the form $R(x_{i_1}, \dots, x_{i_k})$, where R is a k-ary relation in Γ, and $x_{i_1}, \dots, x_{i_k}$ are k distinct D-valued variables from a set $x_1, \dots, x_n$ of n variables. For $c \in [0, 1]$, we say that the system I is c-satisfiable if there is an assignment $f : \{x_1, \dots, x_n\} \to D$ that satisfies at least cm constraints; i.e., that satisfies $(f(x_{i_1}), \dots, f(x_{i_k})) \in R$ for at least cm constraints $R(x_{i_1}, \dots, x_{i_k})$ from I. Note that, as we are counting the number of satisfied constraints, multiplicities matter and this is why we have multi-sets rather than sets of constraints.

We think of a system $I = \{c_1, \dots, c_m\}$ over the constraint language Γ as a finite structure in two ways. In the first encoding, the universe is the disjoint union of $x_1, \dots, x_n$ and $c_1, \dots, c_m$. The vocabulary includes binary relations $E_1, E_2, \dots$ such that $E_i(x, c)$ holds if the constraint c has arity at least i and x is the ith variable in c. The vocabulary also includes a unary relation Z_R for each relation R in Γ such that $Z_R(c)$ holds if c is an R-constraint: a constraint of the form $R(x_{i_1}, \dots, x_{i_k})$ for some variables $x_{i_1}, \dots, x_{i_k}$, where k is the arity of R. In the second encoding, the universe is just the set of variables $x_1, \dots, x_n$, and the vocabulary includes a k-ary relation symbol R for each k-ary relation R in Γ, such that $R(x_{i_1}, \dots, x_{i_k})$ holds if this is one of the constraints in the collection $c_1, \dots, c_m$. Note that in this second encoding the collection of constraints is treated as a set. In particular, the multiplicity of constraints is lost, which could affect its c-satisfiability.

The constraint language Γ is also encoded as a finite structure in two ways. In the first encoding the domain is $D^{\leq r} = D \cup D^2 \cup D^3 \cup \cdots \cup D^r$, where r is the maximal arity of a relation in Γ. The relations $E_1, E_2, \dots$ are interpreted by the projections: $E_i(b, (b_1, \dots, b_k))$ holds for $b \in D$ and $(b_1, \dots, b_k) \in D^k$ if, and only if, $i \leq k$ and $b = b_i$. The relations Z_R are interpreted by the relation R itself as a unary relation over the universe: $Z_R((b_1, \dots, b_k))$

holds if k is the arity of R and $(b_1, \ldots, b_k)$ belongs to R. In the second encoding, the universe is just D, and the relation symbol R is interpreted by R itself. Where it causes no confusion, we do not distinguish between a constraint language Γ and the structure that encodes it, and similarly between an instance I and its encoding structure.

It is easily seen that, in both encodings as finite structures, a system I over Γ is satisfiable if, and only if, there is a homomorphism from the structure that encodes I to the structure that encodes Γ. We say that the system is *k-locally satisfiable* if $I \Rrightarrow_k \Gamma$.

For 3SAT, the constraint language is denoted Γ_{3SAT}. It has domain $D = \{0,1\}$ and the relations are the eight relations $R_1, \ldots, R_8 \subseteq \{0,1\}^3$ defined by the eight possible clauses on three variables. For 3XOR, the constraint language is denoted Γ_{3XOR}. It also has domain $D = \{0,1\}$ and the relations are the two relations $R_0, R_1 \subseteq \{0,1\}^3$ defined by the two possible linear equations $x + y + z = b$ with three variables over $\mathbb{F}_2 = \{0,1\}$. Accordingly, 3XOR instances can be identified with systems of linear equations $Ax = b$ over $\mathbb{F}_2$.

3.2 Gap construction

We now focus on 3XOR and hence on systems of linear equations over $\mathbb{F}_2$. A starting point for us is the following construction which allows us to convert any *k-locally satisfiable* system of equations into a pair of systems that are $\equiv_{C^k}$-indistinguishable. See [1, Prop. 32] for a related construction, which is inspired by the proof in [6] that satisfiability of systems of linear equations over $\mathbb{F}_2$ is not invariant under $\equiv_{C^k}$ for any k.

For any instance I of 3XOR we define another instance $G(I)$ of 3XOR which has two variables x_j^0 and x_j^1 for each variable x_j of I. For each equation $x_j + x_k + x_l = b$ in I, we have eight equations in $G(I)$ given by the eight possible values of $a_1, a_2, a_3 \in \{0,1\}$ in $x_j^{a_1} + x_k^{a_2} + x_l^{a_3} = b + a_1 + a_2 + a_3$. We now establish some properties of this construction.

▶ **Lemma 3.1.** *For any instance I of* 3XOR *and any $c, s \in [0,1]$, the following hold:*
1. *if I is c-satisfiable, then $G(I)$ is c-satisfiable,*
2. *if I is not s-satisfiable, then $G(I)$ is not $(1/2 + s/2)$-satisfiable.*

Proof. In Appendix B. ◀

If I is the system $Ax = b$, then the homogeneous companion of I is the system $Ax = 0$, which we denote I^0. Since any homogeneous system is satisfiable, the system $G(I^0)$ is satisfiable for any I by Lemma 3.1. We show that, despite this, as long as I is locally satisfiable, then $G(I)$ is hard to distinguish from its homogeneous companion $G(I^0)$.

▶ **Lemma 3.2.** *For any instance I of* 3XOR *and any k, if I is k-locally satisfiable, then $G(I) \equiv_{C^k} G(I^0)$.*

Proof. In Appendix B. ◀

To apply this construction to get a gap, we need the following fact. Entirely analogous claims have been known and proved in the context of the proof complexity of propositional resolution; indeed, our proof builds on the methods for resolution width [10], and their relationship to existential pebble games from [5, 7].

In the proof, we need the notion of a graph G that is a *bipartite unique-neighbour expander graph with parameters* (m, n, d, s, e) where m, n, d and s are integer parameters with $s < n$ and e is a positive real number. What this means is that G is a bipartite graph with parts U and V with m and n vertices respectively; each $u \in U$ has exactly d neighbours in V; and for every $A \subseteq U$ with $|A| \leq s$ we have $|\partial A| \geq e|A|$, where $|\partial A|$ denotes the set of vertices in V that are *unique neighbours* of A; i.e., they are neighbours of a single vertex in A.

▶ **Lemma 3.3.** *For every $\epsilon > 0$ there exist an integer $c > 0$ and a $\gamma > 0$ such that for every sufficiently large integer n there is an instance I of* 3XOR *with n variables and cn equations such that I is not $(1/2 + \epsilon)$-satisfiable and I is k-locally satisfiable for $k \leq \gamma n$.*

Proof. Fix $\epsilon > 0$ and let $c > 1/\epsilon^2$. Let $n \geq 2$ be sufficiently large that we can construct a graph G that is a bipartite unique-neighbour expander graph with parameters $(cn, n, 3, \alpha n, e)$ for a fixed $\alpha > 0$. For the existence of such graphs with these parameters see [26, Chaper 4]. For each $b = (b_u : u \in U) \in \{0,1\}^U$, we produce an instance I of 3XOR by introducing one variable x_v for each $v \in V$, and one equation $e_u : x_{v_1(u)} + x_{v_2(u)} + x_{v_3(u)} = b_u$ for each $u \in U$. We claim that there is at least one choice of $b \in \{0,1\}^U$ that makes I be not $(1/2+\epsilon)$-satisfiable. We also show that every choice of $b \in \{0,1\}^U$ gives that I is k-locally satisfiable for $k \leq \gamma n$ with $\gamma = e\alpha/9$.

▶ **Claim 3.4.** *There exists $b \in \{0,1\}^U$ such that system I is not $(1/2 + \epsilon)$-satisfiable.*

Proof. We prove that such a b exists by the probabilistic method: a random $b \in \{0,1\}^U$ has a good chance of making I be not $(1/2+\epsilon)$-satisfiable. For each assignment $f : \{x_v : v \in V\} \to \{0,1\}$ and each $u \in U$, let $X_{f,u}$ be the indicator random variable for the event that $f(x_{v_1(u)}) + f(x_{v_2(u)}) + f(x_{v_3(u)}) = b_u$; i.e., for the event that f satisfies the equation $x_{v_1(u)} + x_{v_2(u)} + x_{v_3(u)} = b_u$. The probability of this event is $1/2$, and all such events, as u ranges over U, are mutually independent. Thus, setting $X_f = \sum_{u \in U} X_{f,u}$, we have that X_f is a binomial random variable with expectation $\mathbb{E}[X_f] = m/2$. By Hoeffding's inequality, the probability that $X_f - \mathbb{E}[X_f] \geq t$ is at most $e^{-2t^2/m}$. In particular, the probability that $X_f \geq (1/2+\epsilon)m$ is at most $e^{-2\epsilon^2 m}$. By the union bound, the probability that some f satisfies $X_f \geq (1/2+\epsilon)m$ is at most $2^n e^{-2\epsilon^2 m}$. Since $m = cn$ and $c > 1/\epsilon^2$ this probability is at most $2^n e^{-2n}$ and so approaches 0 as n grows. Indeed, it is less than $1/2$ for all values of $n \geq 2$. Thus, for any large enough n there exists a b such that I is not $(1/2+\epsilon)$-satisfiable. ◀

▶ **Claim 3.5.** *For every $b \in \{0,1\}^U$, every set of at most αn equations from I is satisfiable.*

Proof. For each $A \subseteq U$, let e_A be the set of equations that are indexed by vertices in A, and let v_A be the set of variables that appear in e_A. We prove, by induction on $t \leq \alpha n$, that if $A \subseteq U$ and $|A| = t$, then there exists an assignment that sets all the variables in v_A and that satisfies all the equations in e_A. For $t = 0$ the claim is obvious. Assume now that $1 \leq t \leq \alpha n$ and let A be a subset of U of cardinality t. Then $|\partial A| \geq e|A| > 0$. Let v_0 be some element in ∂A and let $u_0 \in A$ be the unique neighbour of v_0 in A. The induction hypothesis applied to $B = A \setminus \{u_0\}$ gives an assignment g that sets all the variables in v_B and satisfies all the equations in e_B. The assignment g may assign some of the variables of the equation e_{u_0}, but not all, since v_0 is not a neighbour of any vertex in B. Let f be the unique extension of g that first sets all the variables in $v_A \setminus (v_B \cup \{x_{v_0}\})$ to 0, and then sets x_{v_0} to the unique value that satisfies the equation e_{u_0}. This assignment sets all the variables in v_A and satisfies all the equations in e_A. The proof is complete. ◀

▶ **Claim 3.6.** *For every $b \in \{0,1\}^U$ and $k \leq \gamma n$, the instance I is k-locally satisfiable.*

Proof. If I is satisfiable, then Duplicator certainly has a winning strategy and there is nothing to prove. Assume then that I is unsatisfiable and let I' be a minimally unsatisfiable subsystem; a subset of the equations of I that is unsatisfiable and every proper subset of it is satisfiable. For each equation $e_u : x_{v_1(u)} + x_{v_2(u)} + x_{v_3(u)} = b_u$ of I, let F_u be the four clauses $\{x^{(a_1)}_{v_1(u)}, x^{(a_2)}_{v_2(u)}, x^{(a_3)}_{v_3(u)}\}$ with $a_1, a_2, a_3 \in \mathbb{F}_2$ with $a_1 + a_2 + a_3 = b_u$, where $z^{(a)}$ stands for the negative literal $\neg z$ if $e = 0$ and the positive literal z if $e = 1$. Let F be the 3CNF formula that is the union of all the F_u as u ranges over U. Observe that F is an unsatisfiable 3CNF. We intend to apply Theorem 5.9 from [10] to it.

Let $\mathcal{A}$ be the collection of all Boolean functions $f_u : \{0,1\}^V \to \{0,1\}$ defined by

$$f_u(x_v : v \in V) = x_{v_1(u)} + x_{v_2(u)} + x_{v_3(u)} + b_u \mod 2,$$

for $u \in U$. Each function in $\mathcal{A}$ is sensitive in the sense of Definition 5.5 from [10], and compatible with F in the sense of Definition 5.3 from [10]. Moreover, if $\mathcal{A}_0 \subseteq \mathcal{A}$ is the set of functions that corresponds to the minimally unsatisfiable subsystem I' of I, then its cardinality m_0 satisfies $m_0 > \alpha n$ by Claim 3.5. It follows that the expansion $e(\mathcal{A})$ in the sense of Definition 5.8 from [10] is at least $e\alpha n/3$. By Theorem 5.9 in [10], every resolution refutation of F requires width at least $e\alpha n/3$, and hence at least $3k$ since $k \leq \gamma n = e\alpha n/9$. By Theorem 2 in [7], Duplicator has a winning strategy for the existential $3k$-pebble game played on the structures F and the constraint language $\Gamma_{3\mathrm{SAT}}$ of 3SAT, in the second encoding discussed in Section 3.1. We use this winning strategy to design a winning strategy for Duplicator in the existential k-pebble game played on I and $\Gamma_{3\mathrm{XOR}}$.

While playing the game on I, Duplicator plays the game on F on the side and keeps the invariant that each pebbled variable in the game on I is also pebbled in the side game, and each pebbled equation in the game on I has its three variables pebbled in the side game. Whenever a new variable is pebbled in the game on I, Duplicator pebbles the same variable in the side game, and copies the answer from its strategy on it. Whenever a new equation is pebbled in the game on I, Duplicator pebbles its three variables in the side game, and answers the pebbled equation accordingly from its strategy. Since at each position of the game on I there are no more than k pebbles on the board, at each time during the simulation the side game has no more than $3k$ pebbles on the board. This shows that the simulation can be carried on forever and the proof is complete. ◀

This completes the proof of Lemma 3.3. ◀

We can now prove our first two gap theorems.

▶ **Theorem 3.7.** *For any $\epsilon > 0$, if $\mathcal{C}$ is the collection of* 3XOR *instances that are satisfiable and $\mathcal{D}$ is the collection of* 3XOR *instances that are not $(3/4 + \epsilon)$-satisfiable, then $\mathcal{C}$ and $\mathcal{D}$ are not C^k-separable for any $k = o(n)$.*

Proof. By Lemma 3.3, there is a family of systems $(S_k)_{k\geq 1}$ with $O(k)$ variables and equations such that S_k is k-locally satisfiable but not $(1/2 + 2\epsilon)$-satisfiable. Let $I_k^1 = G(S_k)$ and $I_k^0 = G(S_k^0)$. Note that, by Lemma 3.1, the system I_k^0 is satisfiable and I_k^1 is not $(3/4 + \epsilon)$-satisfiable. However, $I_k^0 \equiv_{C^k} I_k^1$ by Lemma 3.2. Since each of I_k^0 and I_k^1 has two variables for each variable in S_k and eight equations for each equation in S_k, they also have $O(k)$ variables and equations and the result follows. ◀

▶ **Theorem 3.8.** *For any $\epsilon > 0$, if $\mathcal{C}$ is the collection of* 3SAT *instances that are satisfiable and $\mathcal{D}$ is the collection of* 3SAT *instances that are not $(15/16 + \epsilon)$-satisfiable, then $\mathcal{C}$ and $\mathcal{D}$ are not C^k-separable for any $k = o(n)$.*

Proof. Consider again the reduction Θ from 3XOR to 3SAT given by translating each equation into a conjunction of four clauses. Thus $x + y + z = d$ translates into the four clauses $\{x^{(a)}, y^{(b)}, z^{(c)}\}$ with $a, b, c \in \mathbb{F}_2$ with $a + b + c = d$, where $z^{(e)}$ stands for the negative literal $\neg z$ if $e = 0$ and the positive literal z if $e = 1$. This is easily defined in first-order logic. As the set of variables in I is the same as in $\Theta(I)$, it is linearly bounded. We claim that applying Θ to Theorem 3.7 with ϵ reset to $\epsilon/4$ gives the theorem through Lemma 2.1. First, it is clear that if I is a 3XOR instance that is satisfiable, then $\Theta(I)$ is also satisfiable. Now, suppose that I is a system of m equations that is not $(3/4 + \epsilon/4)$-satisfiable, and let g be an assignment of truth values to the variables X of $\Theta(I)$. Applied to I, the assignment g falsifies at least $(1/4 - \epsilon/4)m$ of the equations. For each equation, g must falsify at least one of the four corresponding clauses in $\Theta(I)$. Thus, g falsifies at least $(1/4 - \epsilon/4)m$ clauses in $\Theta(I)$ and so satisfies at most $4m - (1/4 - \epsilon/4)m = (15/16 + \epsilon) \cdot 4m$ of the $4m$ clauses. ◀

4 Amplifying the Gap

In this section we show that certain reductions from the theory of inapproximability can be expressed as FO-interpretations, allowing us to derive optimal and unconditional undefinability results that match the optimal NP-hardness results from [16].

4.1 Parallel repetition

An instance I of the LABEL COVER problem is given by two disjoint sets of variables U and V with domains of values A and B, respectively, a predicate $P : U \times V \times A \times B \to \{0, 1\}$, and an assignment of weights $W : U \times V \to \mathbb{N}$. If all the non-zero weights $W(u, v)$ are equal, then the instance is said to have *uniform weights*. If for all $u \in U$ the sums $W(u) := \sum_{v \in V} W(u, v)$ of incident weights are equal, then the instance is called *left-regular*. A *right-regular* instance is defined analogously in terms of $W(v) := \sum_{u \in U} W(u, v)$. The instance is a *projection game* if for every $(u, v) \in U \times V$ with $W(u, v) \neq 0$ it holds that for every $a \in A$ there is exactly one $b \in B$ satisfying $P(u, v, a, b) = 1$. It is called a *unique game* if $|A| = |B|$ and it is a projection game both ways: from A to B, and from B to A. The instance is said to have parameters (m, n, p, q) if $|U| = m$, $|V| = n$, $|A| = p$ and $|B| = q$. Its *domain size* is $p + q$.

A value-assignment for an instance I is a pair of functions $f : U \to A$ and $g : V \to B$. The weight $v(f, g)$ of the value-assignment (f, g) is the total weight of the pairs $(u, v) \in U \times V$ satisfying the constraint $P(u, v, f(u), g(v)) = 1$; i.e.,

$$v(f, g) = \sum_{(u,v) \in U \times V} W(u, v) P(u, v, f(u), g(v)). \tag{1}$$

For $c \in [0, 1]$, we say that the instance is c-satisfiable if there is a value-assignment whose weight is at least $c \cdot W_0$, where $W_0 = \sum_{(u,v) \in U \times V} W(u, v)$ is the maximum possible weight. We call it satisfiable if it is 1-satisfiable.

The *bipartite reduction* takes an instance I of 3XOR and produces a projection game instance $L(I)$ of LABEL COVER defined as follows. The sets U and V are the set of equations in I and the set of variables in I, respectively. The weight $W(u, v)$ is 1 if v is one of the variables in the equation u, and 0 otherwise. The domains of values associated to U and V are $A = \{(a_1, a_2, a_3) \in \mathbb{F}_2^3 : a_1 + a_2 + a_3 = 0\}$ and $B = \mathbb{F}_2$, respectively. The predicate P associates to the pair (u, v), where u is the equation $v_1 + v_2 + v_3 = b$ and $v = v_i$ for $i \in \{1, 2, 3\}$, the set of pairs $((a_1, a_2, a_3), a) \in A \times B$ satisfying $a = a_i + b$. In other words, $P(u, v, (a_1, a_2, a_3), a) = 1$ if, and only if, v appears in the equation u, and if u is $v_1 + v_2 + v_3 = b$ and $v = v_i$, then the (partial) assignment $\{v_1 \mapsto a_1 + b, v_2 \mapsto a_2 + b, v_3 \mapsto a_3 + b\}$, which satisfies the equation $v_1 + v_2 + v_3 = b$ by construction, agrees with the (partial) assignment $\{v_i \mapsto a\}$. Clearly, this defines a projection game.

▶ **Lemma 4.1.** *For every instance I of* 3XOR *and every $c, s \in [0,1]$, the following hold:*
1. *if I is c-satisfiable, then $L(I)$ is c-satisfiable,*
2. *if I is not s-satisfiable, then $L(I)$ is not $(s+2)/3$-satisfiable.*

Moreover, $L(I)$ is a left-regular projection game that has uniform weights.

Proof. Let m be the number of equations in I, so $L(I)$ has exactly $3m$ pairs (u,v) of unit weight. Such pairs are called constraints. For proving 1, let h be an assignment for I that satisfies at least cm of the m equations in I. For each equation u in I, say $v_1 + v_2 + v_3 = b$, define $f(u) = (h(v_1) + b, h(v_2) + b, h(v_3) + b)$ if h satisfies $v_1 + v_2 + v_3 = b$, and define $f(u) = (0,0,0)$ otherwise. For each variable v in I, define $g(v) = h(v)$. Each equation in I gives rise to exactly three constraints in $L(I)$, and if the equation is satisfied by h, then all three constraints associated to it in $L(I)$ are satisfied by (f,g). Thus (f,g) satisfies at least $3cm$ of the $3m$ constraints in $L(I)$, so $L(I)$ is c-satisfiable. For proving 2, let (f,g) be an assigment for $L(I)$ that satisfies at least $(s+2)m$ of the $3m$ constraints in $L(I)$. For each variable v in I, define $h(v) = g(v)$. Let t be the number of equations of I that are satisfied by h. In terms of t, the assignment (f,g) satisfies at most $3t + 2(m-t)$ of the $3m$ constraints of $L(I)$. Thus $t \geq sm$, so I is s-satisfiable. ◀

The *parallel repetition reduction* takes an instance I of LABEL COVER, and a positive integer $t \geq 1$, and produces another instance $R(I,t)$ of LABEL COVER defined as follows. Let U and V be the sets of variables in I and let $W : U \times V \to \mathbb{N}$ be the weight assignment. The sets of variables of $R(I,t)$ are U^t and V^t. For $\overline{u} = (u_1, \ldots, u_t) \in U^t$ and $\overline{v} = (v_1, \ldots, v_t) \in V^t$, the weight $W(\overline{u}, \overline{v})$ is defined as $\prod_{i=1}^{t} W(u_i, v_i)$. If A and B are the domains of values associated to U and V, then the domains of values associated to U^t and V^t are A^t and B^t respectively. For $\overline{u} = (u_1, \ldots, u_t) \in U^t$, $\overline{v} = (v_1, \ldots, v_t) \in V^t$, $\overline{a} = (a_1, \ldots, a_t) \in A^t$ and $\overline{b} = (b_1, \ldots, b_t) \in B^t$, the predicate $P(\overline{u}, \overline{v}, \overline{a}, \overline{b})$ is defined as $\prod_{i=1}^{t} P(u_i, v_i, a_i, b_i)$. Observe that this definition guarantees that if I is a projection game, then so is $R(I,t)$.

▶ **Theorem 4.2** (Parallel Repetition Theorem [24, 18])**.** *There exists a constant $\alpha > 0$ such that for every instance I of* LABEL COVER *with domain size at most $d \geq 1$, every $s \in [0,1]$ and every $t \geq 1$ the following hold:*
1. *if I is satisfiable, then $R(I,t)$ is satisfiable,*
2. *if I is not s-satisfiable, then $R(I,t)$ is not $(1-(1-s)^3)^{\alpha t/d}$-satisfiable.*

Moreover, if I is a projection game, left-regular, right-regular, or has uniform weights, then so is $R(I,t)$.

Although it is the case that the bipartite and the parallel repetition reductions are both FO-interpretations, we do not need to formulate this. Instead, we show the FO-definability of the composition of these reductions with the long-code reductions that we discuss next.

4.2 First long-code reduction

The *first long-code reduction* that we consider takes a projection game instance I of LABEL COVER and a rational $\epsilon \in [0,1]$ and produces an instance $C(I,\epsilon)$ of 3XOR defined as follows. Let U and V be the sets of variables of sizes m and n, respectively, with associated domains of values $A = [p]$ and $B = [q]$, let $W : U \times V \to \mathbb{N}$ be the weight assignment, let $P : U \times V \times A \times B \to \{0,1\}$ be the predicate of I, and for each $(u,v) \in U \times V$ with $W(u,v) \neq 0$ and each $a \in A$ let $\pi_{u,v}(a)$ be the unique value $b \in B$ that satisfies $P(u,v,a,b) = 1$. The existence of such a function $\pi_{u,v} : A \to B$ is guaranteed from the assumption that I is a projection game. The set of variables of $C(I,\epsilon)$ includes one variable $u(a)$ for each $u \in U$ and $a \in \mathbb{F}_2^{p-1}$, and one variable $v(b)$ for each $v \in V$ and $b \in \mathbb{F}_2^{q-1}$, for a total of $m2^{p-1} + n2^{q-1}$ variables. Before we are able to define the set of equations of $C(I, \epsilon)$ we need a piece of notation. For a vector $z = (z_1, \ldots, z_d) \in \mathbb{F}_2^d$ of dimension $d \geq 2$, we write $S(z) = z_d$ and $F(z) = (z_1 + S(z), \ldots, z_{d-1} + S(z))$. Note that $S(z)$ is a single field element, and $F(z)$ is a vector of dimension $d-1$. With this notation, the set of equations of $C(I, \epsilon)$ includes $W(u,v) \cdot M^q \cdot \epsilon^D \cdot (1-\epsilon)^{q-D}$ copies of the equation $v(F(x)) + u(F(y)) + u(F(z)) = S(x) + S(y) + S(z)$ for each $(u,v) \in U \times V$, each $x \in \mathbb{F}_2^q$ and each $y, z \in \mathbb{F}_2^p$, where M is the denominator of $\epsilon = N/M$ reduced to lowest terms, D is the number of positions $i \in [p]$ such that $z_i \neq x_{\pi(i)} + y_i$, and $\pi = \pi_{u,v}$ if $W(u,v) \neq 0$.

▶ **Theorem 4.3** (Håstad 3-Query Linear Test [16]). *For every $s, \epsilon \in [0,1]$ with $\epsilon > 0$ and $s > 0$ and every projection game instance I of* LABEL COVER*, the following hold:*
1. *if I is satisfiable, then $C(I, \epsilon)$ is $(1-\epsilon)$-satisfiable,*
2. *if I is not s-satisfiable, then $C(I, \epsilon)$ is not $(1/2 + (s/\epsilon)^{1/2}/4)$-satisfiable.*

The proof of Theorem 4.3 follows from Lemmas 5.1 and 5.2 in [16]. There are notational differences that may obscure this and a detailed explanation is provided in Appendix C.

Next, by composing Lemma 4.1, Theorem 4.2, and Theorem 4.3 with the appropriate parameters we get the following:

▶ **Theorem 4.4.** *For every $s, \epsilon \in [0,1]$ with $0 < s < 1$ and $\epsilon > 0$, there is an FO-interpretation Θ that maps instances of* 3XOR *to instances of* 3XOR *in such a way that, for every* 3XOR *instance I the following hold:*
1. *if I is satisfiable, then $\Theta(I)$ is $(1-\epsilon)$-satisfiable,*
2. *if I is not s-satisfiable, then $\Theta(I)$ is not $(1/2+\epsilon)$-satisfiable.*

Proof. First we define $\Theta(I)$ and then check that this definition is an FO-interpretation. In anticipation for the proof, let t be a large enough integer so that the following inequality holds:

$$(1 - (1 - (s+2)/3)^3)^{\alpha t/6} \leq 16\epsilon^3, \tag{2}$$

where α is the constant in Theorem 4.2. Such a t exists because $s < 1$ and $\epsilon > 0$. Apply the bipartite reduction to I to obtain the instance $I' = L(I)$ from Lemma 4.1. Observe that the domain size d of I' is $|A| + |B| = 6$. Next apply the parallel repetition reduction to I' with parameter t to obtain a new instance I''. Finally apply the long-code reduction to I'' with parameter ϵ to obtain the system I'''. The parameters were chosen in a way that the system I''' satisfies properties 1 and 2, through Theorem 4.3.

It remains to argue that I''' can be produced from I by an FO-interpretation. To define I' from I there is no difficulty at all: the FO-interpretation is even linear. To define I'' from I' we note that t is a constant, and that the weights $W(u,v)$ of I' are 0 or 1, so again there is no difficulty. In this case the FO-interpretation has dimension t, and it is n^t-bounded. To define I''' from I'' we note that the domain sizes p and q of the instance I'' are constants, indeed $p = 4^t$ and $q = 2^t$. This means that there are $|U| \cdot 2^{p-1}$ variables of type $u(a)$, and $|V| \cdot 2^{q-1}$ variables of type $v(b)$, and these are constant multiples of $|U|$ and $|V|$, respectively. Such domains are FO-definable by the method of finite expansions (see Section 2). Finally, since the weights $W(u,v)$ of I'' are still zeros or ones and both ϵ and q are constants, the multiplicities of the equations of I''' are also constants, and hence FO-definable. ◀

4.3 Second long-code reduction

The *second long-code reduction* takes a projection game instance I of LABEL COVER and a rational $\delta \in [0,1]$ and produces an instance $D(I, \delta)$ of 3SAT defined as follows. Before we define $D(I, \delta)$, let us define an intermediate instance $D'(I, \epsilon)$ of 3SAT that takes a different parameter $\epsilon \in [0,1]$. Let U, V, m, n, A, B, p, q, W, P, and $\pi_{u,v}(a)$ be as in the first long-code reduction. The set of variables of $D(I,\epsilon)$ is defined as in the first long-code reduction: a variable $u(a)$ for each $u \in U$ and each $a \in \mathbb{F}_2^{p-1}$, and a variable $v(b)$ for each $v \in V$ and each $b \in \mathbb{F}_2^{q-1}$. We also use the *folding* notation $F(z)$ and $S(z)$ from the first long-code reduction. Now the instance $D'(I,\epsilon)$ includes $W(u,v) \cdot M^q \cdot \epsilon^D \cdot (1-\epsilon)^{E-D} \cdot H$ copies of the clause $\{v(F(x))^{(S(x))}, u(F(y))^{(S(y))}, u(F(z))^{(S(z))}\}$ for each $(u,v) \in U \times V$, each $x \in \mathbb{F}_2^q$ and each $y, z \in \mathbb{F}_2^p$, where M is the denominator of $\epsilon = N/M$ reduced to lowest terms, E is the number of positions $i \in [p]$ with $x_{\pi(i)} = 1$ and D is the number of positions $i \in [p]$ with $x_{\pi(i)} = 1$ and $z_i \neq y_i$ for $\pi = \pi_{u,v}$ if $W(u,v) \neq 0$, while $H \in \{0,1\}$ is the indicator for the event that in each position $i \in [p]$ with $x_{\pi(i)} = 0$ we have $z_i \neq y_i$. Finally, to define the instance $D(I,\delta)$, set $t = \lceil \delta^{-1} \rceil$ and $\epsilon_1 = \delta$, and $\epsilon_{i+1} = \delta^{71} 2^{-35} \epsilon_i$ for $i = 1, \ldots, t-1$, and let the instance be $\bigcup_{i=1}^{t} D'(I,\epsilon_i)$.

▶ **Theorem 4.5** (Håstad 3-Query Disjunction Test [16]). *There exists $s_0 > 0$ such that for every $s \in [0,1]$ with $0 < s < s_0$ and every projection game instance I of* LABEL COVER *the following hold:*

1. *if I is satisfiable, then $C(I,\epsilon)$ is satisfiable,*
2. *if I is not s-satisfiable, then $C(I,\epsilon)$ is not $(7/8 + \log_2(1/s)^{-1/2})$-satisfiable.*

For the proof of Theorem 4.5, see Lemmas 6.12 and 6.13 in [16]. As in the first long-code reduction, some explanation is needed for seeing this.

Besides the notational differences that were already pointed out in the first long-code reduction, the second long-code reduction adds the following. First, the constants 71 and 35 in the definition of ϵ_{i+1} come from setting $c = 1/35$ in the definition of Test $\mathrm{F3S}^{\delta}(u)$ in [16]. According to Lemma 6.9 in [16], this is an acceptable setting of c. Second, the constant $s_0 > 0$ in Theorem 4.5 is meant to be chosen small enough so as to ensure that, for each s satisfying $s < s_0$, we have $2^{-64\delta^{-2}/25} < 2^{-d\delta^{-1}\log_2(\delta^{-1})}$ for $\delta = 8\log_2(1/s)^{-1/2}/5$, where d is the constant hidden in the asymptotic O-notation of Lemma 6.13 in [16]. Such an s_0 exists because $N\log_2(N) = o(N^2)$ as $N \to +\infty$. With this notation, Lemma 6.12 in [16] gives point 1, and Lemma 6.13 in [16] with $\delta = 8\log_2(1/s)^{-1/2}/5$ gives point 2 in Theorem 4.5.

By composing Lemma 4.1, Theorem 4.2, and Theorem 4.5 with the appropriate parameters we get the following:

▶ **Theorem 4.6.** *For every $s, \epsilon \in [0,1]$ with $0 < s < 1$ and $\epsilon > 0$, there is an FO-interpretation Θ that maps instances of* 3XOR *to instances of* 3SAT *in such a way that, for every* 3XOR *instance I the following hold:*

1. *if I is satisfiable, then $\Theta(I)$ is satisfiable,*
2. *if I is not s-satisfiable, then $\Theta(I)$ is not $(7/8+\epsilon)$-satisfiable.*

Proof. First we define $\Theta(I)$ and then check that this definition is an FO-interpretation. Let t be a large enough integer so that the following inequality holds:

$$(1 - (1 - (s+2)/3)^3)^{\alpha t/6} \leq \min\{2^{-1/\epsilon^2}, s_0\} \tag{3}$$

where α is the constant in Theorem 4.2 and $s_0 > 0$ is small enough as in Theorem 4.5. Such a t exists because $s < 1$ and $\epsilon > 0$ as well as $s_0 > 0$. Apply the bipartite reduction to I to obtain the instance $I' = L(I)$ from Lemma 4.1. Observe that the domain size d of I' is $|A| + |B| = 6$. Next apply the parallel repetition reduction to I' with parameter t to obtain a new instance I''. Finally apply the second long-code reduction to I'' to obtain the system I'''. The parameters were chosen so that the system I''' satisfies properties 1 and 2, through Theorem 4.5. As in the proof of Theorem 4.4 this reduction is FO-definable. ◀

4.4 Optimal gap inexpressibility

We are ready to state the main results of this section. Composing Theorem 3.7, Theorem 4.4, and Lemma 2.1 we get the following.

▶ **Theorem 4.7.** *For each $\epsilon > 0$, there is a $\delta > 0$ such that if $\mathcal{C}$ is the collection of* 3XOR *instances that are $(1-\epsilon)$-satisfiable and $\mathcal{D}$ is the collection of* 3XOR *instances that are* not *$(1/2+\epsilon)$-satisfiable then $\mathcal{C}$ and $\mathcal{D}$ are not C^k-separable for any $k = o(n^\delta)$.*

Composing Theorem 3.7, Theorem 4.6, and Lemma 2.1 we get the following.

▶ **Theorem 4.8.** *For each $\epsilon > 0$, there is a $\delta > 0$ such that if $\mathcal{C}$ is the collection of* 3SAT *instances that are satisfiable and $\mathcal{D}$ is the collection of* 3SAT *instances that are* not *$(7/8+\epsilon)$-satisfiable then $\mathcal{C}$ and $\mathcal{D}$ are not C^k-separable for any $k = o(n^\delta)$.*

A statement similar to Theorem 4.8 can be obtained from applying the standard reduction from 3XOR to 3SAT to Theorem 4.7 as in Theorem 3.8. However, this would *only* show that the class of 3SAT instances that are $(1-\epsilon)$-satisfiable is C^k-inseparable from the class of instances that are not $(7/8+\epsilon)$-satisfiable; note that Theorem 4.8 states the stronger claim that this is the case for the class of *fully satisfiable* instances, instead of the $(1-\epsilon)$-satisfiable ones. A natural question to ask is whether the $(1-\epsilon)$ in Theorem 4.7 could be improved to 1. This would, however, require different techniques since there is a polynomial-time algorithm that separates the satisfiable instances of 3XOR from the unsatisfiable ones.

On the other hand, $7/8+\epsilon$ bound in Theorem 4.8 and the $1/2+\epsilon$ bound in Theorem 4.7 are optimal. Every instance of 3SAT is 7/8-satisfiable, and every instance of 3XOR is 1/2-satisfiable. Thus, the algorithms that achieve these approximation ratios are trivial and expressible in FPC.

It is also worth comparing the statement of Theorem 3.8 to that of Theorem 4.8. While the latter has the stronger bound on the approximability ratio (7/8 rather than 15/16) the former gives the stronger lower bound on the counting width. One significance of the lower bounds on counting width is that they provide bounds on the number of levels of semidefinite programming hierarchies such as Lasserre hierarchy needed to solve a problem. Thus, it is known [13, 9] that if a constraint maximization problem can be solved using t levels of the Lasserre hierarchy, its counting width is at most $O(t)$. Thus, it is an immediate consequence of our results that approximation algorithms obtained through $o(n^\delta)$ levels of the Lasserre hierarchy cannot achieve an approximation ratio for 3SAT and 3XOR better than the trivial 7/8 and 1/2 respectively. These lower bounds on Lasserre relaxations are already known (see [25]) but our results provide a systematic explanation in terms of definability.

5 Vertex Cover

We investigate gap inexpressibility results for the vertex cover problem VC on graphs. Recall that a set $X \subseteq V$ of vertices in a graph $G = (V, E)$ is a *vertex cover* if every edge in E has at least one of its endpoints in X. If the graph comes with a weight function $w : V \to \mathbb{R}^+$, then the weight of X is the sum of the weights of the vertices in X. If the weights of the vertices are omitted in the specification of the graph, then all the vertices are assumed to have unit weight. The problem of finding the minimum weight vertex cover in a graph is a classic NP-complete problem.

In the following we write $\mathrm{VC}(G)$ for the weight of a minimum weight vertex cover, and $\mathrm{vc}(G) := \mathrm{VC}(G)/W_0$, where $W_0 := \sum_{v \in V} w(v)$. Analogously, we write $\mathrm{IS}(G)$ for the weight of a maximum weight independent set, and $\mathrm{isd}(G) := \mathrm{IS}(G)/W_0$. Clearly $\mathrm{vc}(G) = 1 - \mathrm{isd}(G)$ holds for all weighted graphs.

The standard reduction that proves the NP-completeness of the vertex cover problem (see, e.g. [22, Thm. 9.4]) takes an instance I of 3SAT with n variables and m clauses and gives a graph G with $3m$ vertices in which the minimum vertex cover has size exactly $2cm$, if cm is the maximum number of clauses in I that can be simultaneously satisfied. It is also easy to see that this reduction can be given as an FO-interpretation. This interpretation is linearly bounded and therefore it follows from Theorem 4.8 and Lemma 2.1 that for any $\epsilon > 0$, there is a $\delta > 0$ such that the collection of graphs G with $\mathrm{vc}(G) \leq 7/12 + \epsilon$ and the collection of graphs G with $\mathrm{vc}(G) \geq 2/3 - \epsilon$ cannote be separated in C^k for any $k = o(n^\delta)$. This has the consequence that no approximation algorithm for the vertex cover problem expressible in FPC can achieve an approximation ratio better than 8/7.

We can improve on this by considering instead the so-called FGLSS reduction from 3XOR to vertex-cover, which we describe next.

▶ **Theorem 5.1.** *There is a linearly-bounded first-order reduction G that takes an instance I of* 3XOR *with m equations to a graph $G(I)$ with $4m$ vertices so that if m^* is the maximum number of equations of I that can be simultaneously satisfied, then $\mathrm{vc}(G) = 4m - m^*$.*

Proof. For each equation $x + y + z = b$ in I, $G(I)$ has a 4-clique of vertices, each labelled with a distinct assignment of values to the three variables that make the equation true. In addition, we have an edge between any pair of vertices that are labelled by inconsistent assignments. It is easily seen that the largest independent set in $G(I)$ is obtained by taking an assignment g of values to the variables of I that satisfies m^* equations and, for each satisfied equation, selecting the vertex in its 4-clique that is the projection of g. This yields an independent set of size exactly m^* and the result follows. ◀

From this, and Theorem 3.7, we immediately get the following result.

▶ **Corollary 5.2.** *For any $\epsilon > 0$, if $\mathcal{C}$ is the collection of graphs G with $\mathrm{vc}(G) \leq 3/4$ and $\mathcal{D}$ is the collection of graphs G with $\mathrm{vc}(G) \geq 13/16 - \epsilon$ then $\mathcal{C}$ and $\mathcal{D}$ are not C^k-separable for any $k = o(n)$.*

Similarly, combining Theorem 5.1 and Theorem 4.7 yields the following corollary.

▶ **Corollary 5.3.** *For any $\epsilon > 0$, there is a $\delta > 0$ such that, if $\mathcal{C}$ is the collection of graphs G with $\mathrm{vc}(G) \leq 3/4 + \epsilon$ and $\mathcal{D}$ is the collection of graphs G with $\mathrm{vc}(G) \geq 7/8 - \epsilon$ then $\mathcal{C}$ and $\mathcal{D}$ are not C^k-separable for any $k = o(n^\delta)$.*

These two corollaries are incomparable. While the latter yields the stronger approximation ratio (7/6 rather than 13/12), the former gives the stronger lower bound on k.

Better lower bounds on the approximation ratio are known under the assumption that $\mathrm{P} \neq \mathrm{NP}$. One such lower bound was achieved by Dinur and Safra [14] who showed that, under this assumption, no polynomial-time algorithm for approximating vertex cover can achieve an approximation ratio better than 1.36. In the full version of this paper [8] we argue that this reduction is also an FO-interpretation, so we get the same inapproximability ratio for algorithms that are expressible in FPC, giving a strengthening of Corollary 5.3.

There are straightforward polynomial-time algorithms that yield a vertex cover in a graph with guaranteed approximation ratio 2. It is conjectured that no polynomial-time algorithm can achieve an approximation ratio of $2 - \epsilon$ for *any* $\epsilon > 0$. It would be interesting to prove a version of this conjecture for algorithms expressible in FPC, and without the assumption that $\mathrm{P} \neq \mathrm{NP}$. This could be established by a strengthened version of Corollary 5.3 with better ratios. We next show that we can at least do this for the special case of $k = 2$.

▶ **Theorem 5.4.** *For any $\epsilon > 0$, if $\mathcal{C}$ is the collection of graphs G with $\text{vc}(G) \leq 1/2$ and $\mathcal{D}$ is the collection of graphs G with $\text{vc}(G) \geq 1 - \epsilon$ then $\mathcal{C}$ and $\mathcal{D}$ are not C^2-separable.*

Proof. Let $(G_n)_{n\in\mathbb{N}}$ be a family of 3-regular expander graphs on n vertices, so that the largest independent set in G_n has size $o(n)$. For the existence of such graphs see [26, Chapter 4]. It follows that the smallest vertex cover in G_n has size $n - o(n)$. Hence, we can choose a value m such that G_{2m} has no vertex cover smaller than $2m(1 - \epsilon)$.

Let H_m be a 3-regular bipartite graph on two sets of m vertices. Now, each part of a bipartite graph is a vertex cover, so H_m has a vertex cover of size m. However, it is known that $G \equiv_{C^2} H$ holds for any pair G and H of d-regular graphs with the same number of vertices, for any d. Thus, $G_{2m} \equiv_{C^2} H_m$ and the result follows. ◀

Essentially, Theorem 5.4 tells us that no algorithm that is invariant under $\equiv_{C^2}$ can determine $\text{VC}(G)$ to an approximation better than 2, and Corollary 5.3 tells us that no algorithm that is invariant under $\equiv_{C^k}$ for constant or even slowly growing k can determine $\text{VC}(G)$ to an approximation better than 7/6. A legitimate question at this point is whether there is any algorithm that is invariant under $\equiv_{C^k}$, such as one expressible in FPC would be, that *does* achieve an approximation ratio of 2. The natural polynomial-time algorithms that give a vertex cover with size at most $2\text{VC}(G)$ are not expressible in FPC. Indeed, we cannot expect a formula of FPC to define an actual vertex cover in a graph G as this is not invariant under automorphisms of G. We can only ask for an estimate of the size, i.e. of $\text{VC}(G)$, and this we can get up to a factor of 2. For this, it turns out that $k = 2$ is enough, showing that the lower bound of Theorem 5.4 is tight:

▶ **Theorem 5.5.** *For any δ, if $\mathcal{C}$ is the collection of graphs G with $\text{vc}(G) \leq \delta$ and $\mathcal{D}$ is the collection of graphs G with $\text{vc}(G) > 2\delta$ then $\mathcal{C}$ and $\mathcal{D}$ are $\equiv_{C^2}$-separable.*

The proof of Theorem 5.5 can be found in Appendix D.

6 Conclusions

This paper introduces a new method for studying the hardness of approximability of optimization problems by showing that the approximation cannot be *defined* in a suitable logic such as FPC. This is done by showing that no class of bounded counting width can separate instances of the problem with a high optimum from those with a low one. This raises a number of new challenges in the application of this method. A clear demonstration of the power of this method would be to derive a lower bound stronger than one for which NP-hardness is known. For instance, can we improve, in the context of inexpressibility, on the 1.36-inapproximability for vertex cover from the NP-hardness result of Dinur and Safra [14]? In other words, can show that the class of graphs that have a vertex cover of density δ is not C^k-separable from the class of graphs that do not have a vertex cover of density $c\delta$, for some $\delta \in (0, 1)$ and some constant c greater than 1.36? If this were achieved for unbounded k, it would have major consequences in the study of semidefinite programming hierarchies of relaxations of vertex cover. And this applies, indeed, to any optimization problem for which the exact inapproximability factor is not known, including MAX CUT, sparsest cut, etc.

References

1 S. Abramsky, A. Dawar, and P. Wang. The pebbling comonad in finite model theory. In *Proc. of the 32nd IEEE Symp. on Logic in Computer Science (LICS).*, 2017.

2 M. Anderson, A. Dawar, and B. Holm. Solving linear programs without breaking abstractions. *J. ACM*, 62, 2015.

3 S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

4 S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

5 A. Atserias. On sufficient conditions for unsatisfiability of random formulas. *J. ACM*, 51:281–311, 2004.

6 A. Atserias, A. Bulatov, and A. Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410(18):1666–1683, 2009.

7 A. Atserias and V. Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74:323–334, 2008.

8 A. Atserias and A. Dawar. Definable inapproximability: New challenges for duplicator. *arXiv*, 2018. `arXiv:1806.11307`.

9 A. Atserias and J. Ochremiak. Definable ellipsoid method, sums-of-squares proofs, and the isomorphism problem. arxiv 1802.02388.

10 E. Ben-Sasson and A. Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48:149–169, 2001.

11 A. Dawar. The nature and power of fixed-point logic with counting. *ACM SIGLOG News*, pages 8–21, 2015.

12 A. Dawar and P. Wang. A definability dichotomy for finite valued CSPs. In *24th EACSL Annual Conference on Computer Science Logic, CSL 2015*, pages 60–77, 2015.

13 A. Dawar and P. Wang. Definability of semidefinite programming and lasserre lower bounds for CSPs. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*, 2017. `doi:10.1109/LICS.2017.8005108`.

14 I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162:439–485, 2005.

15 Martin Grohe. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, volume 47 of *Lecture Notes in Logic*. Cambridge University Press, 2017.

16 J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

17 Lauri Hella. Logical hierarchies in PTIME. *Information and Computation*, 129(1):1–19, 1996.

18 T. Holenstein. Parallel repetition: Simplifications and the no-signaling case. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 411–419, New York, NY, USA, 2007. ACM. `doi:10.1145/1250790.1250852`.

19 N. Immerman and E. S. Lander. Describing graphs: A first-order approach to graph canonization. In A. Selman, editor, *Complexity Theory Retrospective*. Springer-Verlag, 1990.

20 S. Khot, D. Minzer, and M. Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. Technical Report TR18-006, Electronic Colloquium on Computational Complexity (ECCC), 2018.

21 Phokion G Kolaitis and Moshe Y Vardi. On the expressive power of Datalog: Tools and a case study. In *Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 61–71. ACM, 1990.

22 Ch. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

23 Ch. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991.

24 R. Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

25 G. Schoenebeck. Linear level lasserre lower bounds for certain k-CSPs. In *Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 593–602, 2008.

26 S. Vadhan. *Pseudorandomness*, volume 7:1–3 of *Foundations and Trends in Theoretical Computer Science*. Now Foundations and Trends, December 2012.

27 V. V. Vazirani. *Approximation Algorithms*. Springer, 2003.

A Proof of Lemma 2.1

Proof of Lemma 2.1. Let $\mathbb{A} \in \mathcal{C}_n$ and $\mathbb{B} \in \mathcal{D}_n$ be two structures. Then, since $\Theta(\mathbb{A})$ and $\Theta(\mathbb{B})$ have size at most $p(n)$, there is a formula $\phi \in C^{k(p(n))}$ such that $\Theta(\mathbb{A}) \models \phi$ and $\Theta(\mathbb{B}) \not\models \phi$. We compose ϕ with the interpretation Θ to obtain ϕ'. That is to say, we replace every relation symbol by its defining formula, including replacing all occurrences of equality by ε, and we relativize all quantifiers to δ. Note that this involves replacing quantification over elements with quantification over tuples. It is well known that a counting quantifier over tuples $\exists^i \overline{x}$ can be replaced by a series of counting quantifiers over single elements without increasing the total number of variables. Then $\mathbb{A} \models \phi'$ and $\mathbb{B} \not\models \phi'$. It is also easy to check that ϕ' has at most $dk(p(n)) + t$ variables. The multiplicative factor d comes from the fact that every variable in ϕ is replaced by a d-tuple and the additive t accounts for any other variables that may appear in the formulas of Θ. ◀

B Proofs Omitted from Section 3.2

Proof of Lemma 3.1. For proving 1, let $h : \{x_1, \ldots, x_n\} \to \{0, 1\}$ be an assignment of values to the variables of I that satisfies at least cm of the m equations in I. Define the assignment g on the variables of $G(I)$ by $g(x^a) = g(x) + a$. For each equation e satisfied by h, all eight equations arising from e are satisfied by g and so g satisfies at least $8cm$ of the $8m$ equations in $G(I)$.

For proving 2, suppose g is an assignment of values in $\{0, 1\}$ to the variables x_i^a in $G(I)$. Let $h : \{x_1, \ldots, x_n\} \to \{0, 1\}$ be the assignment defined by $h(x_j) = g(x_j^0)$. We claim that if e_i is an equation $x_j + x_k + x_l = b$ in I that is not satisfied by h then at least four of the eight equations in $G(I)$ arising from e_i are falsified by g. To see this, consider two cases. First, suppose that $g(x_t^0) = g(x_t^1)$ for some $t \in \{j, k, l\}$. Without loss of generality, we assume $t = j$. Then consider the four *pairs* of equations

$$x_j^0 + x_j^{a_1} + x_k^{a_2} = b_i + a_1 + a_2,$$
$$x_j^1 + x_j^{a_1} + x_k^{a_2} = b_i + a_1 + a_2 + 1$$

obtained by taking the four possible values of a_1 and a_2. Since $g(x_j^0) = g(x_j^1)$, if one equation in a pair is satisfied by g the other is necessarily falsified. Thus, at least four equations are falsified. For the second case, suppose that for each $t \in \{j, k, l\}$ occurring in e_i we have $g(x_t^0) \neq g(x_t^1)$. But then, since we assume that h falsifies e_i, it follows that g falsifies $x_j^0 + x_k^0 + x_l^0 = b$ and hence it falsifies all eight equations arising from e_i. In either case, g falsifies at least four of the equations arising from e_i.

Now, suppose that g satisifes at least $(1/2 + s/2) \cdot 8m$ of the $8m$ equations in $G(I)$. We claim that h satisfies at least sm equations in I. Suppose for contradiction that h falsifies a proportion $\epsilon > 1 - s$ of the equations. By the above argument, then g falsifies at least $4\epsilon m$ of the equations in $G(I)$. But $4\epsilon m > (1/2 - s/2) \cdot 8m$ contradicting the assumption that g satisfies at least $(1/2 + s/2) \cdot 8m$ equations. ◀

Proof of Lemma 3.2. We describe a strategy for Duplicator in the bijective k-pebble game played on $G(I)$ and $G(I^0)$, given a strategy in the existential k-pebble game on I and $\Gamma = \Gamma_{3\text{XOR}}$.

Suppose we have a position in the existential k-pebble game on I and Γ with pebbles on $x_1, \ldots, x_{k'}$, for some $k' \leq k$ in I, and corresponding pebbles on $v_1, \ldots, v_{k'} \in \{0, 1\}$ in Γ. Suppose further that this is a winning position for Duplicator, i.e. she has a strategy to play forever from this position. Then, we claim that the position in the bijective game where the pebbles in $G(I)$ are on $x_1^{a_1}, \ldots, x_{k'}^{a_{k'}}$, for some $a_1, \ldots, a_{k'} \in \{0,1\}$ and the matching pebbles in $G(I^0)$ are on $x_1^{a_1+v_1}, \ldots, x_{k'}^{a_{k'}+v_{k'}}$ is a winning position in the bijective game on these two structures. To see this, note first that, if $x_r+x_s+x_t = b_i$ is an equation in I, for $1 \le r,s,t \le k'$, then by assumption that the position is winning in the existential game, $v_r + v_s + v_t = b_i$. Hence, $x_r^{a_r} + x_s^{a_s} + x_t^{a_t} = b_i$ is an equation in $G(I)$ if, and only if, $x_r^{a_r} + x_s^{a_s} + x_t^{a_t} = 0$ is an equation in $G(I^0)$ if, and only if, $x_r^{a_r+v_r} + x_s^{a_s+v_s} + x_t^{a_t+v_t} = v_r + v_s + v_t$ is an equation in $G(I^0)$, but this last equation is $x_r^{a_r+v_r} + x_s^{a_s+v_s} + x_t^{a_t+v_t} = b_i$. Thus, the map from $x_1^{a_1}, \ldots, x_{k'}^{a_{k'}}$ to $x_1^{a_1+v_1}, \ldots, x_{k'}^{a_{k'}+v_{k'}}$ is a partial isomorphism. To see that Duplicator can maintain the condition, suppose Spoiler moves the pebble on x_j^a. By assumption, Duplicator has a response in the existential game whenever Spoiler moves the pebble from x_j to x_l. This response defines a function f from the variables in x to $\{0,1\}$. We use this to define the bijection taking x_l^a to $x_l^{a+f(x_l)}$. This is a winning move in the bijective game. ◀

C Deriving Theorem 4.3 from [16]

The proof of Theorem 4.3 follows from Lemmas 5.1 and 5.2 in [16]. In order too see this, we need to explain how our notation matches the one in [16]. Besides the obvious and minor correspondance between multiplicative and additive notation for $\mathbb{F}_2$, with $-1 \leftrightarrow 1$ and $+1 \leftrightarrow 0$, there are three other noticeable differences between the statement of Theorem 4.3 and the statements of Lemmas 5.1 and 5.2 in [16].

The first difference is that Theorem 4.3 applies to arbitrary projection game instances of LABEL COVER, while the statements in [16] are phrased only for the special cases of the problem that result from applying parallel repetition to a suitable bipartite reduction applied to a 3SAT instance. We chose to formulate Theorem 4.3 in this more general and modular form because this is what the proofs of Lemmas 5.1 and 5.2 in [16] show, and also because this is how more recent expositions of these results are presented (see, e.g., [3]).

The second difference is that the conclusion of our statement is phrased in terms of the c-satisfiability of a 3XOR instance, while the statements of Lemmas 5.1 and 5.2 in [16] are phrased in terms of the acceptance rate of a probabilistic test that has the following form: given access to certain *tables* A_u and A_v, with $\mathbb{F}_2$ entries $\{A_u(x)\}_{x \in I}$ and $\{A_v(y)\}_{y \in J}$ for certain index sets I and J, respectively, choose a random 3-variables parity test on the $A_u(x)$ and $A_v(y)$ entries under a specially designed distribution, and check if it is satisfied. This difference is only notational and minor: our instance of XOR is built by viewing the $A_u(x)$ and $A_v(y)$ entries as variables $u(x)$ and $v(y)$, and assigning weight to each 3-variable parity equation on these variables proportionally to the probability that it is checked by the probabilistic test on the A_u and A_v tables. With this change, c-satisfiability of the instance translates into the probability of acceptance of the test being at least c, and vice-versa.

The third difference in the notation is that our variables $u(x)$ and $v(y)$, and the corresponding entries $A_u(x)$ and $A_v(y)$ of the tables A_u and A_v, are indexed by $\mathbb{F}_2^{p-1}$ and $\mathbb{F}_2^{q-1}$ instead of the more natural $\mathbb{F}_2^p$ and $\mathbb{F}_2^q$, respectively. This is due to the fact that we implement the operations of *folding over true* and *conditioning upon h* from [16] directly in our construction. In other words, our tables A_u and A_v are what [16] calls $A_{W,h,true}$ and $A_{U,true}$, respectively. Folding over true as in $A_{U,true}$ is achieved for A_v through the notation $S(z)$ and $F(z)$ defined above: we chose to partition $\mathbb{F}_2^p$ into 2^{p-1} pairs of the form $(z,0), (F((z,1)),1)$, as z ranges over $\mathbb{F}_2^{p-1}$, and view an arbitrary $A_v : \mathbb{F}_2^{p-1} \to \mathbb{F}_2$ as representing the function $A'_v : \mathbb{F}_2^p \to \mathbb{F}_2$ defined by $A'_v(z) = A_v(F(z)) + S(z)$ for every $z \in \mathbb{F}_2^p$. It is straightfoward to see that A'_v is folded over true, in the definition of [16], by construction.

Conditioning upon h as in $A_{W,h,true}$ for A_u is achieved through the same mechanism as folding over true with the additional observation that the operation of conditioning upon h is necessary only if the instance of LABEL COVER fails to satisfy the property that for every $(u, v) \in U \times V$ and every $a \in A$ there is at least one $b \in B$ that satisfies the predicate $P(u, v, a, b)$. When this is the case, one defines $h = h_{u,v} : A \to \{0, 1\}$ as the predicate indicating if a given a has at least one b that satisfies $P(u, v, a, b)$, and *conditions the table* A_u *upon* h. In our case we do not require this since the given instance of LABEL COVER is a projection game instance, and, in particular, for every a there is exactly one b, and hence at least one b, such that $P(u, v, a, b) = 1$; i.e., $h = h_{u,v}$ is the constant 1 predicate. It should be added that the reason why we can assume that I is a projection game instance is that our bipartite reduction is designed in such a way that the values a in A are partial assignments that always satisfy the corresponding constraints u in U. In constrast, in [16] the values are taken as arbitrary truth assignments to the variables of a collection of clauses, and not all such assignments satisfy all the clauses. Our exposition is again more modular and also matches more recent expositions of the results in [16] (again, see, e.g., [3]).

With this notational correspondance, it is now easy to see that Lemma 5.1 in [16] gives the first claim in Theorem 4.3, and Lemma 5.2 in [16] applied with $\delta = (s/\epsilon)^{1/2}/4$ gives the second claim in Theorem 4.3.

D Proof of Theorem 5.5

The proof of Theorem 5.5 proceeds through a series of lemmas.

▶ **Lemma D.1.** *If G is a d-regular graph on n vertices, for any $d \geq 1$, then* $\mathrm{VC}(G) \geq n/2$.

Proof. Let S be any set of vertices in G. Then the number of edges incident on vertices in S is at most $d|S|$. Since the number of edges in G is $dn/2$, if S is a vertex cover $d|S| \geq dn/2$ and so $|S| \geq n/2$. ◀

Let G be a graph and $C_1, \ldots, C_m$ be the partition of the vertices of G given by *vertex refinement*. So, there are constants δ_{ij} such that each $v \in C_i$ has exactly δ_{ij} neighbours in C_j. Since the graph is undirected, the number of edges from C_i to C_j is the same as in the other direction and so $\delta_{ij}|C_i| = \delta_{ji}|C_j|$, for all i and j. Also, $\delta_{ij} = 0$ if, and only if, $\delta_{ji} = 0$.

Let $X = \{i \mid \delta_{ii} = 0\}$ and $Y = \{i \mid \delta_{ii} > 0\}$. Consider the undirected graph X_G with vertices X and edges $\{(i, j) \mid \delta_{ij} > 0\}$. Consider the instance (X_G, w) of *weighted vertex cover* obtained by taking the graph X_G and giving each vertex i the weight $w(i) = |C_i|$. Let p_G denote the value of the minimum weighted vertex cover of this instance. Also, let $q_G = \sum_{i \in Y} |C_i|$. Finally, define $v_G = p_G + q_G$.

▶ **Lemma D.2.** *If $G \equiv_{C^2} H$ then $v_G = v_H$.*

Proof. The value v_G is determined entirely by the sizes of C_i in the vertex refinement of G and the corresponding values of δ_{ij}. Since $G \equiv_{C^2} H$, these values are the same for H. ◀

▶ **Lemma D.3.** $\mathrm{VC}(G) \leq v_G$.

Proof. Let $Z \subseteq X$ be a minimum-weight vertex cover in (X_G, w). Take the set $S \subseteq V(G)$ defined by $S = \bigcup_{i \in Y \cup Z} C_i$. Note that the sets Y and Z are disjoint, $\sum_{i \in Y} |C_i| = q_G$ by definition, and $\sum_{i \in Z} |C_i| = p_G$ by construction. So S has exactly v_G vertices. We claim that S is a vertex cover in G. Let e be any edge of G with endpoints in C_i and C_j. If either i or

j is in Y, then the corresponding endpoint of e is in S since $C_i \subseteq S$ for all $i \in Y$. If both i and j are not in Y then both are in X and $\delta_{ij} > 0$. Thus, since Z is a vertex cover for the graph X_G then one of i or j must be in Z and again at least one endpoint of e is in S. ◀

For the proof of the next lemma, we need the notion of a *fractional vertex cover* of a graph $G = (V, E)$. This is a function $f : V \to [0, 1]$ satisfying the condition that for every $(u, v) \in E$, $f(u) + f(v) \geq 1$. It is known that if f is a fractional vertex cover of G, then $\sum_{v \in V} f(v) \geq \text{VC}(G)/2$ (see [27, Thm. 14.2]). More generally, suppose we have an instance of *weighted vertex cover*, i.e. G along with a weight function $w : V \to \mathbb{N}$ where $\text{VC}(G, w)$ is defined as the value of the minimum weighted vertex cover. Then $\sum_{v \in V} f(v)w(v) \geq \text{VC}(G, w)/2$.

▶ **Lemma D.4.** $v_G \leq 2\text{VC}(G)$.

Proof. Let S be any vertex cover of G. Let $U_X = \bigcup_{i \in X} C_i$ and $U_Y = \bigcup_{i \in Y} C_i$ and note that these sets are disjoint. We claim that $|S \cap U_X| \geq p_G/2$ and $|S \cap U_Y| \geq q_G/2$, and therefore $|S| = |S \cap U_X| + |S \cap U_Y| \geq v_G/2$, establishing the result.

First, consider $S \cap U_Y$. Note that for any $i \in Y$, the subgraph of G induced by C_i is δ_{ii}-regular. Since $\delta_{ii} > 0$ by definition of Y, by Lemma D.1 we have $|S \cap C_i| \geq |C_i|/2$ and therefore $|S \cap U_Y| \geq q_G/2$.

Secondly, consider the function $f : X \to [0, 1]$ defined by $f(i) = |S \cap C_i|/|C_i|$. We claim that this is a fractional vertex cover of the graph X_G. To verify this, we need to check that $f(i)+f(j) \geq 1$ whenever $\delta_{ij} > 0$. There are $\delta_{ij}|C_i|$ edges between C_i and C_j. Each element of $S \cap C_i$ can cover at most δ_{ij} of these edges and similarly each element of $S \cap C_j$ covers at most δ_{ji} of them. Thus, since S is a vertex cover $|S \cap C_i|\delta_{ij} + |S \cap C_j|\delta_{ji} \geq \delta_{ij}|C_i|$. Substituting for δ_{ji} using the identity $\delta_{ij}|C_i| = \delta_{ji}|C_j|$ gives $|S \cap C_i|\delta_{ij} + |S \cap C_j|\delta_{ij}|C_i|/|C_j| \geq \delta_{ij}|C_i|$. Now dividing through by $\delta_{ij}|C_i|$ gives $f(i) + f(j) \geq 1$.

Thus, we have that the weighted vertex cover instance (X_g, w) admits the fractional solution f whose total weight is

$$\sum_{i \in X} f(i)|C_i| = \sum_{i \in X} |S \cap C_i| = |S \cap U_X|.$$

Since p_G is the value of the minimum weight vertex cover of (X_g, w), we have $|S \cap U_X| \geq p_G/2$, as was to be shown. ◀

Proof of Theorem 5.5. Suppose for contradiction that there is a $G \in \mathcal{C}$ and $H \in \mathcal{D}$ such that $G \equiv_{C^2} H$. Since G and H must have the same number of vertices, we have $2\text{VC}(G) < \text{VC}(H)$. But, by Lemma D.4 we have $v_G \leq 2\text{VC}(G)$, by Lemma D.3 we have $\text{VC}(H) \leq v_H$ and by Lemma D.2 we have $v_G = v_H$, giving a contradiction. ◀