

# Undefinability of Approximation of 2-To-2 Games

Anuj Dawar   

Department of Computer Science and Technology, University of Cambridge, UK

Bálint Molnár 

Department of Computer Science and Technology, University of Cambridge, UK

---

## Abstract

Recent work by Atserias and Dawar [6] and Tucker-Foltz [26] has established undefinability results in fixed-point logic with counting (FPC) corresponding to many classical complexity results from the hardness of approximation. In this line of work, NP-hardness results are turned into unconditional FPC undefinability results. We extend this work by showing the FPC undefinability of any constant factor approximation of weighted 2-to-2 games, based on the NP-hardness results of Khot, Minzer and Safra. Our result shows that the completely satisfiable 2-to-2 games are not FPC-separable from those that are not  $\epsilon$ -satisfiable, for arbitrarily small  $\epsilon$ . The perfect completeness of our inseparability is an improvement on the complexity result, as the NP-hardness of such a separation is still only conjectured. This perfect completeness enables us to show the FPC undefinability of other problems whose NP-hardness is conjectured. In particular, we are able to show that no FPC formula can separate the 3-colourable graphs from those that are not  $t$ -colourable, for any constant  $t$ .

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Finite Model Theory; Theory of computation  $\rightarrow$  Complexity theory and logic; Theory of computation  $\rightarrow$  Problems, reductions and completeness

**Keywords and phrases** Hardness of Approximation, Unique Games, Descriptive Complexity, Fixed-Point Logic with Counting

**Digital Object Identifier** 10.4230/LIPIcs.CSL.2025.16

**Funding** *Anuj Dawar*: Funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee: grant number EP/X028259/1.

## 1 Introduction

The study of the hardness of approximation of NP-optimization problems began in earnest with the PCP theorem in the 1990s. This theorem showed that for many problems (such as MAX 3SAT), where there are polynomial-time algorithms that can approximate the optimum solution within a constant factor, there is nonetheless a constant  $c$  such that no efficient algorithm can approximate the optimum value within a factor  $c$  unless  $P = NP$ . Indeed, Håstad [17] established tight bounds for MAX 3SAT: there is a trivial algorithm that achieves an  $\frac{8}{7}$  approximation, but none that achieves an  $\frac{8}{7} - \epsilon$  approximation for any  $\epsilon$ , unless  $P = NP$ . Such tight bounds are known for many NP-optimization problems, while for others there is a gap in the approximation ratio between the best known algorithm and the strongest known lower bound. An important problem in the latter category is the *minimum vertex cover* problem, where the best known polynomial-time algorithms yield an approximation ratio of 2, while the strongest proved lower bound is  $\sqrt{2}$ .

Perhaps the most important open question in the field of the hardness of approximation is the *unique games conjecture* of Khot. This states that for any  $\epsilon, \delta > 0$ , there is a set of labels  $\Sigma$  such that it is NP-hard to separate the  $(1 - \epsilon)$ -satisfiable instances of  $\Sigma$ -unique games (the precise definitions follow below) from those that are not even  $\delta$ -satisfiable. The strongest result obtained so far in this direction shows that there is a  $\Sigma$  for which it is NP-hard to separate the  $(\frac{1}{2} - \epsilon)$ -satisfiable instances from the  $\delta$ -unsatisfiable ones. This result is a consequence of the 2-to-2 theorem due to Khot, Minzer and Safra [20, 11, 21].



© Anuj Dawar and Bálint Molnár;

licensed under Creative Commons License CC-BY 4.0

33rd EACSL Annual Conference on Computer Science Logic (CSL 2025).

Editors: Jörg Endrullis and Sylvain Schmitz; Article No. 16; pp. 16:1–16:21

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The hardness of approximation has also been studied in recent years in the context of logical definability. In particular, Atserias and Dawar [6] showed that many of the NP-hardness results can be recast as *unconditional* undefinability results in *fixed-point logic with counting* (FPC). For example, there is an FPC formula which yields an  $\frac{8}{7}$  approximation of the value of a MAX 3SAT instance and there is *provably* no formula that yields an  $\frac{8}{7} - \epsilon$  approximation for any  $\epsilon > 0$ . Recall that FPC is a logic whose expressive power is contained within the complexity class P and which has been characterized as a natural *symmetric* fragment of that class [1]. Tucker-Foltz [26] established the first definability gap in FPC of unique games, by showing that no formula can distinguish the  $\frac{1}{2}$ -satisfiable instances from those that are not  $\frac{1}{3} + \delta$ -satisfiable and also showed that no constant factor approximation is FPC definable.

In the present paper, we consider the FPC definability of 2-to-2 games. The hardness of approximating the optimum value of such games was established through a series of results by Khot, Minzer and Safra [20, 25, 11]. At the core of their proof is a reduction from the problem MAX 3XOR of maximizing the number of satisfied clauses in a 3XOR instance. We show that the reductions used can be formulated, with some modification, as first-order definable reductions. As a consequence, we obtain the result that the completely satisfiable instances of 2-to-2 games cannot be separated by an FPC formula from those that are no more than  $\delta$ -satisfiable. This  $(1, \delta)$  separation is stronger (in terms of approximation ratios) than the known  $(1 - \epsilon, \delta)$  NP-hardness result due to the fact that the FPC undefinability of approximating MAX 3XOR was proved with *perfect completeness* in [6]. A corollary of our result is the FPC undefinability of a  $(\frac{1}{2}, \delta)$  separation for (a weighted version of) unique games. This improves, again in terms of the approximation ratios, the gap obtained by Tucker-Foltz, though it should be noted that the latter gap is for *unweighted* games.

A more striking consequence of our result is that no FPC sentence can separate the class of 3-colourable graphs from those that are not even  $t$ -colourable for any constant  $t \geq 3$ . The NP-hardness of such a separation has only been proved for  $t$  at most 5, though it is conjectured for larger values. Indeed, this is a central open problem in the rapidly growing study of *promise constraint satisfaction problems* (PCSP, see [7]).

The result on graph colouring should be compared with a recent result of Atserias and Dalmau [5] which shows that the promise graph colouring problem cannot be solved by a local consistency algorithm. In particular, this implies that for any constant  $t$  the 3-colourable graphs cannot be separated from those that are not  $t$ -colourable by a class (whose complement is) definable in Datalog. Since Datalog programs can be translated into sentences of FPC, our Theorem 5.3 can be seen as strengthening their result. It is worth examining this relationship more closely. It is known, from results of [4] and [8], that every class of bounded counting width (and therefore, in particular, any FPC definable class) that is the complement of a fixed-template constraint satisfaction problem (CSP) is already definable in Datalog. Hence, we can conclude from the result of Atserias and Dalmau that no FPC definable CSP separates the 3-colourable graphs from the non- $t$ -colourable ones. However, since it is conceivable that a separating class for these two CSPs is FPC definable but not itself a CSP, our result is still a strengthening. But we can say still more. It can be deduced from the proof in [5] that the 3-colourable graphs and the non- $t$ -colourable ones are not separable by any class definable in an existential positive infinitary logic ( $\exists^{+, \omega}$ ). Moreover, it is a consequence of a very recent proof due to Rossman (published in the present volume [24]) that every class of bounded counting width that is preserved under homomorphisms is definable in  $\exists^{+, \omega}$ . Thus, we can conclude from these results that no homomorphism-closed class of bounded counting width separates the 3-colourable graphs from the non- $t$ -colourable ones. Since Theorem 5.3 easily

applies to all classes of bounded counting width and not just the FPC-definable ones; and it is conceivable that a separating class is not necessarily closed under homomorphisms, our result subsumes even this strengthened version of that of Atserias and Dalmau.

In Section 2 we introduce the problems, notation and provide background definitions. An outline of the steps involved in the reduction of Khot, Minzer and Safra is given in Section 3. The proof that the reductions involved are definable as first-order interpretations is given in Section 4 and certain consequences derived in Section 5.

## 2 Preliminaries

### 2.1 Hardness of Approximation in Optimization

We are interested in NP-hard optimization problems. A standard example is the problem MAX 3SAT, where the aim is to find, given a formula in 3CNF, an assignment of values to its variables that maximizes the number of clauses satisfied. Formally, consider a function problem  $M$ , which associates with every possible input instance  $I$  a value  $M(I)$ . In our example, MAX 3SAT maps a formula  $\phi$  to the maximum number  $m$  of clauses of  $\phi$  that can be simultaneously satisfied. While, in practice, we might be interested in finding an assignment that achieves this maximum, for the purpose of proving hardness, it suffices to show that it is hard to compute the number  $m$ . When finding  $M(I)$  is hard, we may wish to approximate it, and we say that an algorithm computes a  $C$ -approximation (for a real number  $C > 1$ ) of  $M$  if it produces a number  $M'(I)$  with the guarantee that  $M'(I) \leq M(I) \leq C \cdot M'(I)$ .

For the sake of uniformity, we consider function problems that take values in  $[0, 1]$ . Thus, MAX 3SAT assigns to a 3CNF formula  $\phi$  the maximum fraction of the clauses of  $\phi$  that can be simultaneously satisfied. For MAX 3SAT, it is known that, unless  $P = NP$ , there is no polynomial-time algorithm that gives a  $C$ -approximation for any  $C < 8/7$ . Such hardness of approximation results are usually proved by means of a *hardness of separation*, which allows us to frame this in terms of the hardness of decision problems.

Formally, let  $A$  and  $B$  be two sets (i.e. decision problems) with  $A \cap B = \emptyset$ . We say that  $A$  and  $B$  are NP-hard to separate, if *every* set  $C$  with  $A \subseteq C \subseteq \overline{B}$  is NP-hard, where  $\overline{B}$  denotes the complement of  $B$ . For a function problem  $M$ , and a constant  $c \in [0, 1]$ , denote by  $c$ - $M$  the set  $\{I \mid M(I) \geq c\}$ . Then, for constants  $c$  and  $s$  with  $0 \leq s < c \leq 1$ , we say that the *gap problem*  $\text{Gap}M(c, s)$  is NP-hard if it is NP-hard to separate the sets  $c$ - $M$  and  $s$ - $\overline{M}$ . This implies, in particular, that unless  $P = NP$ , there is no polynomial-time algorithm giving a  $\frac{c}{s}$ -approximation of  $M$ . The value  $c$  in  $\text{Gap}M(c, s)$  is called the *completeness parameter* and  $s$  the *soundness parameter*.

The first hardness of approximation results come from the PCP theorem [2, 15, 3]: one of its direct consequences is the NP-hardness of  $\text{Gap}3\text{SAT}(1, \eta)$  for some constant  $\eta$  strictly less than 1. Håstad [17] obtained an optimum inapproximability result for MAX 3SAT. Namely, he showed that  $\text{Gap}3\text{SAT}(1, \frac{7}{8} + \epsilon)$  is NP-hard for arbitrarily small  $\epsilon$ . This is optimal since there is an easy  $\frac{8}{7}$ -approximation algorithm. Similarly, he also showed that  $\text{Gap}3\text{XOR}(1 - \epsilon, \frac{1}{2} + \epsilon)$  is NP-hard for arbitrarily small  $\epsilon$ . Again, this is optimal. Here, 3XOR is the problem where we are given a Boolean formula as a conjunction of clauses, each of which is the XOR of three literals and we aim to maximize the number of satisfied clauses. Note that the completeness parameter must be strictly less than 1, since the problem of determining whether such a formula is satisfiable or not is polynomial-time decidable. Thus 1-3XOR can be separated in polynomial time from  $\overline{(1 - \epsilon)\text{-3XOR}}$  for any  $\epsilon$ .

**Reductions**

A common way of deriving further hardness of approximation results is via gap-reductions: given function problems  $A$  and  $B$ , a polynomial-time computable function  $f$  taking instances of  $A$  to instances of  $B$  is a *reduction* from  $\text{Gap}A(c, s)$  to  $\text{Gap}B(c', s')$  if for all instances  $I$  of  $A$

- **Completeness:** if  $A(I) \geq c$ , then  $B(f(I)) \geq c'$ .
- **Soundness:** if  $A(I) \leq s$ , then  $B(f(I)) \leq s'$ .

It is easily seen that, if such a reduction exists and  $\text{Gap}A(c, s)$  is NP-hard, then so is  $\text{Gap}B(c', s')$ .

**2.2 Label Cover Games**

Versions of *label cover* problems are ubiquitous in the study of hardness of approximation (see [13]). A particularly important case are the *unique games* of Khot [18], defined below. To arrive at the definition, we first introduce some terminology. For positive integers  $d$  and  $e$ , a relation  $R \subseteq U \times V$  is said to be *d-to-e* if it relates each element of  $U$  to exactly  $d$  elements of  $V$  and each element of  $V$  to exactly  $e$  elements of  $U$ .

► **Definition 2.1** (*d-to-d games*). *A d-to-d game is a tuple  $(G, \Sigma, \Phi)$ , where  $G = (V, E)$  is a multi-graph<sup>1</sup>,  $\Sigma$  is a finite alphabet and  $\Phi : E \rightarrow \mathcal{P}(\Sigma^2)$  assigns to each edge  $e \in E$  a d-to-d binary relation.*

*A colouring  $\chi : V \rightarrow \Sigma$  satisfies an edge  $(u, v)$  if  $(\chi(u), \chi(v)) \in \Phi(u, v)$ .*

*The value of the game  $(G, \Sigma, \Phi)$  is the maximum over all colourings of the proportion of edges in  $E$  that are satisfied.*

In this paper, we are particularly interested in 2-to-2 games and 1-to-1 games, the latter also being known as *Unique Games*. We write  $\text{UG}_q$  for the function problem of determining the value of an instance of unique games with an alphabet of size  $q$ . We can then state Khot's unique games conjecture.

► **Conjecture 2.2** (Unique Games Conjecture (UGC) [18]). *For any  $\delta, \epsilon > 0$ , there exists a positive integer  $q$  so that  $\text{GapUG}_q(1 - \epsilon, \delta)$  is NP-Hard.*

The significance of the conjecture is that it has been shown that many optimal hardness of approximation results follow from it, including Max Cut and Vertex Cover [19, 23, 18].

The best known hardness result for unique games, towards proving Conjecture 2.2 is that  $\text{GapUG}_q(\frac{1}{2} - \epsilon, \delta)$  is NP-Hard for arbitrarily small  $\delta$  and  $\epsilon$ . This is obtained as a consequence of the hardness of 2-to-2 games established by Khot, Minzer and Safra, which we return to in Section 3.

► **Theorem 2.3** (Khot-Minzer-Safra). *For any  $\delta, \epsilon > 0$ , there exists a positive integer  $q$  so that  $\text{Gap2to2}_q(1 - \epsilon, \delta)$  is NP-Hard.*

It is conjectured that Theorem 2.3 can be strengthened to make the completeness parameter 1, but this remains unproved.

In this paper, we are particularly concerned with *weighted* 2-to-2 and 1-to-1 games, attaching a weight to each constraint.

---

<sup>1</sup> That is to say, there may be multiple edges between the same pair of vertices. In the sequel we refer simply to graphs to mean multi-graphs.

► **Definition 2.4** (Weighted  $d$ -to- $d$  games). A weighted  $d$ -to- $d$  game is a tuple  $(G, \Sigma, \Phi, w)$ , where  $(G, \Sigma, \Phi)$  is a  $d$ -to- $d$  game and  $w : E(G) \rightarrow \mathbb{R}^+$  is a function assigning a positive real weight to each constraint.

Let  $\text{tot} = \sum_{e \in E(G)} w(e)$  be the total weight. The value of the game  $(G, \Sigma, \Phi, w)$  is the maximum over all colourings  $\chi : V \rightarrow \Sigma$  of the fraction  $\sum_{e \in S_\chi} w(e) / \text{tot}$ , where  $S_\chi$  denotes the set of edges  $e = (u, v)$  for which  $(\chi(u), \chi(v)) \in \Phi(e)$ .

We write  $\mathcal{WG}_{2:2;q}$  to denote the class of weighted 2-to-2 games with  $q$  labels and  $\text{WEIGHT2TO2}_q$  to denote the function taking such a game to its value. Similarly, we write  $\text{UG}_q$  and  $\text{WEIGHTUG}_q$  for the functions giving the values of unique games and weighted unique games with  $q$  labels respectively.

### 2.3 Undefinability of Approximation

We assume the reader is familiar with first-order logic and the basics of finite model theory. A good introduction is to be found in [14]. Our structures are finite structures in a finite relational vocabulary. Our main inexpressibility results are stated for *fixed-point logic with counting* (FPC). We do not need a formal definition here but note that every property definable in FPC is decidable in polynomial-time and indeed FPC can be understood as a complexity class defined by *symmetric polynomial-time* computation. For full definitions, refer to [10] and references therein.

The two properties of FPC that we do need are that (1) every class of structures definable in FPC has *bounded counting width*; and (2) that the class of properties definable in FPC is closed under *first-order interpretations*. We elaborate on these below.

For a function problem  $M$ , and real numbers  $c$  and  $s$  with  $0 \leq s < c \leq 1$ , we say that  $\text{Gap}M(c, s)$  is undefinable in FPC if there is no FPC definable class of structures that separates the sets  $c\text{-}M$  and  $\overline{s\text{-}M}$ . Atserias and Dawar [6] initiated a study of the FPC undefinability of approximations, showing that many of the NP-hardness results for gap problems can be reproduced as *unconditional* undefinability results in FPC. In particular  $\text{Gap3SAT}(1, \frac{7}{8} + \epsilon)$  is not FPC definable. More significantly, they established the following

► **Theorem 2.5** (Atserias-Dawar [6]).  $\text{Gap3XOR}(1, \frac{1}{2} + \epsilon)$  is not FPC definable.

Note the completeness parameter of 1 in the statement, which contrasts with  $1 - \epsilon$  in the case of Theorem 2.3. Perfect completeness cannot be established in the case of NP-hardness because satisfiability of XOR formulas is decidable in polynomial-time. However, it is not definable in FPC and this allows the stronger result in the context of undefinability. This is crucial to the application we make of Theorem 2.5 in Section 5.3

Following up on this work, Tucker-Foltz [26] studied the undefinability of gaps in unique games. In particular, he established the inapproximability of unique games in FPC by any constant factor and the FPC-undefinability of  $\text{GapUG}_q(\frac{1}{2}, \frac{1}{3} + \delta)$  for a suitable value of  $q$ .

#### Counting Width

For relational structures  $\mathbb{A}$  and  $\mathbb{B}$  in the same vocabulary, and a positive integer  $k$ ,  $\mathbb{A} \equiv^k \mathbb{B}$  denotes that the two structures cannot be distinguished by any sentence of first-order logic with counting using no more than  $k$  distinct variables. For a class  $\mathcal{C}$  of structures, the *counting width* of  $\mathcal{C}$  is the function  $\nu : \mathbb{N} \rightarrow \mathbb{N}$  such that for any  $n$ ,  $\nu(n)$  is the least  $k$  such that  $\mathcal{C}$ , restricted to structures with at most  $n$  elements is a union of  $\equiv^k$ -equivalence classes. Any class that is definable by a sentence of FPC has counting width bounded by a constant. Almost all results showing that a class is not definable in FPC proceed by showing that it, in fact, does not have bounded counting width.

### Interpretations

A *first-order interpretation* of a relational vocabulary  $\tau$  in a vocabulary  $\sigma$  is a sequence of  $\sigma$ -formulas in first-order logic, which can be seen as mapping  $\sigma$ -structures to  $\tau$ -structures. There are many variations of the precise definition in the literature. We use the version defined in [6] and refer the reader to that for the formal definition. Given a function problem  $A$  whose instances are  $\sigma$ -structures and a function problem  $B$  whose instances are  $\tau$ -structures, an interpretation  $\Theta$  of  $\tau$  in  $\sigma$  is a  $\text{Gap}A(c, s)$  to  $\text{Gap}B(c', s')$  reduction if  $A(\mathbb{A}) \geq c$  implies  $B(\Theta(\mathbb{A})) \geq c'$  and  $A(\mathbb{A}) \leq s$ , then  $B(\Theta(\mathbb{A})) \leq s'$ . Definability in FPC and the property of having bounded counting width are both closed under first-order reductions. That is to say, if  $\text{Gap}B(c', s')$  is FPC-definable and there is a first-order reduction of  $\text{Gap}A(c, s)$  to  $\text{Gap}B(c', s')$ , then  $\text{Gap}A(c, s)$  is FPC-definable as well.

## 3 The Reduction

The proof of Theorem 2.3 was completed in 2018 and remains to this day the most significant advance towards establishing the Unique Games Conjecture since the latter was formulated by Khot in [18]. The proof proceeds by a reduction from  $\text{Gap}3\text{XOR}(1 - \epsilon, \frac{1}{2} + \delta)$  and was presented in a series of papers [20, 25, 11]. The main difficulty lies in proving the combinatorial conditions that the soundness analysis relies on. The full reduction and proof of correctness can be found in [22, Chapter 3].

Our aim in the present paper is to show that the reduction constructed has two crucial properties. First, it preserves perfect completeness and thus can be seen as a reduction from  $\text{Gap}3\text{XOR}(1, \frac{1}{2} + \delta)$ . Secondly, with small modifications which do not affect the soundness or completeness analysis, it can be described as a first-order interpretation. Together these establish the main theorem.

► **Theorem 3.1.** *For every  $\delta > 0$ , there exists  $q \in \mathbb{N}^+$  for which  $\text{Gap}WEIGHT2TO2_q(1, \delta)$  is not FPC definable.*

In proving this, we do not need to reprise the difficult soundness analysis carried out by Khot et al. Rather we study the actual construction involved in the reduction. For this purpose, we describe the reduction in some detail in this section, and take up the two issues of perfect completeness and first-order definability in the next.

### 3.1 Regular 3XOR

An instance of 3XOR can be seen as a system of linear equations over the field  $\mathbb{F}_2$  with exactly three variables appearing in each equation. We say that such an instance is *d-regular* if every variable appears in exactly  $d$  equations and no two equations share more than one variable. It is known that the NP-hardness of  $\text{Gap}3\text{XOR}(1 - \epsilon, \frac{1}{2} + \delta)$  holds even when restricted to *d-regular* instances for some fixed value of  $d$  (indeed, taking  $d = 5$  suffices, see [22, Theorem 3.3.1]). In Section 4.3 we show that this is also true of the undefinability in FPC of  $\text{Gap}3\text{XOR}(1, \frac{1}{2} + \delta)$ . From now on, we restrict attention to *d-regular* instances for a suitable fixed value of  $d$ , and we call the resulting function problem  $\text{Gap}REGULAR3\text{XOR}$ .

### 3.2 Reducing to Transitive Games

In the first step of the reduction, we reduce regular 3XOR instances to label cover games with a mixture of 2-to-2 and 1-to-1 constraints, with an additional transitivity requirement. We formally define these below.

► **Definition 3.2** (Transitive 2-to-2 games). *A transitive 2-to-2 game is a tuple  $(G, \Sigma, \Phi)$  where  $G = (V, E)$  is a graph,  $\Sigma$  is a finite alphabet and  $\Phi : E \rightarrow \mathcal{P}(\Sigma^2)$  assigns to each edge  $e$  either a 2-to-2 or a 1-to-1 relation and whenever  $\Phi(u, v)$  is 1-to-1, then for any edge  $(v, w)$ ,  $\Phi(u, w)$  is the composition of  $\Phi(u, v)$  and  $\Phi(v, w)$ .*

Note that the condition on composition only applies when  $\Phi(u, v)$  is 1-to-1, but  $\Phi(v, w)$  may be 1-to-1 or 2-to-2, and this determines whether  $\Phi(u, w)$  is 1-to-1 or 2-to-2.

Now, fix an instance  $I$  of GAPREGULAR3XOR, with  $X$  being the set of variables that appear in  $I$  and  $E$  the set of equations. Thus, each equation  $e \in E$  is of the form  $x + y + z = b$  for some  $b \in \mathbb{F}_2$ . We refer to  $x, y$  and  $z$  as the variables occurring in  $e$  and  $b$  as the *right-hand side* of  $e$ .

Fix a positive integer  $k$  and let  $\mathcal{U} \subseteq E^k$  be the set of  $k$ -tuples  $U$  of equations, satisfying the following properties:

- no variable occurs in more than one equation of  $U$ ; and
- if variables  $x$  and  $y$  appear in distinct equations of  $U$ , there is no equation in  $E$  (even outside  $U$ ) in which both  $x$  and  $y$  occur.

For  $U = (e_1, \dots, e_k) \in \mathcal{U}$ , let  $X_U$  denote the set of variables occurring in equations in  $U$  and for  $i \in \{1, \dots, k\}$  let  $v_i \in \mathbb{F}_2^X$  denote the vector which has 1s in the three coordinates corresponding to the variables occurring in  $e_i$  and 0s everywhere else. We define the *space of side-conditions* corresponding to  $U$  to be  $H_U = \text{Span}(v_1, \dots, v_k)$ . We say that a linear function  $f : \mathbb{F}_2^X \rightarrow \mathbb{F}_2$  satisfies the equations in  $U$  if  $f(v_i) = b_i$  for all  $i$ , where  $b_i$  is the right-hand side of  $e_i$ .

Now, fix a parameter  $l$  with  $l \leq |X|$ , and we define  $\mathcal{L}_U$  to be the collection of  $l$ -dimensional subspaces of  $\mathbb{F}_2^X$  which are linearly independent of  $H_U$ . That is

$$\mathcal{L}_U = \{L \subseteq \mathbb{F}_2^X \mid \dim(L) = l, L \cap H_U = \{\mathbf{0}\}\}.$$

The trivial intersection ensures that for any subspace  $L \in \mathcal{L}_U$ , any linear function  $f : L \rightarrow \mathbb{F}_2$  can be uniquely extended to one on  $L + H_U$  so that  $f(v_i) = b_i$  for all  $i$ . Therefore, the number of linear functions on  $L + H_U$  satisfying the equations in  $U$  is exactly  $2^l$ .

We can now define the reduction  $\Theta$  that takes the instance  $I$  to a 2-to-2 transitive game  $\Theta(I)$ . The reduction depends on the choice of parameters  $k$  and  $l$ . We omit the details on how to select the right parameters.

**Vertices.** The vertices of  $\Theta(I)$  are pairs  $(U, L)$ , where  $U \in \mathcal{U}$  and  $L \in \mathcal{L}_U$ .

**Alphabet.** The alphabet is a set of labels of size  $2^l$ . As noted above, for each vertex  $(U, L)$ , there are exactly  $2^l$  linear functions on  $L + H_U$  satisfying the equations in  $U$ . We fix, for each  $(U, L)$ , a bijection between the alphabet and this set of linear functions. Henceforth, we simply treat the functions themselves as labels.

**Constraints.** Given a pair of vertices  $u = (U, L)$  and  $v = (U', L')$ , the constraint  $\Phi(u, v)$  is a 1-to-1 relation if

$$\dim(L + H_U + H_{U'}) = \dim(L' + H_U + H_{U'}) = \dim(L + L' + H_U + H_{U'})$$

and a 2-to-2 relation if

$$\dim(L + H_U + H_{U'}) = \dim(L' + H_U + H_{U'}) = \dim(L + L' + H_U + H_{U'}) - 1.$$

<sup>2</sup> Here the sum is to be understood as vector space sum, i.e.  $L + H_U$  is the space spanned by the union of  $L$  and  $H_U$ .

To define the relation, note that any function  $f : L + H_U \rightarrow \mathbb{F}_2$  has a unique extension to  $L + H_U + H_{U'}$  (by the conditions in the definition of  $\mathcal{U}$ ). Then, we relate  $f$  to  $f' : L' + H_{U'} \rightarrow \mathbb{F}_2$  if, and only if,  $f$  and  $f'$  agree on the shared space  $(L + H_U + H_{U'}) \cap (L' + H_U + H_{U'})$ .

It is the case for any pair, that  $\dim(L + H_U + H_{U'}) = \dim(L' + H_U + H_{U'})$  [20, Lemma 4.3]. Let us call this dimension  $D$ . By [20, Lemma 4.4], any linear function  $f : L + H_U \rightarrow \mathbb{F}_2$  satisfying the equations of  $U$  has a unique extension to  $(L + H_U + H_{U'})$  that also satisfies the equations of  $U'$ . Then, it is easily seen that if  $\dim(L + L' + H_U + H_{U'}) = D$ , then  $f$  has exactly one label of  $(U', L')$  that it is consistent with, and if  $\dim(L + L' + H_U + H_{U'}) = D + 1$ , there are exactly two such functions, thanks to the “free dimension”. Hence, the constraints are 1-to-1 or 2-to-2 as required. The transitivity property of these constraints is established in [20, Appendix A].

### 3.3 The final (weighted) 2-to-2 game

The final step of the reduction is to transform the transitive game constructed in Section 3.2 into a *weighted* 2-to-2 game, getting rid of the 1-to-1 constraints. This weighted game is defined as follows.

Recall the transitive 2-to-2 game  $\Theta(I)$  constructed in Section 3.2. The transitivity condition guarantees that the vertices of  $\Theta(I)$  can be partitioned into cliques  $C_1, \dots, C_m$  so that edges in each clique are associated with 1-to-1 constraints. Moreover, these constraints are consistent in the sense that any colouring of a vertex  $V$  in a clique  $C$  can be extended in a unique way to a colouring of all vertices in  $C$  so that all edge constraints in  $C$  are satisfied. Also, by the transitivity condition, for distinct cliques  $C_i$  and  $C_j$ , either all pairs  $(u, v) \in C_i \times C_j$  are connected by 2-to-2 constraints or none are. Furthermore, these 2-to-2 constraints are consistent in the sense that given a clique-consistent colouring for  $C_i$  and  $C_j$ , either all or none of these 2-to-2 constraints are satisfied.

The final (weighted) 2-to-2 instance  $I_{2:2}^w$  we construct from  $\Theta(I)$  has as vertices the vertices of  $\Theta(I)$  and as edges all edges  $(u, v)$  of  $\Theta(I)$  where  $u$  and  $v$  are in distinct cliques. For each such edge, with  $u \in C_i$  and  $v \in C_j$ , we associate the constraint  $\Phi(u, v)$  which is as in  $\Theta(I)$ . The weight  $w(u, v)$  is the probability assigned to  $(u, v)$  by the following sampling process:

- Choose  $U \in \mathcal{U}$ , uniformly at random.
- Choose a random pair  $L, L'$  so that  $(U, L)$  and  $(U, L')$  are connected by a 2-to-2 edge. Let  $C_i$  be the clique containing  $(U, L)$  and  $C_j$  be the clique containing  $(U', L')$
- Choose uniformly at random a pair of vertices  $(u, v) \in C_i \times C_j$ .

### 3.4 Irregular soundness case

For the result in Section 5.3, we need the FPC-undefinability of a different gap problem based on 2-to-2 games. Specifically, we define the value of a game to be, not the fraction of constraints that can be satisfied, but the fraction of the vertices formed by the largest set  $X$  so that all constraints between nodes in  $X$  are satisfied. Moreover, we relax the notion of colouring to allow vertices to be coloured by multiple colours.

► **Definition 3.3.** For a 2-to-2 game  $((V, E), \Sigma, \Phi)$ , a colouring  $c : V \rightarrow \binom{\Sigma}{j}$  satisfies a set  $X \subseteq V$  if  $\forall (u, v) \in E \cap X^2. \exists a \in c(u), b \in c(v). (a, b) \in \Phi(u, v)$ .

That is to say, a  $j$ -colouring, i.e., one that assigns a set of  $j$  colours to each vertex satisfies a set  $X$  if each constraint between vertices in  $X$  is satisfied by some choice among the colours assigned to the vertices.



► **Definition 3.4** (Irregular Values). For constants  $j$  and  $q$  define the function  $\text{IRREG2TO2}_{j,q}$  to take a 2-to-2 game  $((V, E), \Sigma, \Phi)$  to the fraction  $|X|/|V|$  where  $X$  is the largest subset of  $V$  that is satisfied by some  $j$ -colouring  $c : V \rightarrow \binom{\Sigma}{j}$ .

We can now state the theorem below, which is a consequence of Theorem 3.1.

► **Theorem 3.5** (Definable 2-to-2 Games Theorem with irregular soundness). For every  $\delta$  with  $0 < \delta < 1$  and  $j \in \mathbb{N}^+$ , there exists  $q \in \mathbb{N}^+$  so that  $\text{GapIRREG2TO2}_{j,q}(1, \delta)$  is not FPC definable.

It is not hard to see that this is a consequence of Theorem 3.1, and the corresponding claim for NP-hardness appears in e.g. [20]. For completeness, we give a short proof.

► **Lemma 3.6.** For a weighted 2-to-2 game  $I = ((V, E), \Sigma, \Phi, w)$  with  $q = |\Sigma|$ , if  $\text{IRREG2TO2}_{j,q}((V, E), \Sigma, \Phi) = \delta$ , then  $\text{WEIGHT2TO2}(I) = \Omega(\frac{\delta^2}{j^2})$ .

**Proof.** Let  $c$  be a  $j$ -colouring of  $V$  that satisfies a set  $X$  with  $|X|/|V| \geq \delta$ . By [22, Remark 3.4.9], there is a  $\Omega(\delta^2)$  (weighted) fraction of the edges  $E$  which are satisfied by  $c$ , in the sense that for each such edge  $(u, v)$  there are colours  $a$  and  $b$  in  $c(u)$  and  $c(v)$  respectively such that  $(a, b) \in \Phi(u, v)$ . We now construct a standard colouring by a random process. That is, for each vertex  $v \in V$ , independently choose a colour  $\chi(v)$  from  $c(v)$  uniformly at random. For an edge  $(u, v)$ , let  $\Xi(u, v)$  be the indicator variable indicating whether  $(\chi(u), \chi(v)) \in \Phi(u, v)$  and let  $\Xi$  be the overall value of the colouring  $\chi$ . If  $(u, v) \in X^2$ , the probability that  $\chi$  satisfies the constraint  $\Phi(u, v)$  is at least  $\frac{1}{j^2}$ , as by definition, among the  $j^2$  pairs in  $c(u) \times c(v)$ , at least one satisfies the constraint. Then

$$\begin{aligned} \mathbb{E}[\Xi] &= \mathbb{E}\left[\sum_{(u,v) \in E} w(u,v)\Xi(u,v)\right] = \sum_{(u,v) \in E} w(u,v)\mathbb{E}[\Xi(u,v)] \\ &\geq \sum_{(u,v) \in E \cap X^2} w(u,v)\mathbb{E}[\Xi(u,v)] \geq \sum_{(u,v) \in E \cap X^2} w(u,v)\frac{1}{j^2} \geq \Omega(\delta^2)\frac{1}{j^2} \end{aligned}$$

Thus, there is a colouring that satisfies at least  $\Omega(\frac{\delta^2}{j^2})$  (weighted) fraction of the constraints. ◀

From Lemma 3.6, we can conclude Theorem 3.5. For any fixed  $\delta$  and  $j$ , the proof of Theorem 3.1 gives us a  $q$  and an FO reduction that takes satisfiable 3XOR instances to satisfiable 2-to-2 games and instances that are at most  $\eta$ -satisfiable to 2-to-2 games with value at most  $\Omega(\frac{\delta^2}{j^2})$ . Then, by Lemma 3.6, this same reduction also maps at most  $\eta$ -satisfiable 3XOR instances to 2-to-2 games  $I$  for which  $\text{IRREG2TO2}_{j,q}(I) < \delta$ .

### 3.5 2 ↔ 2 games

The definition of 2-to-2 games, Definition 2.4 only requires each constraint  $\Phi(u, v)$  to be a 2-to-2 relation, meaning that each element on the left is related to exactly two elements on the right and vice versa. However, the reductions yield games of a more restricted kind and this will be useful in Section 5.3. Say that a binary relation  $R \subseteq A \times B$  is 2 ↔ 2 if it is the disjoint union of bipartite graphs  $K_{2,2}$ . That is to say  $A$  and  $B$  can be each partitioned into sets  $A = \bigcup_i A_i$  and  $B = \bigcup_i B_i$  so that each  $A_i$  and  $B_i$  has exactly two elements and  $R = \bigcup_i A_i \times B_i$ .

## 16:10 Undefinability of Approximation of 2-To-2 Games

We claim that the reductions in the proof of Theorem 3.1 yield games in which all constraint relations are  $2 \leftrightarrow 2$ . Specifically, given linear functions  $f \neq f' : L + H_U \rightarrow \mathbb{F}_2$  so that their unique extension to the domain  $L + H_U + H_{U'}$  only differ in their “free dimension”, i.e. they agree in values on  $(L + H_U + H_{U'}) \cap (L' + H_U + H_{U'})$ ,  $f$  and  $f'$  are related to the same two linear functions on  $L' + H_{U'}$  (uniquely extensible to  $L' + H_U + H_{U'}$ ) in  $\Phi((U, L), (U', L'))$ . Thus, the constraint relations constructed are  $2 \leftrightarrow 2$ .

### 4 Definability

The aim in this section is to show that the reduction outlined in Section 3 can, with minor modifications, be implemented as a first-order interpretation, preserving perfect completeness. Thus, it gives a first-order definable reduction from  $\text{Gap3XOR}(1, \frac{1}{2} + \delta)$  to  $\text{GapWEIGHT2TO2}_q(1, \delta')$  for a suitable choice of parameters. This establishes Theorem 3.1.

#### 4.1 Perfect completeness

To show that the reduction from Section 3 preserves perfect completeness, it suffices to verify that instances of 3XOR that are satisfiable (i.e. have value 1) are mapped by the reduction to instances of  $\mathcal{WG}_{2:2}$  which also have value 1.

Assume  $I$  is an 3XOR instance on a set of variables  $X$  that is satisfiable, and let  $s : X \rightarrow \mathbb{F}_2$  be an assignment of values to the variables that satisfies it. Let  $I_{2:2}^w$  denote the weighted 2-to-2 game that  $I$  maps to under the reduction. Then, for each vertex  $(U, L)$  of  $I_{2:2}^w$  the restriction of  $s$  to  $L + H_U$  is a valid label since all equations are satisfied, and it is easily seen that this labelling satisfies all constraints.

#### 4.2 Vocabularies

An instance of 3XOR is defined as a structure over the vocabulary  $\tau_{\text{3XOR}} = \langle \text{Eq}_0, \text{Eq}_1 \rangle$  with two ternary relations. We think of the universe of a  $\tau_{\text{3XOR}}$ -structure  $\mathbb{A}$  as a set of variables. For  $b \in \{0, 1\}$ , a triple  $(x, y, z) \in \text{Eq}_b$  is understood as representing the equation  $x + y + z = b$ , where addition is modulo 2.

For each positive integer  $q$ , we define a vocabulary  $\tau_{(\text{T}) 2\text{-to-}2_q}$  such that structures in this vocabulary represent instances of transitive 2-to-2 games over a label alphabet of size  $q$ . Let  $S_q$  denote the collection of permutations of  $[q] = \{1, \dots, q\}$ . Note that there is a natural bijective correspondence between  $S_q$  and the 1-to-1 relations on  $[q]$ . Now, let  $S_q^{\#2}$  denote the set of pairs of permutations  $(\pi_1, \pi_2) \in S_q \times S_q$  such that for all  $i \in [q]$ ,  $\pi_1(i) \neq \pi_2(i)$ . Then, it is easily seen that each 2-to-2 relation on  $[q]$  can be seen as the union of such a pair of permutations. Our vocabulary  $\tau_{(\text{T}) 2\text{-to-}2_q}$  contains a binary relation for each element of  $S_q$  and one for each element of  $S_q^{\#2}$ :

$$\tau_{(\text{T}) 2\text{-to-}2_q} = \langle (C_\pi)_{\pi \in S_q}, (C_{\pi_1, \pi_2})_{(\pi_1, \pi_2) \in S_q^{\#2}} \rangle.$$

We write  $\mathcal{C}_1$  for the collection of relation symbols  $(C_\pi)_{\pi \in S_q}$  and  $\mathcal{C}_2$  for the collection of relation symbols  $(C_{\pi_1, \pi_2})_{(\pi_1, \pi_2) \in S_q^{\#2}}$ . Note that the vocabulary itself does not enforce the transitivity property, only a subset of the structures with this vocabulary are transitive 2-to-2 games.

For *weighted* 2-to-2 games, we construct a vocabulary that allows us to code instances with positive integer weights. This is more limiting than allowing rational weights, but as we show below in Section 4.6, it suffices for our purpose. Specifically,

$$\tau_{(\text{w}) 2\text{-to-}2_q} = \langle C, (\Phi_{\pi_1, \pi_2})_{(\pi_1, \pi_2) \in S_q^{\#2}} \rangle,$$

where  $C$  is unary, and the relations  $\Phi_{\pi_1, \pi_2}$  are all ternary. A  $\tau_{(w)}$  2-to-2 $_q$ -structure  $\mathbb{A}$  is to be understood as an instance  $I_{2:2}^w = (G, \Sigma, \Phi)$  of  $\mathcal{G}_{2:2}^w$  with integer weights. The universe of  $\mathbb{A}$  is the disjoint union of the set  $V$  of vertices of  $I_{2:2}^w$ , and the set  $C$  of constraints, with the unary relation  $C$  picking out this set. For each  $(\pi_1, \pi_2) \in S_q^{\#2}$ , the relation  $\Phi_{\pi_1, \pi_2} \subseteq V^2 \times C$  contains those triples  $(u, v, c)$  where  $\Phi(u, v)$  is a pair  $(R, w)$  with  $R$  being the 2-to-2 relation associated with the pair  $(\pi_1, \pi_2)$ . The integer weight  $w$  is given by the number of elements  $c$  for which  $(u, v, c)$  is in the relation. We assume our structures satisfy the (first-order) axiom that ensures that there is at most one relation  $\Phi_{\pi_1, \pi_2}$  in which triples  $(u, v, c)$  appear, for each choice of  $u$  and  $v$ .

### 4.3 Undefinability of Regular 3XOR

The reduction in Section 3 starts from *regular* games. In contrast, the undefinability result in Theorem 2.5 is stated for general 3XOR. Thus, we begin by arguing that the proof of Theorem 2.5 can actually be used to show the undefinability of  $\text{GapREGULAR3XOR}(1, \eta)$  for some  $\eta$  strictly smaller than 1.

We first note that the  $\text{Gap3XOR}(1, \frac{1}{2} + \delta)$  is FPC undefinable even for “half-regular” 3XOR instances. That is, 3XOR instances where each variable appears in the same number of equations. To see this, note that Lemma 5 in [6] uses a bipartite unique-neighbour expander graph with  $r|X|$  nodes on the left and  $|X|$  nodes on the right. Thus the graph is 3-left-regular and is an  $(\alpha|X|, \beta)$  expander. Such graph exists for every  $X$  by [27, Chapter 4]. By a variation shown for Theorem 4.4 in [27], we claim the existence of such a graph with the extra condition that the graph is right-regular. Using this extra assumption on the graph in Lemma 5 in [6] the proof establishes that  $\text{Gap3XOR}(1, \frac{1}{2} + \delta)$  is FPC undefinable even for “half-regular” 3XOR instances.

A half-regular instance can be converted into a regular one by ensuring that any two equations share at most one variable.

First, by the unique-neighbour expander property of the graph in Lemma 5 in [6], we can assume that the half-regular 3XOR instance has no repeated equations or repeated variables within an equation. This half-regular instance  $(X, \text{Eq})$  can be converted into a regular one (call it  $(X^*, \text{Eq}^*)$ ) by replacing every equation  $e : x + y + z = b$  with three equations (as done in [22]):  $x + y_e + z_e = b$ ,  $x_e + y + z_e = b$ ,  $x_e + y_e + z = b$ , where  $x_e, y_e$ , and  $z_e$  are new variables only used for these equations.

As shown in [22], if  $X$  is fully satisfiable then so is  $X^*$  and if  $X$  is no more than  $\frac{1}{2} + \delta$ -satisfiable, then  $X^*$  is at most  $\eta$ -satisfiable for some  $\eta < 1$  (for example, taking  $\eta = 0.9$  suffices).

The reduction can be easily defined by a first-order interpretation.

### 4.4 Shuffling variables

One issue that arises with the games constructed in the reduction from Section 3 is that we have a fixed alphabet of size  $q = 2^l$  and we associate with each vertex  $(U, L)$  an arbitrary bijection between this and the  $2^l$  distinct linear functions on the space  $L + H_U$  that satisfy the equations in  $U$ . The consistency across different vertices is then enforced by the constraint relations. In order to turn this into a first-order reduction, we want to choose these bijections in a symmetry-preserving fashion.

Let  $I$  be our starting instance of 3XOR and  $I_{2:2}^T = \Theta(I)$  the transitive 2-to-2 game obtained from the first step of the reduction of Section 3, and let  $X$  be the set of variables of  $I$ . Let  $\rho \in \text{Sym}_X$  be a permutation of  $X$ . This permutation has a natural action on other

## 16:12 Undefinability of Approximation of 2-To-2 Games

objects constructed from  $X$ . In particular, for an equation  $e$  of the form  $x + y + z = b$ , we write  $\rho(e)$  for the equation  $\rho(x) + \rho(y) + \rho(z) = b$ . When  $U$  is a tuple of such equations, we write  $\rho(U)$  for the tuple obtained by applying  $\rho$  componentwise to each element of the tuple. Similarly, for other objects obtained by set and tuple constructions from  $X$ , we apply the permutation  $\rho$  to denote the natural induced action without defining it formally.

Furthermore, we also use  $\rho$  to denote the invertible linear map on  $\mathbb{F}_2^X$  obtained by applying  $\rho$  to the basis  $(e_x)_{x \in X}$ , and extending linearly to all of  $\mathbb{F}_2^X$ . Thus, in particular, for a subspace  $L \subseteq \mathbb{F}_2^X$ ,  $\rho(L)$  denotes the image of this space under this map.

The following is now straightforward.

► **Lemma 4.1** (Shuffling Variables 1). *For any permutation  $\rho \in \text{Sym}_X$ , if  $U$  and  $\rho(U)$  are both in  $\mathcal{U}$ , and  $(U, L) \in V(I_{2,2}^T)$ , then  $\rho(U, L) \in V(I_{2,2}^T)$ .*

**Proof.** Since  $\rho$  maps the basis of  $H_U$  formed by the left-hand sides of the equations in  $U$  to the corresponding basis of  $H_{\rho(U)}$ , we have  $\rho(H_U) = H_{\rho(U)}$ . By invertibility of  $\rho$ , a space  $L$  is then linearly independent of  $H_U$  if, and only if,  $\rho(L)$  is linearly independent of  $H_{\rho(U)}$ . ◀

Now, we want to choose the bijections between our set of  $2^l$  labels and the linear functions associated with a vertex  $(U, L)$  in such a way that whenever  $(U, L)$  and  $\rho(U, L)$  are both vertices in  $I_{2,2}^T$ , then they commute with  $\rho$ . For this, fix a *canonical* space  $\mathbb{F}_2^{3k}$  of dimension  $3k$ . For each  $U \in \mathcal{U}$ , we write  $X_U \subseteq X$  for the set of variables that appear in  $U$ . Since  $U$  is a sequence of  $k$  equations with pairwise disjoint sets of variables, we can fix a bijection between  $X_U$  and  $[3k]$  which induces an isomorphism  $\mu_U : \mathbb{F}_2^{X_U} \rightarrow \mathbb{F}_2^{3k}$ . These isomorphisms are easily seen to be  $\rho$ -invariant (for all  $\rho$ ), that is,

$$\forall S \in \mathbb{F}_2^{X_U}. \quad \mu_{\rho(X_U)}(\rho(S)) = \mu_U(S).$$

Under this map, there is a fixed subspace  $H \subseteq \mathbb{F}_2^{3k}$  of dimension  $k$  such that  $\mu_U(H_U) = H$  for all  $U$ . Similarly, there is a fixed collection  $\mathcal{L}$  of  $l$ -dimensional spaces such that  $\mu_U(\mathcal{L}_U) = \mathcal{L}$ . Thus, we can identify the vertices of  $I_{2,2}^T$  uniquely with pairs  $(U, L^*)$  where  $U \in \mathcal{U}$  and  $L^* \in \mathcal{L}$ . This is to be understood as the representation of the vertex  $(U, \mu_U^{-1}(L^*))$ .

Similarly, for linear functions  $f$  over  $L \in \mathcal{L}_U$ , we can define

$$(\rho(f))(x) = f(\rho^{-1}(x)) : \rho(L) \rightarrow \mathbb{F}_2 \quad \text{and}$$

$$(\mu_U(f))(x) = f(\mu_U^{-1}(x)) : \mu_U(L) \rightarrow \mathbb{F}_2.$$

Then, a linear function  $f$  on  $L + H_U$  satisfies the equations in  $U$  if, and only if,  $\mu_U(f)$  satisfies the equations in  $\mu_U(U)$ . Hence, we can interpret in a canonical way the label of a node  $(U, \mu_U^{-1}(L^*))$  as a linear function with domain  $H + L^*$  satisfying the equations in  $\mu_U(U)$ .

We now show that this can be consistently applied to the constraints of the game.

► **Lemma 4.2** (Shuffling Variables 2). *Suppose  $(U, L), (U', L') \in E(I_{2,2}^T)$  and  $\rho(U), \rho(U')$  are both in  $\mathcal{U}$ . Then*

- $(\rho(U, L), \rho(U', L')) \in E(I_{2,2}^T)$
- $\Phi((U, L), (U', L')) = \Phi(\rho(U, L), \rho(U', L'))$

**Proof.** By Lemma 4.1,  $\rho(U, L), \rho(U', L') \in V(I_{2,2}^T)$ . Also

$$\dim(\rho(L) + H_{\rho(U)} + H_{\rho(U')}) = \dim(\rho(L + H_U + H_{U'})) = \dim(L + H_U + H_{U'})$$

The equalities hold because the mapping  $\rho$  is an automorphism of  $\mathbb{F}_2^X$ . The analogous dimensionality property holds with the mapping of subspaces  $(L' + H_U + H_{U'})$  and  $(L + L' + H_U + H_{U'})$ . Therefore, the dimensionality constraint for drawing edges is invariant under the action of  $\rho$ . This proves the first bullet point.

Then if  $(f, f') \in \Phi((U, L), (U', L'))$ , it means  $\mu_U^{-1}(f)$  and  $\mu_{U'}^{-1}(f')$  are consistent on the intersection of their domains. Then  $\mu_{\rho(U)}^{-1}(f) = \rho(\mu_U^{-1}(f))$  and  $\mu_{\rho(U')}^{-1}(f') = \rho(\mu_{U'}^{-1}(f'))$  are consistent too, meaning  $(f, f') \in \Phi(\rho(U, L), \rho(U', L'))$ . Hence  $\Phi((U, L), (U', L')) \subseteq \Phi(\rho(U, L), \rho(U', L'))$ . Applying the same argument to  $\rho^{-1}$  yields the other direction. ◀

## 4.5 The reduction to the transitive game

We now describe how the reduction  $\Theta$  from Section 3.2 can be given as a first-order interpretation. Fix positive integers  $k$  and  $l$ , which are the parameters to the reduction. Given a (regular) 3XOR instance  $\mathbb{A} = (X, \text{Eq}_0^{\mathbb{A}}, \text{Eq}_1^{\mathbb{A}})$ , our interpretation maps it to the following (transitive) 2-to-2 game (with alphabet size  $2^l$ )  $\mathbb{B}$ .

**Universe.** The universe of  $\mathbb{B}$  consists of tuples of elements of  $X$  of length  $4k + 2^{3k}$ . These tuples can be seen as broken up into three parts.

- The first  $3k$  elements  $(u_{1,1}, \dots, u_{k,3})$  are the  $3k$  variables in some  $U \in \mathcal{U}$ . To define this, we need to say that they are, in order, the collection of variables of a  $k$ -tuple of equations, that no variable appears more than once, and that when two variables appear in distinct equations, they do not occur together in some other equation in  $\mathbb{A}$ .
- The next  $k$  elements  $r_1, \dots, r_k$  define the right-hand sides of the  $k$  equations in  $U$ . To encode these as binary values, we use  $r_i = u_{1,1}$  to encode the value 0 and  $r_i = u_{1,2}$  to encode the value 1. Since  $u_{1,1}$  and  $u_{1,2}$  are distinct, this works and can be specified by a first-order formula.
- The next  $2^{3k}$  elements also encode bits, using the values of  $u_{1,1}$  and  $u_{1,2}$  as 0 and 1. Think of these as specifying a subset of  $\mathbb{F}_2^{3k}$ . We can write a first-order formula that says that this subset is a subspace  $L^*$  of dimension  $l$  (since  $l$  and  $k$  are fixed, the formula is simply a big disjunction over all subspaces). Finally, we can also write a first-order formula that checks that  $L^*$  is in  $\mathcal{L}$ .

For completeness, here is the first-order sentence checking all these conditions.

$$\begin{aligned} \pi^U = & \bigwedge_{i=1}^k [\text{Eq}_0(u_{i,1}, u_{i,2}, u_{i,3}) \wedge r_i = 0] \vee [\text{Eq}_1(u_{i,1}, u_{i,2}, u_{i,3}) \wedge r_i = 1] \\ & \wedge \bigwedge_{(a,i) \neq (b,j)} u_{a,i} \neq u_{b,j} \\ & \wedge \bigwedge_{a \neq b, i, j} \neg \left( \exists x \bigvee_{(\alpha, \beta, \gamma) \in \text{Perm}(u_{a,i}, u_{b,j}, x)} \text{Eq}_0(\alpha, \beta, \gamma) \vee \text{Eq}_1(\alpha, \beta, \gamma) \right) \\ & \wedge \bigvee_{L^* \in \mathcal{L}} \left( \bigwedge_{i=0}^{2^{3k}-1} b_i = L_i^* \right) \end{aligned}$$

Where  $\text{Perm}(x, y, z)$  describes the set of permutations of  $x, y, z$ .

We can thus, as required, identify the elements of  $\mathbb{B}$  with pairs  $(U, L)$  which are the vertices of  $\Theta(\mathbb{A})$ .

**Relations.** Given two vertices  $(U, L)$  and  $(U', L')$  of  $\mathbb{B}$ , the type of constraint between them (1-to-1, 2-to-2 or no constraint at all) only depends on  $\mu_U(L), \mu_{U'}(L'), r, r'$  and  $I(U, U')$ , where  $r, r'$  are the vectors of the right-hand sides of the equations and

$$I(U, U') \triangleq \{((a, i), (b, j)) \in (\{1, \dots, k\} \times \{1, 2, 3\})^2 \mid u_{a,i} = u'_{b,j}\}$$

## 16:14 Undefinability of Approximation of 2-To-2 Games

If two pairs of vertices agree on all five of these values, there is a permutation  $\rho$  of the variables that will take one to the other and then by Lemma 4.2, they must have the same constraint between them.

Note that each of these five parameters can take only a constant number of different values, so for each constraint  $C \in \mathcal{C}_1 \cup \mathcal{C}_2$ , there is a (constant) finite set  $S_C$  containing such 5-tuples so that  $(U, L)$  and  $(U', L')$  are connected by a constraint  $C$  if, and only if,  $(\mu_U(L), \mu_{U'}(L'), r, r', I(U, U')) \in S_C$ . The formula  $\pi^C$  defining the relation  $C$  in  $\mathbb{B}$  simply states that the 5-tuple corresponding to a pair of vertices is in  $S_C$ . This translates to a disjunction of a finite number of cases and is clearly FO-definable. This concludes the reduction to the transitive game.

### 4.6 Weight approximation

We now show how to get a weighted 2-to-2 game, that is an approximation of the instance  $I_{2:2}^w$  constructed in Section 3.3. The vertices of the game are exactly those in the structure  $\mathbb{B}$  above. The main task is to define the weights, by defining a suitable set  $C$  of constraints. Recall that the vertices of  $I_{2:2}^w$  are partitioned into cliques  $C_1, \dots, C_m$  based on the 1-to-1 constraints. Suppose  $(U_1, L_1) \in C_i$  and  $(U_2, L_2) \in C_j$  are two vertices connected by a 2-to-2 constraint. Then, the weight of the constraint is

$$\sum_{\substack{U, L, L' \\ L, L' \in \mathcal{L}_U \\ \dim(L \cap L') = l-1}} 1_{(U, L) \in C_i \wedge (U, L') \in C_j} \frac{1}{|\mathcal{U}|} \frac{1}{|\{L, L' \in \mathcal{L}_U \mid \dim(L \cap L') = l-1\}|} \frac{1}{|C_i||C_j|}.$$

Each of the three factors (apart from the indicator variable) describes the probability of a certain choice in the steps of the random process which define the weights.

Of course,  $\frac{1}{|\mathcal{U}|}$  is constant for all pairs  $(U_1, L_1), (U_2, L_2)$ . Similarly,  $\frac{1}{|\{L, L' \in \mathcal{L}_U \mid \dim(L \cap L') = l-1\}|}$  is constant by the symmetry argument presented in Section 4.4. Thus, removing them from the expression does not change the relative weights of the constraints. Also, the clique size only depends on  $(U_1, L_1), (U_2, L_2)$ , so the weight expression (without the normalising factors) simplifies to

$$\frac{|\{(U, L, L') \mid (U, L) \in C_i, (U, L') \in C_j\}|}{|C_i||C_j|}. \quad (1)$$

These weights are rational, so we cannot express them directly in structures over  $\tau_{(w)} 2\text{-to-}2_q$ , which is our vocabulary for describing integer-weighted games. One potential way to handle rational weights would be to multiply all weights with a common denominator. This is not a viable option since the number of different-sized cliques grows with the size of the input, making the common denominator too large. However, we have a workaround: instead of these weights, we give an approximation that does not change the soundness parameter significantly but makes the common denominator of the weights small enough (polynomial as a function of the input size) to be definable.

► **Lemma 4.3.** *Given a weighted 2-to-2 game  $G = (V, \Sigma, \Phi, w)$ , whose value is at most  $\delta$ , any game  $G' = (V, \Sigma, \Phi, w')$  where  $\forall \phi \in \Phi. \frac{1}{\gamma} < \frac{w(\phi)}{w'(\phi)} < \gamma$  has value at most  $\delta\gamma^2$ .*

**Proof (sketch).** The sum of weights drops at most by a factor  $\gamma$ , and the sum of the weights of the satisfied constraints increases by at most a factor of  $\gamma$ . ◀

So, the idea is to approximate clique sizes so that the number of possible denominators is constant and their product grows only polynomially with the input size, while bounding the change with a suitable multiplicative factor  $\gamma$ .

Fix a vertex  $(U, L)$  in a clique  $C_i$ . Recall that  $(U', L') \in C_i$  if, and only if, there is a one-to-one constraint between  $(U, L)$  and  $(U', L')$  in  $\mathbb{B}$ . First, let us split the equations in  $U'$  into two groups: “useful” and “useless” ones. An equation in  $U'$  is useful (for  $U$ ) if it shares at least one variable with  $U$  and useless otherwise. Note that the number of useful equations of  $(U', L')$  only depends on  $U'$ , not on  $L'$ .

Next, we define an equivalence relation  $\equiv_U$  on the vertices of the game as follows:  $(U_1, L_1) \equiv_U (U_2, L_2)$  iff

- $\mu_{U_1}(L_1) = \mu_{U_2}(L_2)$ .
- $U_1$  and  $U_2$  have the same useful equations (for  $U$ ), and these equations are in the same positions within the  $k$ -tuple.
- The right-hand sides of the equations in  $U_1$  and  $U_2$  are the same.

It is easily seen that this is, indeed, an equivalence relation.

Note that the clique  $C_i$  is invariant under the equivalence relation  $\equiv_U$ : each equivalence class is either contained in  $C_i$  or disjoint with it, by Lemma 4.2 (choosing  $\rho$  to be a permutation that fixes the variables of  $U$  and any useful equations).

Now, for any  $f$  with  $0 \leq f \leq k$ , we can establish an upper bound on the number of equivalence classes with  $f$  useful equations. Recall that any node  $(U', L')$  can be uniquely represented by  $U'$  and the subspace  $\mu_{U'}(L') = L^* \in \mathcal{L}$ :

- The number of possible subspaces  $L^* \subseteq \mathbb{F}_2^{3k}$  is at most  $2^{2^{3k}}$ , as that is an upper bound for  $|\mathcal{L}|$  (in fact, it is much smaller, but for our purposes, this upper bound suffices).
- The number of ways to choose the positions of the useful equations is  $\binom{k}{f} \leq 2^k$ .
- The number of choices for the right-hand sides of the equations is  $2^k$ .
- Since the 3XOR instance is regular (each variable appears in at most  $d$  equations), the number of equations sharing a variable with  $U$  is at most  $3kd$ , so the number of ways of choosing the useful equations is bounded by  $(3kd)^k$ .

These bounds are all constants, so the number of equivalence classes within the clique, with  $f$  useful equations (call it  $\nu_{U,L}^f$ ) is bounded by a constant  $\Psi$  for all  $f, U, L$ .

The number of elements in an equivalence class with  $f$  useful equations is simply the number of ways to set the remaining  $k - f$  equations. This can be approximated by  $|\text{Eq}|^{k-f}$ . Given  $f$  useful equations, the probability of a random set of  $k - f$  equations having common variables with  $U$ , the set of useful equations or each other, or making the  $k$ -tuple invalid by having two variables from different equations which have a common equation in the 3XOR instance, converges to zero ( $O\left(\frac{k^2}{|X|}\right)$ ) as the instance size grows, due to the regularity condition. By adding all the approximate sizes of the equivalence classes within  $C_i$ , we can conclude that the approximation

$$\chi(\nu_{U,L}) \triangleq \chi(\nu_{U,L}^0, \nu_{U,L}^1, \dots, \nu_{U,L}^k) \triangleq \sum_{f=0}^k \nu_{U,L}^f |\text{Eq}|^{k-f} \approx |C_i|$$

is accurate within an arbitrarily small factor as the input size grows. Using this approximation in the weight expression (1), we see that  $\prod_{\mathbf{v} \in \{0, \dots, \Psi\}^{k+1}} \chi(\mathbf{v})^2$  is a common denominator of all weights. Multiplying all weights by this number, we get the expression

$$w((U_1, L_1), (U_2, L_2)) = |\{(U, L, L') \mid (U, L) \in C_i, (U, L') \in C_j\}| \cdot \prod_{\mathbf{v} \in \{0, \dots, \Psi\}^{k+1}} \begin{cases} \chi(\mathbf{v}) & \text{if } \mathbf{v} \neq \nu_{(U_1, L_1)} \\ 1 & \text{if } \mathbf{v} = \nu_{(U_1, L_1)} \end{cases} \cdot \prod_{\mathbf{v} \in \{0, \dots, \Psi\}^{k+1}} \begin{cases} \chi(\mathbf{v}) & \text{if } \mathbf{v} \neq \nu_{(U_2, L_2)} \\ 1 & \text{if } \mathbf{v} = \nu_{(U_2, L_2)} \end{cases} \quad (2)$$

## 16:16 Undefinability of Approximation of 2-To-2 Games

As we see next, we can define a reduction in FO to weighted 2-to-2 games using these approximate weights.

### 4.7 Defining the weighted game

Finally, we are ready to show that the construction of a weighted 2-to-2 game with approximate weights as above can be given by an FO interpretation.

**Universe.** We need to define the set of vertices, and the set of constraints. The elements of the universe are tuples of elements of  $X$  (the set of variables of the 3XOR instance  $I$ ) of length  $8k + 1 + 2^{3k+1} + Q$ , where  $Q$  is a parameter we define below.

A vertex  $(U, L)$  is coded by the first  $4k + 2^{3k}$  elements of this tuple, as before, followed by a sequence of 0s. Recall that we code bits 0 and 1 by the first and second elements of the tuple. The first of these 0s is to be interpreted as an indicator that the tuple is a vertex (it will be 1 for a constraint), and the rest are padding to make the length of the tuples match.

A constraint  $c$  is coded by a tuple where the first  $4k + 2^{3k}$  elements represent a vertex  $(U, L)$ , this is followed by a 1 (i.e. a repeat of the second element of the tuple) and then the next  $4k + 2^{3k}$  represent a second vertex  $(U', L')$ . The rest of the tuple codes a unique identifier of the constraint, ID. We construct the interpretation so that for all fixed  $(U, L), (U', L')$ , there are  $w((U, L), (U', L'))$  different identifiers where  $w$  is the approximate weight described above. We show that for this weight function, there is a formula  $W$  which defines a set of exactly  $w((U, L), (U', L'))$  tuples extending the description of  $(U, L)$  and  $(U', L')$ .

► **Lemma 4.4.** *There exists  $Q \in \mathbb{N}^+$  and a first-order formula  $W$  which defines a set  $T$  of tuples coding pairs  $(U, L), (U', L')$  together with a  $Q$ -element unique identifier and such that for each fixed  $(U, L), (U', L')$ ,  $T$  contains exactly  $w((U, L), (U', L'))$  many tuples extending  $(U, L), (U', L')$ .*

The proof of this lemma, constructing the formula  $W$  is in Section 4.8 below.

Thus, we can define the formulas defining the set of vertices and constraints. For simplicity, we use  $U, L, U', L', \text{ID}$  to describe the sub-tuple of variables in their corresponding parts of the  $N$ -tuple, where  $N = 8k + 1 + 2^{3k+1} + Q$ .

$$\text{Node}(U, L, \text{IsConstraint}, U', L', \text{ID}) \equiv \text{IsConstraint} = 0 \wedge \pi^U(U, L) \wedge \bigwedge_{x \in (U', L', \text{ID})} x = 0$$

To check if it is a valid constraint, we need

$$\begin{aligned} \text{Constraint}(U, L, \text{IsConstraint}, U', L', \text{ID}) \equiv & \text{IsConstraint} = 1 \wedge \pi^U(U, L) \wedge \pi^U(U', L') \\ & \wedge \bigvee_{C \in \mathcal{C}_2} \pi^C((U, L), (U', L')) \wedge W((U, L), (U', L'), \text{ID}) \end{aligned}$$

**Constraints.** For each  $C_{\pi_1, \pi_2} \in \mathcal{C}_2$ , we can construct the formula that defines the set of triples  $(x, y, c)$  where  $x = (U, L, 0, \dots, 0)$ ,  $y = (U', L', 0, \dots, 0)$  and  $c = (U, L, 1, U', L', \text{ID})$ , such that there is a constraint of type  $C$  between  $x$  and  $y$  and ID is a valid id of a constraint between them.

$$\Phi_{\pi_1, \pi_2}(x, y, c) \equiv \pi^{C_{\pi_1, \pi_2}}(x, y) \wedge (U, L) = (U_1, L_1) \wedge (U', L') = (U_2, L_2).$$

This completes the proof of Theorem 3.1.



## 4.8 Defining $W$

To prove Lemma 4.4 we define a first-order formula  $W(x, y, z)$  in the vocabulary  $\tau_{3\text{XOR}}$ , where  $x$ ,  $y$  and  $z$  are tuples of free variables. The formula is such that if  $x$  and  $y$  are interpreted by the elements coding the nodes  $(U, L)$  and  $(U', L')$  respectively, then there are exactly  $w((U, L)(U', L'))$  assignments of values to the tuple  $z$  that make  $W$  true. Here  $w(U, L)(U', L')$  is the expression given in Equation 2.

To define  $W$ , we construct formulas defining various elements of Equation 2. More precisely, for various numerical expressions  $e(x, y)$ , which depend on the values assigned to  $x$  and  $y$ , we construct formulas we denote  $\omega_{q,e}(x, y, z)$ , where  $q$  is the length of the tuple of variables  $z$ . These formulas have the property that when  $x$  and  $y$  are interpreted by the elements coding the nodes  $(U, L)$  and  $(U', L')$  the number of  $q$ -tuples that can be assigned to  $z$  to make  $\omega_{q,e}$  true is exactly  $e(x, y)$ . As before, we use 0 and 1 to denote the first and second elements of the tuple. Also, for a first-order formula  $\phi(x, y)$ , let  $\mathbf{1}_\phi$  denote the indicator variable that  $\phi$  is true (under an assignment of values to  $x$  and  $y$ ).

$e = \mathbf{1}$ :  $\omega_{1,e}(x, y, z) \equiv (z = 0)$

$e = \mathbf{1}_\phi$ :  $\omega_{1,e}(x, y, z) \equiv (z = 0) \wedge \phi(x, y)$

$e = e_1 \times e_2$ : Given  $\omega_{q_1,e_1}$  and  $\omega_{q_2,e_2}$ , we can define

$$\omega_{q_1+q_2,e}(x, y, z_1, \dots, z_{q_1}, z_{q_1+1}, \dots, z_{q_2}) \equiv \omega_{q_1,e_1}(z_1, \dots, z_{q_1}) \wedge \omega_{q_2,e_2}(z_{q_1+1}, \dots, z_{q_2})$$

$e = e_1 + e_2$ : Given  $\omega_{q_1,e_1}$  and  $\omega_{q_2,e_2}$ , (assuming without loss of generality that  $q_2 \geq q_1$ ), we can define

$$\omega_{1+q_2,e}(x, y, z_1, z_2, \dots, z_{q_2+1}) \equiv \left[ z_1 = 0 \wedge \omega_{q_1,e_1}(z_2, \dots, z_{q_1+1}) \wedge \bigwedge_{i=q_1+2}^{q_2+1} z_i = 0 \right] \vee \left[ z_1 = 1 \wedge \omega_{q_2,e_2}(z_2, \dots, z_{q_2+1}) \right]$$

$e = |\text{Eq}|$ : It suffices to take a formula defining the disjoint union of the relations  $\text{Eq}_0$  and  $\text{Eq}_1$ .

$$\omega_{4,e}(x, y, z_1, z_2, z_3, z_4) \equiv (z_1 = 0 \wedge \text{Eq}_0(z_2, z_3, z_4)) \vee (z_1 = 1 \wedge \text{Eq}_1(z_2, z_3, z_4))$$

$e = |\{(U_1, L_1, L_2) \mid (U_1, L_1) \in C_i, (U_1, L_2) \in C_j\}|$ : The numerator in Equation 1 (and a term in Equation 2) is  $e = |\{(U_1, L_1, L_2) \mid (U_1, L_1) \in C_i, (U_1, L_2) \in C_j\}|$ . We can get a formula for this by defining exactly this set of tuples. Here  $z$  is a tuple of variables composed of three tuples  $z_1$ ,  $z_2$  and  $z_3$  where  $z_1$  has length  $4k$  and each of  $z_2$  and  $z_3$  is of length  $2^{3k}$ .

$$\begin{aligned} \omega_{4k+2*2^{3k},e}(x, y, z) &= \pi^U(z_1, z_2) \wedge \pi^U(z_1, z_3) \wedge \bigvee_{C \in \mathcal{C}_2} C((z_1, z_2), (z_1, z_3)) \\ &\wedge \bigvee_{C \in \mathcal{C}_1} C((z_1, z_2), x) \wedge \bigvee_{C \in \mathcal{C}_1} C((z_1, z_3), y) \end{aligned}$$

**Defining the size of the equivalence classes.** Another element of Equation 2 are conditions of the form  $\nu_{U,L}^f = r$  for various values of  $r$ . We now construct a formula  $\nu^{f, \geq r}(x)$  with  $4k + 2^{3k}$  free variables that expresses the condition  $\nu_{U,L}^f \geq r$  when  $x$  is interpreted by the tuple coding  $(U, L)$ . In the following, lower case letters  $u, l$ , possibly with subscript indices always denote tuples of variables of length  $4k$  and  $2^{3k}$  respectively. Recall that two elements in the clique are in the equivalence relation  $\equiv_{(U,L)}$  if, and only if, their  $L$  values are the same and share the same useful equations with the same positions.

## 16:18 Undefinability of Approximation of 2-To-2 Games

We begin with defining a couple of auxiliary formulas. For any  $j \in \{1, \dots, k\}$ , the formula  $\text{useful}_j(x, u)$  says of a tuple  $u$  that the  $j$ th equation it represents is useful and the formula  $\text{diff}_j(u_1, u_2)$  asserts that the two tuples  $u_1$  and  $u_2$  differ in the  $j$ th equation:

$$\text{useful}_j(x, u) \equiv \bigvee_{i \in \{1, \dots, 3k\}} (u_{3(j-1)+1} = x_i \vee u_{3(j-1)+2} = x_i \vee u_{3(j-1)+3} = x_i); \text{ and}$$

$$\text{diff}_j(u_1, u_2) \equiv (u_1)_{3(j-1)+1} \neq (u_2)_{3(j-1)+1} \vee (u_1)_{3(j-1)+2} \neq (u_2)_{3(j-1)+2} \vee (u_1)_{3(j-1)+3} \neq (u_2)_{3(j-1)+3} \vee (u_1)_{3k+j} \neq (u_2)_{3k+j}.$$

With these, we can define  $\nu^{f, \geq r}(x)$  as a formula which asserts the existence of  $r$  nodes

$$\exists u_1, l_1, \dots, u_r, l_r \bigwedge_i \pi^U(u_i, l_i);$$

which are in the same clique as the node coded by  $x$

$$\bigwedge_i \bigvee_{C \in \mathcal{C}_1} C(x, u_i, l_i);$$

all have  $f$  useful equations

$$\bigwedge_{i \in \{1, \dots, r\}} \left\{ \bigvee_{S \subseteq \{1, \dots, k\}, |S|=f} \left[ \bigwedge_{j \in S} \text{useful}_j(x, u_i) \leftrightarrow j \in S \right] \right\};$$

and such that no two nodes are  $\equiv_{(U, L)}$  equivalent when  $x$  is interpreted as  $(U, L)$

$$\bigwedge_{i \neq j \in \{1, \dots, r\}} l_i \neq l_j \vee \bigvee_{o \in \{1, \dots, k\}} (\text{useful}_o(u_i) \wedge \text{diff}_o(u_i, u_j)).$$

Then, as usual,  $\nu^{f, r}(x) \equiv \nu^{f, \geq r}(x) \wedge \neg \nu^{f, \geq (r+1)}(x)$ . To give an expression for  $\omega_q(\nu_{U, L}^f)$  for some  $q$ , we can rewrite it as  $\sum_{r=1}^{\Psi} 1_{\nu_{U, L}^{f, r}} \cdot r$  and construct the expression using the composition rules (constants can be constructed via repeated addition of ones, addition, multiplication and indicator variables are defined above)

**Putting it all together.** For each term in Equation 2, we have described how to define a corresponding formula. Case splits can be handled via indicator variables and constants by repeatedly adding 1s. By a repeated application of the addition and multiplication rules,  $W$  can be constructed.

## 5 Consequences

### 5.1 Unique Games

An immediate corollary of the definable 2-to-2 games theorem is the inapproximability of unique games by any constant factors:

Given a (weighted) 2-to-2 game  $I$ , we can map it to a Unique Game  $I'$  by splitting every constraint into two: given a constraint of type  $C_{\pi_1, \pi_2}$ , we can replace them with two 1-to-1 constraints of type  $C_{\pi_1}$  and  $C_{\pi_2}$ . A colouring of the nodes then satisfies the constraint  $C_{\pi_1, \pi_2}$  in  $I$  if, and only if, exactly one of the two constraints is satisfied in  $I'$ . Note that a colouring can only satisfy at most one of the two constraints. This gives a reduction from  $\text{GapWEIGHT2-TO-}2_q(1, \delta)$  to  $\text{GapWEIGHTUG}_q(\frac{1}{2}, \frac{\delta}{2})$  for any  $\delta > 0$ .

This reduction is clearly FO-definable: the universe remains the same; then, for a 1-to-1 constraint  $C_\pi$ , we can determine if  $(x, y, c)$  represents a constraint of this type with the sentence  $\Phi_\pi(x, y, c) \equiv \bigvee_{\pi_2} \Phi_{\pi, \pi_2}(x, y, c)$ .

► **Theorem 5.1.** *For every  $\delta > 0$ , there exists  $q \in \mathbb{N}^+$  so that  $\text{GapWEIGHTUG}_{q, \frac{1}{2}}(\delta)$  is FPC undefinable.*

This undefinability gap is stronger, at least in terms of the completeness and soundness parameters, than the gaps proved by Tucker-Foltz [26], the only previously known undefinability gaps for Unique Games. However, our construction uses weighted instances, so we can only conclude the undefinability gap over the domain of weighted unique games. Since the gaps in [26] are proved for unweighted games, they are incomparable to Theorem 5.1.

## 5.2 Vertex Cover

Another consequence of Theorem 2.3 is the NP-Hardness of approximating the Vertex Cover problem by a factor better than  $\sqrt{2}$ . The Unique Games Conjecture implies that nothing better than a factor 2 approximation is possible. This is tight, since polynomial-time algorithms achieving a 2-approximation are known. Before the results of Khot et al. establishing Theorem 2.3 the best known inapproximability result, conditional only on  $P \neq NP$ , was  $\approx 1.36$ . Atserias and Dawar [6] showed a corresponding unconditional FPC undefinability result. We improve on this with the following.

► **Theorem 5.2 (FPC-IS).** *For every  $\epsilon, \delta > 0$ ,  $\text{GapIS}(1 - \frac{1}{\sqrt{2}} - \delta, \epsilon)$  is not definable in FPC.*

Here IS is the function problem giving the size of a maximal independent set in a graph as a proportion of the total number of vertices. This is equivalent to the FPC undefinability of  $\text{GapVertexCover}(\frac{1}{1-\epsilon, \sqrt{2}} + \delta)$ , implying the FPC-inapproximability of vertex cover by a factor smaller than  $\sqrt{2}$ . The theorem follows from the reduction presented in [22, Chapter 5] which can be defined in First-Order Logic using standard methods.

## 5.3 Graph Colouring

Perhaps the most striking consequence of our result is the following.

► **Theorem 5.3.** *For every  $t \geq 3$ , the class of 3-colourable graphs are not FPC separable from those that are not  $t$ -colourable.*

Theorem 5.3 should be contrasted with what is known about the NP-hardness of promise graph colouring. It is known that it is NP-hard to separate the 3-colourable graphs from those that are not 5-colourable [7]. It is conjectured that it is NP-hard to separate the 3-colourable graphs from those that are not  $t$ -colourable for all  $t \geq 3$ , but this is open even for  $t = 6$ . Thus, Theorem 5.3 provides the first significant example of an FPC hardness of approximation result that is open in the classical setting of NP-hardness.

Guruswami and Sandeep [16] show a reduction from  $\text{GapIRREG2TO2}_{j,q}(1, \delta)$  to the problem of separating 3-colourable graphs from non- $t$ -colourable ones [12]. The reduction is easily definable in first-order logic, proving Theorem 5.3.

## 6 Conclusion

We have shown that the reductions involved in the proof of the celebrated proof by Khot, Minzer and Safra of the 2-to-2 games theorem can all be implemented as interpretations in first-order logic. This means that the NP-hardness they establish of separating nearly satisfiable instances from highly unsatisfiable ones can be turned into an unconditional inseparability result in FPC. Moreover, the result is achieved with *perfect completeness*: it is impossible to separate with an FPC sentence the fully satisfiable 2-to-2 games from those that are highly unsatisfiable.

From this result we are able to derive a number of consequences, the most striking of which is that it is impossible to separate with an FPC sentence the graphs that are 3-colourable from those that are not  $t$ -colourable for any constant  $t$ . The NP-hardness of such a separation is only conjectured for values  $t$  larger than 5. We also obtain strong FPC undefinability results for approximation of unique games. In terms of approximation ratios these are an improvement over those of Tucker-Foltz [26]. However, the latter results were obtained for *unweighted* games while ours are for weighted games.

This work suggests a number of further directions to pursue. One is an investigation of the FPC definability of promise constraint satisfaction problems (PCSP). The  $t$ -colouring of 3-colourable graphs is one such example, but PCSP are a very active current area of investigation. Our results could also be tightened by showing them for unweighted instances rather than with weights. Indeed, we believe that Theorem 5.1 could be improved to apply to unweighted games as well, making it a direct improvement of the results of [26]. For this improvement, it would be sufficient to prove the FPC analogue for the result of Crescenzi et al. [9] showing a gap reduction from weighted CSP instances to unweighted ones. The proof of Khot, Minzer and Safra applies this reduction to establish Theorem 2.3 on unweighted games. This merits further study.

---

## References

- 1 Matthew Anderson and Anuj Dawar. On symmetric circuits and fixed-point logics. *Theory of Computing Systems*, 60(3):521–551, July 2017. doi:10.1007/s00224-016-9692-2.
- 2 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, May 1998. doi:10.1145/278298.278306.
- 3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of np. *J. ACM*, 45(1):70–122, January 1998. doi:10.1145/273865.273901.
- 4 Albert Atserias, Andrei Bulatov, and Anuj Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410(18):1666–1683, 2009. Automata, Languages and Programming (ICALP 2007). doi:10.1016/j.tcs.2008.12.049.
- 5 Albert Atserias and Víctor Dalmau. Promise constraint satisfaction and width. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022*, pages 1129–1153. SIAM, 2022. doi:10.1137/1.9781611977073.48.
- 6 Albert Atserias and Anuj Dawar. Definable inapproximability: New challenges for duplicator, 2019. arXiv:1806.11307.
- 7 Libor Barto, Jakub Bulín, Andrei Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. *Journal of the ACM*, 68(4):1–66, July 2021. doi:10.1145/3457606.
- 8 Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. ACM*, 61(1), January 2014. doi:10.1145/2556646.
- 9 Pierluigi Crescenzi, Riccardo Silvestri, and Luca Trevisan. On weighted vs unweighted versions of combinatorial optimization problems. *Inf. Comput.*, 167(1):10–26, May 2001. doi:10.1006/inco.2000.3011.

- 10 Anuj Dawar. The nature and power of fixed-point logic with counting. *ACM SIGLOG News*, 2(1):8–21, January 2015. doi:10.1145/2728816.2728820.
- 11 Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. Towards a proof of the 2-to-1 games conjecture? In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 376–389, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3188745.3188804.
- 12 Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. *SIAM Journal on Computing*, 39(3):843–873, January 2009. doi:10.1137/07068062x.
- 13 Irit Dinur and Shmuel Safra. On the hardness of approximating label-cover. *Information Processing Letters*, 89(5):247–254, March 2004. doi:10.1016/j.ipl.2003.11.007.
- 14 Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Springer, 2nd edition, 1999.
- 15 Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, March 1996. doi:10.1145/226643.226652.
- 16 Venkatesan Guruswami and Sai Sandeep. d-to-1 hardness of coloring 3-colorable graphs with  $\epsilon$  colors. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.
- 17 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001. doi:10.1145/502090.502098.
- 18 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*, STOC '02, pages 767–775, New York, NY, USA, 2002. Association for Computing Machinery. doi:10.1145/509907.510017.
- 19 Subhash Khot. On the unique games conjecture (invited survey). In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 99–121, 2010. doi:10.1109/CCC.2010.19.
- 20 Subhash Khot, Dor Minzer, and Muli Safra. On independent sets, 2-to-2 games, and grassmann graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 576–589, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3055399.3055432.
- 21 Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in Grassmann graph have near-perfect expansion. *Annals of Mathematics*, 198(1):1–92, 2023. doi:10.4007/annals.2023.198.1.1.
- 22 Dor Minzer. *On Monotonicity Testing and the 2-to-2 Games Conjecture*, volume 49. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022.
- 23 Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 245–254, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1374376.1374414.
- 24 Benjamin Rossman. Equi-rank homomorphism preservation theorem on finite structures. In *33rd EACSL Annual Conference on Computer Science Logic, CSL*, 2025.
- 25 Khot Subhash, Dor Minzer, and Muli Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 592–601, 2018. doi:10.1109/FOCS.2018.00062.
- 26 Jamie Tucker-Foltz. Inapproximability of Unique Games in Fixed-Point Logic with Counting. *Logical Methods in Computer Science*, Volume 20, Issue 2, April 2024. doi:10.46298/lmcs-20(2:3)2024.
- 27 Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012. doi:10.1561/0400000010.