Satisfiability of commutative vs. non-commutative CSPs*

Andrei A. Bulatov
SFU
abulatov@sfu.ca

Stanislav Živný
University of Oxford
standa.zivny@cs.ox.ac.uk

Abstract

The Mermin-Peres magic square is a celebrated example of a system of Boolean linear equations that is not (classically) satisfiable but is satisfiable via linear operators on a Hilbert space of dimension four. A natural question is then, for what kind of problems such a phenomenon occurs? Atserias, Kolaitis, and Severini answered this question for all Boolean Constraint Satisfaction Problems (CSPs): For 2-SAT, HORN-SAT, and DUAL HORN-SAT, classical satisfiability and operator satisfiability is the same and thus there is no gap; for all other Boolean CSPs, the two notions differ as there is a gap, i.e., there are unsatisfiable instances that are satisfied via operators on a finite-dimensional Hilbert space. We generalize their result to CSPs on *arbitrary finite* domains: CSPs of so-called bounded-width have no satisfiability gap, whereas all other CSPs have a satisfiability gap.

1 Introduction

Symmetry leads to efficient computation. This phenomenon has manifested itself in several research areas that have one aspect in common, namely a model of computation with local constraints that restrict the solution space of the problem of interest. An elegant way to describe such problems is in the framework of Constraint Satisfaction Problems (CSPs). CSPs have driven some of the most influential developments in theoretical computer science, from NP-completeness to the PCP theorem to semidefinite programming algorithms to the Unique Games Conjecture. The mathematical structure of tractable decision CSPs [15, 57], infinite-domain CSPs [7, 8], optimization CSPs [53], as well as approximable CSPs [50, 10], is now known to be linked to certain forms of higher-order symmetries of the solution spaces. A recently emerging research direction links CSPs with foundational topics in physics and quantum computation [18, 17, 2, 47, 41].

Constraint Satisfaction Problems CSPs capture some of the most fundamental computational problems, including graph and hypergraph colorings, linear equations, and variants and generalizations of satisfiability. Informally, one is given a set of variables and a set of constraints, each depending only on constantly many variables. Given a CSP instance, the goal is to find an assignment of values to all the variables so that all constraints are satisfied. For example, if the domain is $\{r, g, b\}$ and the constraints are of the form $R(x, y)$, where $R = \{(r, g), (g, r), (g, b), (b, g), (r, b), (b, r)\}$ is the binary disequality relation on $\{r, g, b\}$, we obtain the classic graph 3-COLORABILITY problem. If the domain is $\{r, g, b, o\}$ and the constraints are of the form $R(x, y)$, where $R = \{(r, g), (g, r), (g, b), (b, g), (b, o), (o, b), (o, r), (r, o)\}$, we obtain a variant of the graph 4-COLORABILITY problem in which adjacent vertices must be assigned different colors and, additionally, red and blue vertices must not be adjacent and green and orange vertices must not be adjacent.

*This work was supported by UKRI EP/X024431/1 and NSERC Discovery grant. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. All data is provided in full in the results section of this paper.

Back in 1978, Schaefer famously classified all Boolean CSPs as solvable in polynomial time or NP-hard [51].[1] The tractable cases are the standard textbook problems, namely 2-SAT, HORN-SAT, DUAL HORN-SAT, and system of linear equations on a two-element set.[2] Hell and Nešetřil studied a special case of CSPs known as graph homomorphisms [28]. These are CSPs in which all constraints involve the same binary symmetric relation, i.e., a graph. The above-mentioned 3-COLORABILITY problem is the homomorphism problem to K_3, the undirected clique on three vertices, say $\{r, g, b\}$. The above-mentioned variant of 4-COLORABILITY is the homomorphism problem to C_4, the undirected cycle on four vertices, say $\{r, g, b, o\}$. Generalizing greatly the classic result of Karp that k-COLORABILITY is solvable in polynomial time for $k \leq 2$ and NP-hard for $k \geq 3$ and other concrete problems such as the (tractable) variant of 4-COLORABILITY above, Hell and Nešetřil obtained in 1990 a complete classification of such CSPs [27]. Motivated by these two results and the quest to identify the largest subclass of NP that could exhibit a dichotomy and thus avoid NP-intermediate cases, Feder and Vardi famously conjectured that *all* CSPs on finite domains admit a dichotomy; i.e., are either solvable in polynomial time or are NP-hard [23].

Following the so-called algebraic approach to CSPs, pioneered by Jeavons, Cohen an Gyssens [30] and Bulatov, Jeavons, and Krokhin [12], the conjecture was resolved in the affirmative in 2017 by Bulatov [15] and, independently, by Zhuk [56, 57]. The algebraic approach allows for a very clean and precise characterization of what makes certain CSPs computationally tractable — this is captured by the notion of polymorphisms, which can be thought of as multivariate symmetries of solutions spaces of CSPs. Along the way to the resolution of the Feder-Vardi dichotomy conjecture, the algebraic approach has been successfully used to establish other results about CSPs, e.g., characterizing the power of local consistency algorithms for CSPs [11, 3], characterizing robustly solvable CSPs [25, 20, 4], classifying the complexity of exact optimization CSPs [53, 37], the tremendous progress on classifying the complexity of CSPs on infinite domains [7], and very recently using SDPs for robustly solving certain promise CSPs [9].

Operator Constraint Satisfaction Problems Consider the following instance of a Boolean CSP, consisting in nine variables $x_1, \ldots, x_9$ over the Boolean domain $\{-1, +1\}$ and the following six constraints:

$$\begin{aligned} x_1x_2x_3 &= +1, & x_1x_4x_7 &= +1, \\ x_4x_5x_6 &= +1, & x_2x_5x_8 &= +1, \\ x_7x_8x_9 &= +1, & x_3x_6x_9 &= -1. \end{aligned} \tag{1}$$

Graphically, this system of equations can be represented by a square, where each equation on the left of (1) comes from a row, and each equation on the right of (1) comes from a column.

x_1	x_2	x_3	$+1$
x_4	x_5	x_6	$+1$
x_7	x_8	x_9	$+1$
$+1$	$+1$	-1	

The system of equations (1) has no solution in the Boolean domain $\{-1, +1\}$: By multiplying the left-hand sides of all equations we get $+1$ because every variable occurs twice in the system and $x_i^2 = +1$ for every $1 \leq i \leq 9$. However, by multiplying the right-hand sides of all equations, we get -1. Note that this argument used implicitly the assumption that the variables commute pairwise even if they do not appear in

[1] We call a CSP *Boolean* if the domain of all variables is of size two. Some papers call such CSPs binary. We use the term *binary* for a relation of arity two and for CSPs whose constraints involve binary relations.

[2] There are also two trivial, uninteresting cases called 0-valid and 1-valid.

the same equation, which is true over $\{-1, +1\}$. Thus, this argument does not hold if one assumes that only variables occurring in the same equation commute pairwise. In fact, Mermin famously established that the system (1) has a solution consisting of linear operators on a Hilbert space of dimension four [44, 45] and the construction is now know as the Mermin-Peres magic square [48]. This construction proves the Bell-Kochen-Specker theorem on the impossibility to explain quantum mechanics via hidden variables [6, 34].

Any Boolean CSP instance, just like the one above, can be associated with a certain non-local game with two players, say Alice and Bob, who are unable to communicate while the game is in progress. Alice is given a constraint at random and must return a satisfying assignment to the variables in the constraint. Bob is given a variable from the constraint and must return an assignment to the variable. The two players win if they assign the same value to the chosen variable. With shared randomness, Alice and Bob can play the game perfectly if and only the instance is satisfiable [18]. The Mermin-Peres construction [44, 48] was the first example of an instance where the players can play perfectly by sharing an entangled quantum state although the instance is not satisfiable. We note that [44, 48] were looking at quantum contextuality scenarios rather than non-local games and it was Aravind who reformulated the construction in the above-described game setting [1], cf. also [16]. The game was introduced for any Boolean CSP by Cleve and Mittal [18] and further studied by Cleve, Liu, and Slofstra [17].

Every Boolean relation can be identified with its characteristic function, which has a unique representation as a multilinear polynomial via the Fourier transform. The multilinear polynomial representation of Boolean relations (and thus also Boolean CSPs) makes it possible to consider *relaxations* of satisfiability in which the variables take values in a suitable space, rather than in $\{-1, +1\}$. Such relaxations of satisfiability have been considered in the foundations of physics long ago, playing an important role in our understanding of the differences between classical and quantum theories. In detail, given a Boolean CSP instance, a classical assignment assigns to every variable a value from $\{-1, +1\}$. An operator assignment assigns to every variable a linear operator A on a finite-dimensional Hilbert space (which means, up to a choice of basis, a matrix with complex entries) so that $A^2 = I$ and each A is self-adjoint, i.e., $A = A^*$ and thus in particular A is normal, meaning that $AA^* = A^*A$.[3] Furthermore, it is required that operators assigned to variables from the scope of some constraint pairwise commute.

Ji showed that for Boolean CSPs corresponding to 2-SAT there is no difference between (classical) satisfiability and satisfiability via operators [31]. Later, Atserias, Kolaitis, and Severini established a complete classification for all Boolean CSPs parameterized by the set of allowed constraint relations. In particular, they showed that only CSPs whose relations come from 2-SAT, HORN-SAT, or DUAL HORN-SAT have "no satisfiability gap" in the sense that (classic) satisfiability is equivalent to operator satisfiability; for all other Boolean CSPs, there is a satisfiability gap in the sense that there are instances that are not (classically) satisfiable but are satisfiable via operators just as in the Mermin-Peres magic square. The "no-gap" part of the result is established by the substitution method [18]. The "gap" part of the result is established by showing that reductions between CSPs based on primitive positive formulas, which preserve complexity and were used to establish Schaefer's classification of Boolean CSPs, preserve satisfiability gaps.

Contributions As our main contribution, we generalize the result of Atserias et al. [2] from Boolean CSPs to CSPs on *arbitrary finite* domains. As has been done in, e.g, [24, 19], we represent a finite domain of size d by the d-th roots of unity, and require that each operator A in an operator assignment should be normal (i.e. $AA^* = A^*A$) and should satisfy $A^d = I$. The representation of non-Boolean CSPs relies on multi-dimensional Fourier transform. Our main result proves that CSPs of *bounded width* (on arbitrary finite domains) do not have a satisfiability gap, meaning that classical satisfiability is equivalent to satisfiability via operators (Theorem 9); and all other CSPs do have a satisfiability gap, meaning that classical satisfiability is not equivalent to satisfiability via operators (Theorem 19).

[3] A^* denotes the conjugate transpose; all concepts are fully defined and details provided in Section 2.

Theorem (Main result, informal statement)**.** *Let* Γ *be an arbitrary finite set of relations on a finite domain. If* $\mathrm{CSP}(\Gamma)$ *has bounded width then classical and operator satisfiability are the same for* $\mathrm{CSP}(\Gamma)$*. Otherwise, classical and operator satisfiability are not the same for* $\mathrm{CSP}(\Gamma)$.

The proof relies on several ingredients. Firstly, we observe that results establishing that primitive positive definability preserve satisfiability gaps [2] can be lifted from Boolean to arbitrary finite domains. Secondly, for CSPs of bounded width we show that there is no difference between classical and operator satisfiability by simulating the inference by the so-called Singleton Linear Arc Consistency (SLAC) algorithm in polynomial equations. We note that while there are several (seemingly stronger) algorithms for CSPs of bounded width, our proof relies crucially on the special structure of SLAC and the breakthrough result of Kozik that SLAC solves all CSPs of bounded width [38]. Thirdly, to prove that CSPs of unbounded width have satisfiability gaps we use the algebraic approach to CSPs, namely, we show that not only primitive positive definitions but also other reductions, namely going to the core, adding constants, and restrictions to subalgebras and factors, preserve satisfiability gaps. Finally, for all odd d we give an explicit construction of a CSP instance with a satisfiability gap, which is a generalization of the Mermin-Peres square.

We note that that there is a significant hurdle to go from Boolean CSPs to CSPs over arbitrary finite domains. While any non-Boolean CSP can be Booleanized via indicator variables and extra constraints, such constructions do not immediately imply classifications of non-Boolean CSPs as the "encoding constraints" are intended to be used in only a particular way. Indeed, while the complexity of Boolean CSPs was established by Schaefer in 1978 [51] and the complexity of CSPs on three-element domains was established by Bulatov in 2002 [13, 14], the dichotomy for all finite domains was only established in 2017 [15, 56, 57]. Similarly for other variants of CSPs and different notions of tractability, results on Boolean domains, including the work of Atserias et al. [2], rely crucially on the explicit knowledge of the structure of relations on Boolean domains (established by Post [49]), which is not known for non-Boolean CSPs. Indeed, on the tractability side, the structure of relations in tractable Boolean CSPs is simple and very well understood; on the intractability side, reductions based only on primitive positive definitions suffice for a complete classification of Boolean CSPs. Neither of these two facts is true for non-Boolean CSPs.

We find it fascinating that bounded width is the borderline for satisfiability gaps for CSPs, thus linking a notion coming from a natural combinatorial algorithm for CSPs with a foundational topic in quantum computation. This is yet another result confirming how fundamental the notion of bounded width is, capturing not only the power of the local consistency algorithm [40, 43, 11, 3] as conjectured in [23] with links to Datalog, pebble games, and logic [23, 36], but also robust solvability of CSPs [4], exact solvability of valued CSPs by LP [54] and SDP [55] relaxations, and now also satisfiability via operators.

Related work The present article is concerned with distinguishing classical satisfiability from satisfiability over finite-dimensional operators. Atserias et al. [2] also considered satisfiability over infinite-dimensional bounded operators and obtained a complete classification of Boolean CSPs, relying on a breakthrough result of Slofstra, who established that there is system of linear equations that is not satisfiable via finite-dimensional operators but is satisfiable via infinite-dimensional bounded operators [52]. We believe that our methods can be used for the infinite-dimensional case as well but leave it for future work. Very recently, Paddock and Slofstra [47] streamlined the results of Atserias et al. [2]. Moreover, [47] gives an overview of other notions of satisfiability and their relationship, including the celebrated MIP*=RE result of Ji et al. [32, 33]. We note that the notion of quantum homomorphism from the work of Mančinska and Roberson [42] is different from ours as it requires that the operators A should be idempotent, i.e., $A^2 = A$. Recent work of Culf, Mousavi, and Spirig studies approximability of operator CSPs [19].

2 Preliminaries

We denote by $[r]$ the set $\{1, 2, \ldots, r\}$.

CSPs An instance of the *constraint satisfaction problem* (CSP) is a triple $\mathcal{P} = (V, D, \mathcal{C})$, where V is a set of variables, D is a set of domain values, and $\mathcal{C}$ is a set of constraints. Every constraint in $\mathcal{C}$ is a pair $\langle \mathbf{s}, R \rangle$, where $\mathbf{s} \in V^r$ is the constraint scope and $R \subseteq D^r$ is the constraint relation of arity $r = ar(R)$. Given a CSP instance $\mathcal{P}$, the task is to determine whether there is an assignment $s : V \to D$ that assigns to every variable from V a value from D in such a way that all the constraints are satisfied; i.e., $(s(v_1), \ldots, s(v_r)) \in R$ for every constraint $\langle (v_1, \ldots, v_r), R \rangle \in \mathcal{C}$. An assignment satisfying all the constraints is also called a *solution*.

Let D be a fixed finite set. A finite set Γ of relations over D is called a *constraint language* over D. We denote by $\mathrm{CSP}(\Gamma)$ the class of CSP instances in which all constraint relations belong to Γ. A mapping $\varrho : D \to D$ is an *endomorphism* or *unary polymorphism* of Γ if, for any $R \in \Gamma$ (say, r-ary) and any $(a_1, \ldots, a_r) \in R$, the tuple $(\varrho(a_1), \ldots, \varrho(a_r))$ belongs to R.

Bounded width Intuitively, CSPs of bounded width are those CSPs for which unsatisfiable instances can be refuted via local propagation. An obvious obstruction to bounded width, in addition to NP-hard CSPs, is CSPs encoding systems of linear equations [23]. A celebrated result of Barto and Kozik established that CSPs of bounded width are precisely those CSPs that cannot simulate, in a precise sense, linear equations [3]. While bounded width has several characterizations [40, 43, 11, 3, 39],[4] we will rely on the result of Kozik [38] that established that every CSP of bounded width can be solved through constraint propagation of a very restricted type, so-called *Singleton Linear Arc-Consistency* (SLAC).

In order to explain SLAC, we need to start with *Arc-Consistency* (AC). AC is one of the basic levels of local consistency notions. It is a property of a CSP and also an algorithm turning an instance $\mathcal{P} \in \mathrm{CSP}(\Gamma)$ into an equivalent subinstance $\mathcal{P}' \in \mathrm{CSP}(\Gamma)$ that satisfies the AC property. Intuitively, given an instance $\mathcal{P} = (V, D, \mathcal{C}) \in \mathrm{CSP}(\Gamma)$, the AC algorithm starts with setting the domain $D_v = D$ for every variable $v \in V$. Then, it prunes the sets $\{D_v\}_{v \in V}$ in an iterative fashion, terminating (in polynomial time in the size of $\mathcal{P}$) with a maximal subinstance of $\mathcal{P}$ that satisfies the AC condition; namely, for every variable $v \in V$, every value $a \in D_v$, and every constraint $\langle \mathbf{s}, R \rangle \in \mathcal{C}$ such that $\mathbf{s}[i] = v$ for some i, there is a tuple $\mathbf{a} \in R$ with $\mathbf{a}[i] = a$. The resulting subinstance $\mathcal{P}'$ is *equivalent* to $\mathcal{P}$ in the sense that $\mathcal{P}$ has a solution if and only if $\mathcal{P}'$ has a solution. We say that AC *solves* an instance $\mathcal{P}$ if $\mathcal{P}$ has a solution whenever $\mathcal{P}'$ is consistent; i.e., none of the sets D_v in $\mathcal{P}'$ is empty. AC is not strong enough to solve all CSPs of bounded width (e.g., 2-SAT) but its full power is understood [23, 21].

Equivalently, Arc-Consistency can be described in terms of a Datalog program [35]. In general, a Datalog program derives facts about an instance $\mathcal{P} \in \mathrm{CSP}(\Gamma)$ using a fixed set of rules that depend on the constraint language Γ. The rules are defined using relations from Γ called *extensional databases* (EDBs) as well as a number of auxiliary relations called *intensional databases* (IDBs). Each rule consists of a *head*, which is a single IDB, and the *body*, which is a sequence of IDBs and EDBs. The execution of the program updates the head IDB whenever the body of the rule is satisfied, that is, every EDB and IDB in the body is satisfied. The computation of the program ends when no relation can be updated, or when the goal predicate is reached. If we require that a Datalog program should only include unary IDBs and that every rule should have at most one EDB then the power of the program for CSPs amounts to AC. In detail, the AC Datalog program has a unary relation (IDB) $T_S(v)$ for each subset $S \subseteq D$. Then for every $\langle (v_1, \ldots, v_r), R \rangle \in \mathcal{C}$ and for any IDBs $T_{S_1}, \ldots, T_{S_m}, T_S$ the program contains the rule

$$T_S(v_i) : -R(v_1, \ldots, v_r), T_{S_1}(v_{i_1}), \ldots, T_{S_m}(v_{i_m})$$

[4] One characterization implies that checking whether a constraint language has bounded width is decidable [39].

Input: A CSP instance $\mathcal{P} = (V, D, \mathcal{C})$.
Output: A SLAC-consistent instance $\mathcal{P}'$ equivalent to $\mathcal{P}$

1. **for each** $v \in V$ **set** $D_v = U_d$
2. $\mathcal{P}' = \mathcal{P} + \sum_{v \in V} \langle v, D_v \rangle$
3. **until** the process stabilizes

   3.1 **pick** a variable $v \in V$

   3.2 **for each** $a \in D_v$ **do**

      3.2.1 **run** Linear Arc-Consistency on $\mathcal{P}' + \langle v, \{a\} \rangle$

      3.2.2 **if** the problem is inconsistent, **set** $D_v = D_v - \{a\}$

      endfor

   enduntil
4. **return** $\mathcal{P}'$

Figure 1: SLAC.

if for any $\mathbf{a} \in R$ such that $\mathbf{a}[i_j] \in S_j$ we have $\mathbf{a}[i] \in S$. The Arc-Consistency algorithm is *Linear* if $m = 1$ for every rule in the corresponding Datalog program.

The Singleton Arc-Consistency (SAC) algorithm is a modification of the AC algorithm [22]. SAC updates the sets $\{D_v\}_{v \in V}$ as follows: it removes a from D_v if the current instance augmented with the constraint fixing the value a to the variable v is found inconsistent by the AC algorithm. Finally, the Singleton Linear Arc-Consistency algorithm is a modification of SAC (due to Kozik [38] and Zhuk [57]) that uses the Linear AC algorithm rather than the AC algorithm. Kozik has shown that SLAC solves all CSPs of bounded width [38]. As with AC, SLAC is not only an algorithm but also a condition (of the instance $\mathcal{P}'$ produced by the algorithm). We say that an instance $\mathcal{P}$ is SLAC-consistent if the SLAC algorithm, given in Figure 1, does not change the instance.

Multi-dimensional Fourier transform Let U_d be the set of d-th roots of unity, that is, $U_d = \{\lambda_k = e^{\frac{2\pi i}{d} k} \mid 0 \leq k < d\}$. The *Fourier transform* (FT) of a function $f : U_d^n \to U_d$ is defined, for $\mathbf{a} \in U_d^n$, as $FT(f, \mathbf{a}) = \sum_{\mathbf{b} \in U_d^n} f(\mathbf{b}) \lambda_1^{\mathbf{a} \cdot \mathbf{b}}$. Then it is not hard to see that $f(\mathbf{a}) = \sum_{\mathbf{b} \in U_d^n} FT(f, \mathbf{b}) \lambda_1^{\mathbf{a} \cdot \mathbf{b}}$, which gives rise to a representation of f by a polynomial

$$f(\overline{x}) = \sum_{\mathbf{b} \in U_d^n} FT(f, \mathbf{b}) \overline{x}^{\mathbf{b}'},$$

where $\mathbf{b}' = (k_1, \ldots, k_n)$ and $\mathbf{b}[j] = \lambda_{k_j}$. This representation is unique [46].

Linear operators and Hilbert spaces Let V be a complex vector space. A *linear operator* on V is a linear map from V to V. The identity linear operator on V is denoted by I. The linear operator that is identically 0 is denoted by 0. Let A and B be two linear operators. Then, their pointwise addition is denoted by $A + B$, their composition is denoted by AB, and the pointwise scaling of A by a scalar $c \in \mathbb{C}$ is denoted by cA. All of these are linear operators and thus we can plug linear operators in polynomials. Note that if $A_1, \ldots, A_n$ pairwise commute (i.e., $A_i A_j = A_j A_i$ for every $i, j \in [n]$) then $P(A_1, \ldots, A_n)$ is well defined. We denote by $\mathbb{C}[x_1, \ldots, x_n]$ the ring of polynomials with complex coefficients and commuting variables $x_1, \ldots, x_n$.

A *Hilbert space* is a complex vector space with an inner product whose norm induces a complete metric. All Hilbert spaces of finite dimension d are isomorphic to $\mathbb{C}^d$ with the standard complex inner product. Thus, after the choice of basis, linear operators on a d-dimensional Hilbert space can be identified with matrices in $\mathbb{C}^{d\times d}$, and operator composition becomes matrix multiplication. All Hilbert spaces in this paper will be of finite dimension; thus we will freely switch between the operator and matrix terminology.

A *diagonal* matrix has all off-diagonal entries equal to 0. For a matrix A, we denote by A^* its conjugate transpose. Recall that $(AB)^* = B^*A^*$. A matrix A is *unitary* if $A^*A = AA^* = I$, where I is the identity matrix. Two matrices A and B *commute* if $AB = BA$. A collection of matrices $A_1, \ldots, A_r$ *pairwise commute* if $A_iA_j = A_jA_i$ for all $i, j \in [r]$. A linear operator A is called *normal* if it commutes with A^*; i.e., $AA^* = A^*A$.

We will use the following form of the so-called *strong spectral theorem*.

Theorem 1 ([26]). *Let $A_1, \ldots, A_r$ be normal matrices. If $A_1, \ldots, A_r$ pairwise commute then there exists a unitary matrix U and diagonal matrices $E_1, \ldots, E_r$ such that $A_i = U^{-1}E_iU$ for every $i \in [r]$.*

3 Operator CSP

In order to relax the notion of satisfiability, we first consider CSPs on U_d for some d and represent CSPs via polynomials. Let Γ be a constraint language over U_d. Every constraint $\langle \mathbf{s}, R\rangle$ of an instance $\mathcal{P} = (V, U_d, \mathcal{C})$ of CSP(Γ) is represented by a polynomial $P_R(\mathbf{s})$ that represents the characteristic function f_R of R:

$$f_R(\mathbf{s}) = \begin{cases} \lambda_0, & \text{if } R(\mathbf{s}) \text{ is true,} \\ \lambda_1, & \text{otherwise.} \end{cases}$$

We note that our choice of the polynomial representation is somewhat arbitrary but other choices lead to the same results. (For instance, [2] studied the $d = 2$ case and used λ_1 to represent true.)

An operator A over a Hilbert space $\mathcal{H}$ is a *normal operator of order* d if A is normal and $A^d = I$. Operator assignment to the instance $\mathcal{P}$ over a Hilbert space $\mathcal{H}$ is a mapping that assigns to every variable from V an operator A_v over $\mathcal{H}$ such that

(a) A_v is a normal operator of order d for every $v \in V$;

(b) the operators $A_{v_1}, \ldots, A_{v_r}$ pairwise commute for every constraint $\langle (v_1, \ldots, v_r), R\rangle \in \mathcal{C}$.

We call an operator assignment $\{A_v\}$ *satisfying for* $\mathcal{P}$ if $P_R(A_{v_1}, \ldots, A_{v_r}) = I$ for every constraint $\langle (v_1, \ldots, v_r), R\rangle \in \mathcal{C}$. We say that CSP$(\Gamma)$ has a *satisfiability gap* if there are instances of CSP(Γ) that are not satisfiable over U_d but are satisfiable by an operator assignment over a finite-dimensional Hilbert space.

We shall repeatedly use the following simple lemma.

Lemma 2. *Let $x_1, \ldots, x_r$ be variables, let $Q_1, \ldots, Q_m, Q$ be polynomials in $\mathbb{C}[x_1, \ldots, x_r]$, and let $\mathcal{H}$ be a Hilbert space. If every assignment over U_d that satisfies the equations $Q_1 = \cdots = Q_m = 0$ also satisfies the equation $Q = 0$, then every fully commuting operator assignment over $\mathcal{H}$ that satisfies the equations $Q_1 = \cdots = Q_m = 0$ also satisfies the equation $Q = 0$.*

The proof of this lemma is very similar to that of the analogous claim in [2, Lemma 3], where it was established for $d = 2$. The main difference is to use $A^d = I$ rather than $A^2 = I$. For the sake of completeness we give the proof here.

Proof. Suppose that the conditions of the lemma hold and $A_1, \ldots, A_r$ are pairwise commuting operators such that the equations $Q_1 = \cdots = Q_m = 0$ are true when these matrices are assigned to $x_1, \ldots, x_r$. Then, since $A_1, \ldots, A_r$ are normal and commute, by Theorem 1 there is a unitary matrix U such that $E_i = UA_iU^{-1}$ is a diagonal matrix. Then, $E_i^d = I$, because $A_i^d = I$. Therefore, every diagonal entry $E_i(jj)$ belongs to U_d. For every equation Q_ℓ we have $Q_\ell(A_1, \ldots, A_r) = 0$ implying $Q_\ell(E_1, \ldots, E_r) = UQ_\ell(A_1, \ldots, A_r)U^{-1} = 0$. Since every E_i is diagonal, for every j it also holds $Q_\ell(E_1(jj), \ldots, E_r(jj)) = 0$. By the conditions of the lemma we also have $Q((E_1(jj), \ldots, E_r(jj)) = 0$, and $Q(A_1, \ldots, A_r) = U^{-1}Q(E_1, \ldots, E_r)U = 0$. ∎

4 Overview of results and techniques

In this section, we give an overview of how our main result is proved. All definitions and details are provided in the main part of the paper comprizing of Sections 5–8.

Bounded width One direction of our main result is the following.

Theorem 3. *Let Γ be a constraint language over U_d. If $\mathrm{CSP}(\Gamma)$ has bounded width then it has no satisfiability gap.*

The main idea behind the proof of Theorem 3 is to simulate the inference provided by SLAC by inference in polynomial equations. Let $\mathcal{S}$ be a SLAC-program solving $\mathrm{CSP}(\Gamma)$. In order to prove Theorem 3 we take an instance $\mathcal{P} = (V, U_d, \mathcal{C})$ of $\mathrm{CSP}(\Gamma)$ that is not SLAC-consistent, and therefore has no solution, as $\mathrm{CSP}(\Gamma)$ has bounded width, and prove that it also has no operator solution. We will prove it by contradiction, assuming $\mathcal{P}$ has an operator solution $\{A_v\}$ and then using the rules of a SLAC-program solving $\mathrm{CSP}(\Gamma)$ to infer stronger and stronger conditions on $\{A_v\}$ that eventually lead to a contradiction.

Recall that every rule of a SLAC-program has the form $(x \in S) \wedge R(x, y, z_1, \ldots, z_r)) \to (y \in S')$ for some variables $x, y \in V$, a constraint $\langle (x, y, z_1, \ldots, z_r), R \rangle$, and sets $S, S' \subseteq U_d$. Therefore, we need to show how to encode unary relations and rules of a SLAC-program through polynomials. For any $S \subseteq U_d$, the unary constraint restricting the domain of a variable x to the set S is represented by the polynomial

$$Dom_S(x) = \prod_{k \in S} (\lambda_k - x) + 1.^{5}$$

Similarly, the rule $(x \in S) \wedge R(x, y, z_1, \ldots, z_r)) \to (y \in S')$ of the SLAC program is represented by

$$Rule_{S,R,S'}(x, y, z_1, \ldots, z_r) = (Dom_{\overline{S}}(x) - 1)(P_R(x, y, z_1, \ldots, z_r) - \lambda_1)(Dom_{S'}(y) - 1).$$

To give an idea of how Theorem 3 is proved, we sketch the proof of the following.

Lemma 4. *Let $(v_1 \in S_1) \to \cdots \to (v_\ell \in S_\ell)$ be a derivation in the SLAC-program $\mathcal{S}$ and $\{A_v\}$ an operator assignment for $\mathcal{P}$. Then for each $i = 2, \ldots, \ell$*

$$(Dom_{\overline{S}_1}(A_{v_1}) - I)(Dom_{S_i}(A_{v_i}) - I) = 0.$$

Sketch. First, one shows via Lemma 2 that any operator assignment is a zero of $Rule_{S,R,S'}$ (cf. Lemma 11). This can be used to establish the claim for $i = 2$ (cf. Lemma 12). The rest of the proof is done by induction on i. In the induction case we have equations

$$(Dom_{\overline{S}_1}(A_{v_1}) - I)(Dom_{S_i}(A_{v_i}) - I) = 0, \tag{2}$$

and

$$(Dom_{\overline{S}_i}(A_{v_i}) - I)(Dom_{S_{i+1}}(A_{v_{i+1}}) - I) = 0. \tag{3}$$

The idea is to multiply (2) by $(Dom_{S_{i+1}}(A_{v_{i+1}}) - I)$ on the right, multiply (3) by $(Dom_{\overline{S}_1}(A_{v_1}) - I)$ on the left and subtract. The problem is, however, that

$$Dom_{S_i}(A_{v_i}) - Dom_{\overline{S}_i}(A_{v_i})$$

[5] This is not the representation of S as in the beginning of Section 3, as $Dom_S(a)$ is not necessarily equal to λ_1. However, it suffices for our purposes, because we only need the property that $Dom_S(a) = 1$ if and only if $a \in S$.

is not a constant polynomial. So, we also need to prove that any polynomial of the form

$$Dom_S(x) - Dom_{\overline{S}}(x)$$

is invertible modulo $x^d - 1$. The polynomial has the form

$$p(x) = \prod_{k\in S}(x-\lambda_k) - \prod_{k\notin S}(x-\lambda_k).$$

As is easily seen, if $S \neq U_d$ and $S \neq \varnothing$, then λ_k is not a root of $p(x)$ for any $\lambda_k \in U_d$. Therefore the greatest common divisor of $p(x)$ and $x^d - 1$ has degree 0, and hence there exists $q(x)$ such that

$$p(x)q(x) = c + r(x)(x^d - 1).$$

Thus before subtracting equations (2) and (3) we also multiply them by $q(A_{v_i})$. Then we get

$$\begin{aligned}
(Dom_{\overline{S}_1}(A_{v_1}) - I)(Dom_{S_i}(A_{v_i})q(A_{v_i}) - q(A_{v_i})Dom_{\overline{S}_i}(A_{v_i}))(Dom_{S_{i+1}}(A_{v_{i+1}}) - I) &= 0\\
(Dom_{\overline{S}_1}(A_{v_1}) - I)q(A_{v_i})(Dom_{S_i}(A_{v_i}) - Dom_{\overline{S}_i}(A_{v_i}))(Dom_{S_{i+1}}(A_{v_{i+1}}) - I) &= 0\\
c(Dom_{\overline{S}_1}(A_{v_1}) - I)(Dom_{S_{i+1}}(A_{v_{i+1}}) - I) &= 0.
\end{aligned}$$

The first transformation uses the fact that A_{v_i} commutes with itself, while the second one uses the property $A_{v_i}^d = I$. The result follows. $\square$

To complete the proof of Theorem 3 note that the lack of SLAC-consistency means that for some $v \in V$ the statement $(v = \lambda_k) \to (v \neq \lambda_k)$ can be derived from $\mathcal{P}$ for every $\lambda_k \in U_d$. By Lemma 4, for any operator assignment $\{A_w\}$ and any $\lambda_k \in U_d$ the operator A_v satisfies the equation $\prod_{j\neq k}(A_v - \lambda_j I) = 0$. By reverse induction on the size of S, one can show that for any $S \subseteq U_d$ these equations imply $\prod_{j\in S}(A_v - \lambda_j I) = 0$. Then for $S = \varnothing$ we get $I = 0$, witnessing that $\mathcal{P}$ has no satisfying operator assignment.

Magic squares modulo d Here we construct a CSP that is not satisfied over U_d but is satisfied over normal operators of order d. The construction is similar to the Mermin-Peres magic square [44, 45, 48].

Let let $g, f : \mathbb{Z}_d \to \mathbb{Z}_d$ be any function. By $A_{g,f}$ we denote the $d \times d$ matrix such that $A_{g,f}(x, y) \neq 0$ if and only if $y = x + g(x)$ (addition is modulo d), in which case $A_{g,f}(x, y) = \lambda_1^{f(x)}$. For instance,

$$A_{1,x} = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0\\ 0 & 0 & \lambda_1 & 0 & \cdots & 0\\ 0 & 0 & 0 & \lambda_1^2 & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & & \vdots\\ \lambda_1^{d-1} & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

We remark that the solutions to the Mermin-Peres magic square constructed from Pauli matrices

$$\sigma_x = \begin{pmatrix} 0 & 1\\ 1 & 0\end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i\\ i & 0\end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0\\ 0 & -1\end{pmatrix}$$

fit into this framework: $I \otimes \sigma_z = A_{0,2x}, \sigma_z \otimes I = A_{0,x-x^2}, \sigma_z \otimes \sigma_z = A_{0,3x-x^2}, \sigma_x \otimes I = A_{2,0}, I \otimes \sigma_x = A_{2x+1,0}, \sigma_x \otimes \sigma_x = A_{3+2x,0}, -\sigma_x \otimes \sigma_z = A_{2,2x+2}, -\sigma_z \otimes \sigma_x = A_{2x+1,x-x^2+2}, -\sigma_y \otimes \sigma_y = A_{3+2x,3x-x^2}$, respectively, assuming $\lambda_1 = i$. Also, $A_{0,0}$ is the identity matrix. In this section $g(x)$ is always a constant, and we use the notation $A_{p,f}$ for $g(x) = p \in \mathbb{Z}_d$.

$$\mathrm{CSP}(\Gamma) \leftrightarrow \mathrm{CSP}(\mathsf{core}(\Gamma)) \leftrightarrow \mathrm{CSP}(\mathsf{core}(\Gamma)^*) \leftarrow \mathrm{CSP}(\mathsf{core}(\Gamma)^*|_B) \leftarrow \mathrm{CSP}(\mathsf{core}(\Gamma)^*|_B/_\theta)$$

Figure 2: Reductions between CSPs corresponding to derivative languages.

As is easily seen, multiplication of matrices $A_{p,f}$ satisfies the following identity:

$$A_{p_1,f_1} \cdot A_{p_2,f_2} = A_{p_1+p_2,f_1(x)+f_2(x+p_1)}.$$

Since every row of $A_{p,f}$ contains exactly one nonzero entry, the same is true for any Kronecker product of such matrices. Therefore, a convenient way to represent elements of a product

$$A_{\overline{p},f} = A_{p_1,f_1} \otimes \cdots \otimes A_{p_k,f_k},$$

$f(x_1,\ldots,x_k) = f_1(x_1) + \cdots + f_k(x_k)$, is to specify the only nonzero element in row $(x_1,\ldots,x_k)$. We denote this element $A_{\overline{p},f}(x_1,\ldots,x_k)$. Note that this element is in the column $(x_1+p_1,\ldots,x_k+p_k)$ and is equal to $\lambda_1^{f(x_1,\ldots,x_k)}$.

Consider the following Kronecker products of $d+1$ matrices

$$\begin{aligned} B_\ell &= A_{(0,\ldots,0,-1,0,\ldots,0),-x_\ell}, \\ C_\ell &= A_{(1,\ldots,1),x_\ell-1}, \end{aligned}$$

where the nonzero entry for the matrix B_ℓ is in position ℓ. The magic square (or rather rectangle) for an odd $d > 2$ is then constructed as follows:

$$M_d = \begin{pmatrix} (B_1 \cdot C_1)^{-1} & \cdots & (B_{d+1} \cdot C_{d+1})^{-1} & B_1 \cdot C_1 \cdot \cdots \cdot B_{d+1} \cdot C_{d+1} \\ B_1 & \cdots & B_{d+1} & (B_1 \cdot \cdots \cdot B_{d+1})^{-1} \\ C_1 & \cdots & C_{d+1} & (C_1 \cdot \cdots \cdot C_{d+1})^{-1} \end{pmatrix}$$

In Section 6 we prove that the matrices above satisfy the required properties (cf. Proposition 14).

Unbounded width The second direction of our main result is the following.

Theorem 5. *Let Γ be a constraint language over U_d. If $\mathrm{CSP}(\Gamma)$ does not have bounded width then $\mathrm{CSP}(\Gamma)$ has a satisfiability gap.*

The overall idea of proving Theorem 5 is to "implement" the "magic rectangle" from above in $\mathrm{CSP}(\Gamma)$ provided $\mathrm{CSP}(\Gamma)$ does not have bounded width. We achieve this in several steps via a chain of reductions that lies at the heart of the algebraic approach to CSPs [12]. The reductions are shown in Figure 2. They are known to preserve satisfiability; we show that they also preserve satisfiability gaps.

The most basic reduction (used in the chain in several places) is that of primitive positive definitions. Let Γ be a constraint language over U_d, let r be an integer, and let $x_1,\ldots,x_r$ be variables ranging over the domain U_d. A primitive positive formula (*pp-formula*) over Γ is a formula of the form

$$\phi(x_1,\ldots,x_r) = \exists y_1 \cdots \exists y_s (R_1(\mathbf{z}_1) \wedge \cdots \wedge R_m(\mathbf{z}_m)), \tag{4}$$

where $R_i \in \Gamma$ is a relation over U_d of arity r_i and each $\mathbf{z}_i$ is an r_i-tuple of variables from $\{x_1,\ldots,x_r\} \cup \{y_1,\ldots,y_s\}$. A relation $R \subseteq U_d^r$ is primitive positive definable (*pp-definable*) from Γ if there exists a pp-formula $\phi(x_1,\ldots,x_r)$ over Γ such that R is equal to the set of models of ϕ, that is, the set of r-tuples $(a_1,\ldots,a_r) \in U_d^r$ that make the formula ϕ true over U_d if a_i is substituted for x_i in ϕ for every $i \in [r]$.

Theorem 6. *Let Γ be a constraint language over U_d and let R be pp-definable from Γ. Then, if* $\mathrm{CSP}(\Gamma \cup \{R\})$ *has a satisfiability gap then so does* $\mathrm{CSP}(\Gamma)$.

Let $R \subseteq U_d^r$ be a pp-definable formula over Γ via the pp-formula $\phi(x_1, \ldots, x_r)$ as in (4). Given an instance $\mathcal{P} \in \mathrm{CSP}(\Gamma \cup \{R\})$, one can turn it into an instance $\mathcal{P}' \in \mathrm{CSP}(\Gamma)$ that is equivalent to $\mathcal{P}$. Intuitively, each constraint $\langle \mathbf{u}, R\rangle$ of $\mathcal{P}$ is replaced with constraints from its pp-definition in (4) over fresh new variables. This construction is known as the *gadget construction* in the CSP literature and it is known that $\mathcal{P}$ has a solution over U_d if and only if $\mathcal{P}'$ has a solution over U_d [12, 5]. Thus, to prove Theorem 6, it suffices to show the following lemma, whose proof is a simple generalization of the special case for $d = 2$ proved in [2].

Lemma 7. *Let Γ be a constraint language over U_d and let R be pp-definable from Γ. Furthermore, let $\mathcal{P} \in \mathrm{CSP}(\Gamma \cup \{R\})$ and let $\mathcal{P}' \in \mathrm{CSP}(\Gamma)$ be the gadget construction replacing constraints involving R in $\mathcal{P}$. If there is a satisfying operator assignment for $\mathcal{P}$ then there is a satisfying operator assignment for $\mathcal{P}'$.*

If $\mathrm{CSP}(\Gamma \cup \{R\})$ has a satisfiability gap then there is an unsatisfiable instance $\mathcal{P} \in \mathrm{CSP}(\Gamma \cup \{R\})$ with a satisfying operator assignment. By [12] (cf. also [5]), $\mathcal{P}'$ is unsatisfiable. By Lemma 7, $\mathcal{P}'$ has a satisfying operator assignment. Hence $\mathcal{P}'$ establishes that $\mathrm{CSP}(\Gamma)$ has a satisfiability gap, and Theorem 6 is proved.

Pp-definitions are the starting point of the algebraic approach to CSPs [12] and suffice for dealing with Boolean CSPs, not only in [2] but also in all papers on Boolean (variants of) CSPs. For CSPs over larger domains, more tools are needed.

A constraint language Γ is a *core* language if all its endomorphisms are permutations; that is, Γ has no endomorphisms that are not automorphisms. There always exists an endomorphism ϱ of Γ such that $\varrho(\Gamma)$ is core and $\varrho \circ \varrho = \varrho$ [12]. We will denote this core language by $\mathsf{core}(\Gamma)$, as it (up to an isomorphism) does not depend on the choice of ϱ.

A constraint language Γ is called *idempotent* if it contains all the *constant* relations, that is, relations of the form $C_a = \{(a)\}$, $a \in U_d$. For an arbitrary language Γ over U_d we use $\Gamma^* = \Gamma \cup \{C_a \mid a \in U_d\}$. A unary relation (a set) $B \subseteq U_d$ pp-definable in Γ is called a *subalgebra* of Γ. For a subalgebra B we introduce the *restriction* $\Gamma|_B$ of Γ to B defined as $\Gamma|_B = \{R \cap B^{ar(R)} \mid R \in \Gamma\}$.

An equivalence relation θ pp-definable in Γ is said to be a *congruence* of Γ. The equivalence class of θ containing $a \in U_d$ will be denoted by $a/_\theta$, and the set of all equivalence classes, the *factor-set*, by $U_d/_\theta$. Congruences of a constraint language allow one to define a *factor-language* as follows. For a congruence θ of the language Γ the factor language $\Gamma/_\theta$ is the language over $U_d/_\theta$ given by $\Gamma/_\theta = \{R/_\theta \mid R \in \Gamma\}$, where $R/_\theta = \{(a_1/_\theta, \ldots, a_n/_\theta) \mid (a_1, \ldots, a_n) \in R\}$.

All the languages above are related to each other by the reducibility of the corresponding CSPs, as Figure 2 indicates (cf. Proposition 20). In Section 8, we show that all arrows in Figure 2 preserve satisfiability gaps. To relate these reductions with bounded width and magic squares we use the following result.

Proposition 8 ([11, 3, 5])**.** *For a constraint language Γ over U_d, $\mathrm{CSP}(\Gamma)$ does not have bounded width if and only there exists a language Δ pp-definable in Γ, a subalgebra B of $\mathsf{core}(\Delta)^*$, a congruence θ of $\mathsf{core}(\Delta)^*|_B$, and an Abelian group $\mathbb{A}$ of prime order p such that $\mathsf{core}(\Delta)^*|_B/_\theta$ contains relations $R_{3,a}, R_{p+2}$ for every $a \in \mathbb{A}$ given by $R_{3,a} = \{(x, y, z) \mid x + y + z = a\}$ and $R_{p+2} = \{(a_1, \ldots, a_{p+2}) \mid a_1 + \cdots + a_{p+2} = 0\}$.*[6]

To prove Theorem 5, suppose that $\mathrm{CSP}(\Gamma)$ does not have bounded width. By Proposition 8 there is a language Δ pp-definable in Γ, a subalgebra B of $\mathsf{core}(\Delta)^*$, a congruence θ of $\mathsf{core}(\Delta)^*|_B$, and an Abelian group $\mathbb{A}$ of prime order p such that $\mathsf{core}(\Delta)^*|_B/_\theta$ contains relations $R_{3,a}, R_{p+2}$ for every $a \in \mathbb{A}$. As reductions preserve satisfiability gaps, it suffices to prove that $\mathrm{CSP}(\Delta_p)$, where $\Delta_p = \{R_{3,a} \mid a \in \mathbb{A}\} \cup \{R_{p+2}\}$,

[6]The relations $R_{3,a}, R_{p+2}$ are chosen here because they are needed for our purpose. In fact, they can be replaced with any relations expressible by linear equations over $\mathbb{A}$.

has a satisfiability gap. For $p = 2$ the Mermin-Peres magic square from [44] provides a gap instance of $\mathrm{CSP}(\{R_{3,1}, R_{3,-1}\})$, and our construction above provides a gap instance for $\mathrm{CSP}(\Delta_p)$ for all odd $p > 2$. Note that in both cases the group $\mathbb{A}$ is the multiplicative group of roots of unity.

To give an idea of the gap-preservation proofs, we sketch how a satisfiability gap is preserved from subalgebras. Let Γ be a constraint language over the set U_d and let B be its subalgebra. We show that if $\mathrm{CSP}(\Gamma|_B)$ has a satisfiability gap then so does $\mathrm{CSP}(\Gamma)$.

Let $\Delta = \Gamma|_B$. Then by Theorem 6 we may assume $\Delta \subseteq \Gamma$ and $B \in \Gamma$. Let $e = |B|$ and $\pi : U_e \to U_d$ a bijection between U_e and B. Let $\mathcal{P} = (V, U_e, \mathcal{C})$ be a gap instance of $\mathrm{CSP}(\pi^{-1}(\Delta))$ and the instance $\mathcal{P}^\pi = (V, U_d, \mathcal{C}^\pi)$ constructed as follows: For every $\langle \mathbf{s}, R\rangle \in \mathcal{C}$ the instance $\mathcal{P}^\pi$ contains $\langle \mathbf{s}, \pi(R)\rangle$. As is easily seen, $\mathcal{P}^\pi$ has no classic solution. Therefore, it suffices to show that for any satisfying operator assignment $\{A_v \mid v \in V\}$ for $\mathcal{P}$, the assignment $C_v = \pi(A_v)$ is a satisfying operator assignment for $\mathcal{P}^\pi$.

By a technical lemma that shows that injective maps on finite sets that are interpolated by polynomials preserve normal operators that pairwise commute (cf. Lemma 22), the C_v's are normal, satisfy the condition $C_v^d = I$, and locally commute. For $\langle \mathbf{s}, R\rangle \in \mathcal{C}$, $\mathbf{s} = (v_1, \dots, v_k)$, let $f_R^\pi(x_1, \dots, x_k) = f_R(\pi^{-1}(x_1), \dots, \pi^{-1}(x_k))$. It can be shown that $\pi^{-1}(C_v) = A_v$, and therefore $f_R^\pi(C_{v_1}, \dots, C_{v_k}) = I$. For any $a_1, \dots, a_k \in U_d$, if $(a_1, \dots, a_k) \in \pi(R)$ then $a_1, \dots, a_k \in B$. Therefore, $f_R^\pi(a_1, \dots, a_k) = 1$ then $f_{\pi(R)}(a_1, \dots, a_k) = 1$. By Lemma 2 this implies $f_{\pi(R)}(C_{v_1}, \dots, C_{v_k}) = I$.

The other reductions use similar ideas, carefully relying on the spectral theorem given in Theorem 1 to simultaneously diagonalize the restriction of an operator assignment to the scope of a constraint, Lemma 2 that relates polynomial equations over U_d and operators, and the above mentioned result on preservation of operator assignments by certain polynomials (Lemma 22).

5 Bounded width and no gaps

In this section we prove the first direction of our main result.

Theorem 9. *Let Γ be a constraint language over U_d. If $\mathrm{CSP}(\Gamma)$ has bounded width then it has no satisfiability gap.*

The main idea behind the proof of Theorem 9 is to simulate the inference provided by SLAC by inference in polynomial equations. Let $\mathcal{S}$ be a SLAC-program solving $\mathrm{CSP}(\Gamma)$. In order to prove Theorem 9 we take an instance $\mathcal{P} = (V, U_d, \mathcal{C})$ of $\mathrm{CSP}(\Gamma)$ that is not SLAC-consistent, and therefore has no solution, as $\mathrm{CSP}(\Gamma)$ has bounded width, and prove that it also has no satisfying operator assignment. We will prove it by contradiction, assuming $\mathcal{P}$ has a satisfying operator assignment $\{A_v\}$ and then using the rules of a SLAC-program solving $\mathrm{CSP}(\Gamma)$ to infer stronger and stronger conditions on $\{A_v\}$ that eventually lead to a contradiction. We start with a series of lemmas that will help to express the restrictions on $\{A_v\}$.

The following lemma introduces a restriction that is satisfied by any operator assignment.

Lemma 10. *Let $\mathcal{P} = (V, U_d, \mathcal{C}) \in \mathrm{CSP}(\Gamma)$. For any operator assignment $\{A_v\}$ for $\mathcal{P}$ we have*

$$\prod_{k=0}^{d-1} (\lambda_k I - A_v) = 0.$$

Proof. Note that the equation $\prod_{k=0}^{d-1} (\lambda_k - x) = 0$ is true for any $x \in U_d$, that is, it follows from the empty set of equations. By Lemma 2, it also holds for any fully commuting operator assignment. However, as the equation contains only one variable any operator assignment is fully commuting, and the result follows. $\square$

Recall that every rule of a SLAC-program has the form $(x \in S) \wedge R(x, y, z_1, \dots, z_r)) \to (y \in S')$ for some variables $x, y \in V$, a constraint $\langle (x, y, z_1, \dots, z_r), R\rangle$, and sets $S, S' \subseteq U_d$. Therefore, we need to show how to encode unary relations and rules of a SLAC-program through polynomials. For any $S \subseteq U_d$, we represent the unary constraint restricting the domain of a variable x to the set S by the polynomial

$$Dom_S(x) = \prod_{k \in S} (\lambda_k - x) + 1.$$[7]

Similarly, the rule $(x \in S) \wedge R(x, y, z_1, \ldots, z_r)) \to (y \in S')$ of the SLAC program is represented by

$$Rule_{S,R,S'}(x, y, z_1, \ldots, z_r) = (Dom_{\overline{S}}(x) - 1)(P_R(x, y, z_1, \ldots, z_r) - \lambda_1)(Dom_{S'}(y) - 1).$$

As the next lemma shows, any operator assignment is a zero of $Rule_{S,R,S'}$.

Lemma 11. *Let $\mathcal{P} = (V, U_d, \mathcal{C}) \in$ CSP(Γ). For any operator assignment $\{A_v\}$ for $\mathcal{P}$ and any rule $(x \in S) \wedge R(x, y, z_1, \ldots, z_r)) \to (y \in S')$ of the SLAC program for* CSP(Γ) *we have*

$$\begin{aligned} & Rule_{S,R,S'}(A_x, A_y, A_{z_1}, \ldots, A_{z_r}) \\ = \ & (Dom_{\overline{S}}(A_x) - I)(P_R(A_x, A_y, A_{z_1}, \ldots, A_{z_r}) - \lambda_1 I)(Dom_{S'}(A_y) - I) = 0. \end{aligned}$$

Proof. Note that the equation $Rule_{S,R,S'}(x, y, z_1, \ldots, z_r) = 0$ is true for any $x, y, z_1, \ldots, z_r \in U_d$, that is, it follows from the empty set of equations. By Lemma 2, it also holds for any fully commuting operator assignment. However, as all the variables $x, y, z_1, \ldots, z_r$ belong to the scope of the same constraint, the operators $A_x, A_y, A_{z_1}, \ldots, A_{z_r}$ pairwise commute. The result follows. $\square$

Now, assume $\mathcal{P} = (V, U_d, \mathcal{C})$ is not SLAC-consistent and D_v denote the domain of $v \in V$ obtained after establishing SLAC-consistency. This means that for some $v \in V$ there is a derivation of $D_v = \varnothing$ using only facts $R(\mathbf{s})$ for $\langle \mathbf{s}, R \rangle \in \mathcal{C}$ and $T_B(x_i) \wedge R(x_1, \ldots, x_k) \to T_C(x_j)$ for the rules of the SLAC-program. Moreover, this derivation can be subdivided into sections of the form $(v = a) \to (v \neq a)$, each of which is linear. The latter condition means that each such section looks like a chain $(v = a) \to (v_1 \in S_1) \to \cdots \to (v_\ell \in S_\ell) \to (v \in D_v - \{a\})$, where each step is by a rule of the form $((v_i \in S_i) \wedge R_i(v_i, v_{i+1}, u_1, \ldots, u_r)) \to (v_{i+1} \in S_{i+1})$.

Lemma 12. *For any satisfying operator assignment $\{A_v\}$ for $\mathcal{P}$ and any rule $(x \in S) \wedge R(x, y, z_1, \ldots, z_r)) \to (y \in S')$ of the SLAC program for* CSP(Γ) *if $R(x, y, z_1, \ldots, z_r) \in \mathcal{C}$ then*

$$(Dom_{\overline{S}}(A_x) - I)(Dom_{S'}(A_y) - I) = I.$$

Proof. By Lemma 11, the equation

$$(Dom_{\overline{S}}(A_x) - I)(P_R(A_x, A_y, A_{z_1}, \ldots, A_{z_r}) - \lambda_1 I)(Dom_{S'}(A_y) - I) = 0$$

holds as well as the equation

$$P_R(A_x, A_y, A_{z_1}, \ldots, A_{z_r}) - I = 0.$$

Multiplying the latter one by $(Dom_{\overline{S}}(A_x) - I)$ on the left, and by $(Dom_{S'}(A_y) - I)$ on the right and subtracting it from the first equation we obtain

$$-(Dom_{\overline{S}}(A_x) - I)(1 - \lambda_1)I(Dom_{S'}(A_y) - I) = 0.$$

The result follows. $\square$

[7] This is not the representation of S as in the beginning of Section 3, as $Dom_S(a)$ is not necessarily equal to λ_1. However, it suffices for our purposes, because we only need the property that $Dom_S(a) = 1$ if and only if $a \in S$.

Lemma 13. *Let $(v_1 \in S_1) \to \cdots \to (v_\ell \in S_\ell)$ be a derivation in the SLAC-program $\mathcal{S}$ and $\{A_v\}$ a satisfying operator assignment for $\mathcal{P}$. Then for each $i = 2, \dots, \ell$*

$$(Dom_{\overline{S}_1}(A_{v_1}) - I)(Dom_{S_i}(A_{v_i}) - I) = 0.$$

Proof. We proceed by induction on i. For $i = 2$ the equation holds by Lemma 12. In the induction case we have equations

$$(Dom_{\overline{S}_1}(A_{v_1}) - I)(Dom_{S_i}(A_{v_i}) - I) = 0, \tag{5}$$

and

$$(Dom_{\overline{S}_i}(A_{v_i}) - I)(Dom_{S_{i+1}}(A_{v_{i+1}}) - I) = 0. \tag{6}$$

The idea is to multiply (5) by $(Dom_{S_{i+1}}(A_{v_{i+1}}) - I)$ on the right, multiply (6) by $(Dom_{\overline{S}_1}(A_{v_1}) - I)$ on the left and subtract. The problem is, however, that

$$Dom_{S_i}(A_{v_i}) - Dom_{\overline{S}_i}(A_{v_i})$$

is not a constant polynomial. So, we also need to prove that any polynomial of the form

$$Dom_S(x) - Dom_{\overline{S}}(x)$$

is invertible modulo $x^d - 1$. The polynomial has the form

$$p(x) = \prod_{k \in S}(x - \lambda_k) - \prod_{k \notin S}(x - \lambda_k).$$

As is easily seen, assuming that the product of an empty set of factors equals 1, λ_k is not a root of $p(x)$ for any $\lambda_k \in U_d$. Therefore the greatest common divisor of $p(x)$ and $x^d - 1$ has degree 0, and hence there exists $q(x)$ such that

$$p(x)q(x) = c + r(x)(x^d - 1).$$

Thus before subtracting equations (5) and (6) we also multiply them by $q(A_{v_i})$. Then we get

$$\begin{aligned}
(Dom_{\overline{S}_1}(A_{v_1}) - I)(Dom_{S_i}(A_{v_i})q(A_{v_i}) - q(A_{v_i})Dom_{\overline{S}_i}(A_{v_i}))(Dom_{S_{i+1}}(A_{v_{i+1}}) - I) &= 0 \\
(Dom_{\overline{S}_1}(A_{v_1}) - I)q(A_{v_i})(Dom_{S_i}(A_{v_i}) - Dom_{\overline{S}_i}(A_{v_i}))(Dom_{S_{i+1}}(A_{v_{i+1}}) - I) &= 0 \\
c(Dom_{\overline{S}_1}(A_{v_1}) - I)(Dom_{S_{i+1}}(A_{v_{i+1}}) - I) &= 0.
\end{aligned}$$

The first transformation uses the fact that A_{v_i} commutes with itself, while the second one uses the property $A_{v_i}^d = I$. The result follows. ☐

Proof of Theorem 9. To complete the proof of Theorem 9 note that the lack of SLAC-consistency means that for some $v \in V$ the statement $(v = \lambda_k) \to (v \neq \lambda_k)$ can be derived from $\mathcal{P}$ for every $\lambda_k \in U_d$. By Lemma 13, for any operator assignment $\{A_w\}$ and any $\lambda_k \in U_d$ the operator A_v satisfies the equation

$$\prod_{j \neq k}(A_v - \lambda_j I) = 0.$$

We show that for any $S \subseteq U_d$ these equations imply

$$\prod_{j \in S}(A_v - \lambda_j I) = 0.$$

Then for $S = \varnothing$ we get $I = 0$, witnessing that $\mathcal{P}$ has no satisfying operator assignment.

We proceed by (reverse) induction on the size of S. Suppose the statement is true for all sets of size r and let $S \subseteq U_d$ be such that $|S| = r-1$. Without loss fo generality, assume that $S = \{\lambda_0, \dots, \lambda_{r-1}\}$. Let $S_1 = S \cup \{\lambda_r\}, S_2 = S \cup \{\lambda_{r+1}\}$. Consider

$$\prod_{j \in S_1} (A_v - \lambda_j I) = 0 \tag{7}$$

and

$$\prod_{j \in S_2} (A_v - \lambda_j I) = 0. \tag{8}$$

Subtracting (8) from (7) we obtain

$$(A_v - \lambda_r I - A_v + \lambda_{r+1} I) \prod_{j=0}^{r-1} (A_v - \lambda_j I) = (\lambda_{r+1} - \lambda_r) I \prod_{j=0}^{r-1} (A_v - \lambda_j I) = 0,$$

implying the equation for S. $\square$

6 Magic square modulo d

Here we construct a CSP that is not satisfied over U_d but is satisfied over normal operators of order d. The construction is similar to the Mermin-Peres magic square [44, 45, 48].

Let let $g, f : \mathbb{Z}_d \to \mathbb{Z}_d$ be any function. By $A_{g,f}$ we denote the $d \times d$ matrix such that $A_{g,f}(x, y) \neq 0$ if and only if $y = x + g(x)$ (addition is modulo d), in which case $A_{g,f}(x, y) = \lambda_1^{f(x)}$. For instance,

$$A_{1,x} = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \lambda_1^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ \lambda_1^{d-1} & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

We remark that the solutions to the Mermin-Peres magic square constructed from Pauli matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

fit into this framework: $I \otimes \sigma_z = A_{0,2x}, \sigma_z \otimes I = A_{0,x-x^2}, \sigma_z \otimes \sigma_z = A_{0,3x-x^2}, \sigma_x \otimes I = A_{2,0}, I \otimes \sigma_x = A_{2x+1,0}, \sigma_x \otimes \sigma_x = A_{3+2x,0}, -\sigma_x \otimes \sigma_z = A_{2,2x+2}, -\sigma_z \otimes \sigma_x = A_{2x+1,x-x^2+2}, -\sigma_y \otimes \sigma_y = A_{3+2x,3x-x^2}$, respectively, assuming $\lambda_1 = i$. Also, $A_{0,0}$ is the identity matrix. In this section $g(x)$ is always a constant, and we use the notation $A_{p,f}$ for $g(x) = p \in \mathbb{Z}_d$.

As is easily seen, multiplication of matrices $A_{p,f}$ satisfies the following identity:

$$A_{p_1,f_1} \cdot A_{p_2,f_2} = A_{p_1+p_2, f_1(x)+f_2(x+p_1)}. \tag{9}$$

Indeed, for $x \in [d]$ the only nonzero element of the x-th row of $C = A_{p_1,f_1} \cdot A_{p_2,f_2}$ is the element

$$A_{p_1,f_1}(x, x+p_1) \cdot A_{p_2,f_2}(x+p_1, x+p_1+p_2) = \lambda^{f_1(x)} \cdot \lambda^{f_2(x+p_1)} = \lambda^{f_1(x)+f_2(x+p_1)}.$$

We will also use the mixed-product rule for Kronecker product $\otimes$.

$$(A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D).$$

Since every row of $A_{p,f}$ contains exactly one nonzero entry, the same is true for any Kronecker product of such matrices. Therefore, a convenient way to represent elements of a product

$$A_{\overline{p},f} = A_{p_1,f_1} \otimes \cdots \otimes A_{p_k,f_k},$$

$f(x_1, \ldots, x_k) = f_1(x_1) + \cdots + f_k(x_k)$, is to specify the only nonzero element in row $(x_1, \ldots, x_k)$. We denote this element $A_{\overline{p},f}(x_1, \ldots, x_k)$. Note that this element is in the column $(x_1 + p_1, \ldots, x_k + p_k)$ and is equal to $\lambda^{f(x_1,\ldots,x_k)}$.

Consider the following Kronecker products of $d + 1$ matrices

$$\begin{aligned} B_\ell &= A_{(0,\ldots,0,-1,0,\ldots,0),-x_\ell}, \\ C_\ell &= A_{(1,\ldots,1),x_\ell-1}, \end{aligned}$$

where the nonzero entry for the matrix B_ℓ is in position ℓ. The magic square (or rather rectangle) for an odd $d > 2$ is then constructed as follows:

$$M_d = \begin{pmatrix} (B_1 \cdot C_1)^{-1} & \cdots & (B_{d+1} \cdot C_{d+1})^{-1} & B_1 \cdot C_1 \cdot \cdots \cdot B_{d+1} \cdot C_{d+1} \\ B_1 & \cdots & B_{d+1} & (B_1 \cdot \cdots \cdot B_{d+1})^{-1} \\ C_1 & \cdots & C_{d+1} & (C_1 \cdot \cdots \cdot C_{d+1})^{-1} \end{pmatrix}$$

We now prove that the matrices above satisfy the required properties.

Proposition 14. *If d is odd then for the entries of the matrix M_d it holds that*

(a) every entry D of M_d is normal and satisfies $D^d = I$;

(b) the matrices in each row and each column of M_d commute;

(c) the product of the matrices in every row and every column of M_d except for the last column is the identity matrix;

(d) the product of matrices in the last column of M_d is a scalar matrix, but not the identity matrix.

Proof. We start with (b) for matrices B_ℓ, C_ℓ, which will also allow us to find the remaining matrices. By the mixed product rule and because $A_{0,0} = I$, in order to verify $B_\ell \cdot C_\ell = C_\ell \cdot B_\ell$ it suffices to show that

$$A_{-1,-x} \cdot A_{1,x-1} = A_{0,x-1-(x+1)} = A_{0,-2} = A_{1,x-1} \cdot A_{-1,-x}.$$

This also implies

$$B_\ell \cdot C_\ell = A_{(1,\ldots,1,0,1,\ldots,1),-2} \quad \text{and} \quad D_\ell = (B_\ell \cdot C_\ell)^{-1} = A_{(-1,\ldots,-1,0,-1,\ldots,-1),2}.$$

For $B_\ell \cdot B_j$ and $C_\ell \cdot C_j$, $\ell \neq j$, we have

$$\begin{aligned} B_\ell \cdot B_j(x_1, \ldots, x_d) &= B_\ell(x_1, \ldots, x_d) B_j(x_1, \ldots, x_{\ell-1}, x_\ell - 1, x_{\ell+1}, \ldots, x_d) \\ &= \lambda^{-x_\ell - x_j}, \\ &= B_j(x_1, \ldots, x_d) B_\ell(x_1, \ldots, x_{j-1}, x_j - 1, x_{j+1}, \ldots, x_d) \\ &= B_j \cdot B_\ell(x_1, \ldots, x_{d+1}); \\ C_\ell \cdot C_j(x_1, \ldots, x_d) &= C_\ell(x_1, \ldots, x_d) C_j(x_1 + 1, \ldots, x_d + 1) \\ &= \lambda^{x_\ell - 1 + (x_j + 1) - 1}, \\ &= \lambda^{x_j - 1 + (x_\ell + 1) - 1}, \\ &= C_j \cdot C_\ell(x_1, \ldots, x_d). \end{aligned}$$

This also implies that

$$B_\ell \cdot B_j = A_{(0,\dots,0,-1,0,\dots,0,-1,0,\dots,0),-x_\ell-x_j}, \quad C_\ell \cdot C_j = A_{(2,\dots,2),x_\ell+x_j-1}.$$

Therefore,

$$\begin{aligned} B_1 \cdot \dots \cdot B_{d+1} &= A_{(-1,\dots,-1),-(x_1+\dots+x_{d+1})} \\ C_1 \cdot \dots \cdot C_{d+1} &= A_{(d+1,\dots,d+1),x_1-1+(x_2+1)-1+\dots+(x_{d+1}+d)-1} \\ &= A_{(d+1,\dots,d+1),x_1+x_2+\dots+x_{d+1}-1+\frac{d(d+1)}{2}} \\ &= A_{(1,\dots,1),x_1+x_2+\dots+x_{d+1}-1}, \end{aligned}$$

which implies

$$\begin{aligned} F = (B_1 \cdot \dots \cdot B_{d+1})^{-1} &= A_{(1,\dots,1),(x_1+\dots+x_{d+1})+1} \\ G = (C_1 \cdot \dots \cdot C_{d+1})^{-1} &= A_{(-1,\dots,-1),-(x_1+x_2+\dots+x_{d+1})}. \end{aligned}$$

Finally,

$$\begin{aligned} E = B_1 \cdot C_1 \cdot \dots \cdot B_{d+1} \cdot C_{d+1} &= A_{(0,1,\dots,1),-2} \cdot \dots \cdot A_{(1,\dots,1,0),-2} \\ &= A_{(d,\dots,d),-2(d+1)} \\ &= A_{(0,\dots,0),-2}. \end{aligned}$$

(a) By Fuglede's theorem if two normal operators commute, their product is normal. By what is proved above it suffices to show that B_ℓ, C_ℓ are normal. Indeed, the only case that does not follows from this is the matrix E, but this matrix is diagonal. Then by the mixed-product rule and the equalities $(A^*)^{-1} = (A^{-1})^*$, $(A \otimes B)^* = A^* \otimes B^*$ it suffices to check that matrices $A_{0,0}, A_{-1,-x}, A_{1,x-1}$ are normal. Note first that $(A_{p,f})^* = A_{-p,-f}$. Then we have that $A_{0,0}$ is the identity matrix,

$$\begin{aligned} A_{-1,-x} \cdot (A_{-1,-x})^* &= A_{-1,-x} \cdot A_{1,x} = A_{0,-x+x-1} = A_{0,-1} \\ &= A_{0,x-(x+1)} = A_{1,x} \cdot A_{-1,-x} = (A_{-1,-x})^* \cdot A_{-1,-x}; \\ A_{1,x-1} \cdot (A_{1,x-1})^* &= A_{1,x-1} \cdot A_{-1,-x+1} = A_{0,x-1+(-(x+1)+1)} = A_{0,-1} \\ &= A_{0,-x+1+((x-1)-1)} = A_{-1,-x+1} \cdot A_{1,x-1} = (A_{1,x-1})^* \cdot A_{1,x-1}. \end{aligned}$$

Next, we need to find $B_\ell^d, C_\ell^d, D_\ell^d, E^d, F^d, G^d$:

$$\begin{aligned} B_\ell^d &= (A_{(0,\dots,0,-1,0,\dots,0),-x_\ell})^d \\ &= A_{(0,\dots,0),-x_\ell+(-x_\ell-1)+\dots+(-x_\ell-(d-1))} \\ &= A_{(0,\dots,0),-\frac{(d-1)d}{2}} \\ &= A_{(0,\dots,0),0}; \\ C_i^d &= (A_{(1,\dots,1),x_\ell})^d \\ &= A_{(0,\dots,0),x_\ell+(x_\ell+1)+\dots+(x_\ell+(d-1))} \\ &= A_{(0,\dots,0),\frac{(d-1)d}{2}} \\ &= A_{(0,\dots,0),0}; \\ D_i^d &= (A_{(-1,\dots,-1,0,-1,\dots,-1),2})^d = A_{(0,\dots,0),0}; \\ E^d &= (A_{(0,\dots,0),-2})^d = A_{(0,\dots,0),0}. \end{aligned}$$

Since B_ℓ, B_j commute, as do C_i, C_j, we have

$$F^d = (B_1 \cdot \ldots \cdot B_{d+1})^{-d} = B_1^{-d} \cdot \ldots \cdot B_{d+1}^{-d} = A_{(0,\ldots,0),0};$$
$$G^d = (C_1 \cdot \ldots \cdot C_{d+1})^{-d} = C_1^{-d} \cdot \ldots \cdot C_{d+1}^{-d} = A_{(0,\ldots,0),0}.$$

(b) We have verified that B_i, C_i commute, and so do B_i, B_j, and C_i, C_j. This means that D_i commutes with B_i, C_i, and also F commutes with each of $B_1, \ldots, B_{d+1}$, and G commutes with each of $C_1, \ldots, C_{d+1}$. Matrices D_i and E are scalar, so they commute with each other, and E commutes with F, G. It remains to verify that F, G commute, as well. We have

$$\begin{aligned} F \cdot G &= A_{(1,\ldots,1),(x_1+\cdots+x_{d+1})+1} \cdot A_{(-1,\ldots,-1),-(x_1+x_2+\cdots+x_{d+1})} \\ &= A_{(0,\ldots,0),(x_1+\cdots+x_{d+1})+1-((x_1+1)+(x_2+1)+\cdots+(x_{d+1}+1))} \\ &= A_{(0,\ldots,0),-(d+1)+1} \\ &= A_{(0,\ldots,0),0}. \end{aligned}$$

On the other hand,

$$\begin{aligned} G \cdot F &= A_{(-1,\ldots,-1),-(x_1+\cdots+x_{d+1})} \cdot A_{(1,\ldots,1),(x_1+x_2+\cdots+x_{d+1})+1} \\ &= A_{(0,\ldots,0),-(x_1+\cdots+x_{d+1})+((x_1-1)+(x_2-1)+\cdots+(x_{d+1}-1))+1} \\ &= A_{(0,\ldots,0),1-(d+1)} \\ &= A_{(0,\ldots,0),0}. \end{aligned}$$

(c) follows from the construction.

(d) By (b) we have that

$$E \cdot F \cdot G = A_{(0,\ldots,0),-2} \cdot A_{(0,\ldots,0),0} = A_{(0,\ldots,0),-2},$$

as required. □

7 Reductions through pp-definitions

In this section we prove that the so-called primitive positive definitions, a key tool in the algebraic approach to CSPs [12], not only give rise to (polynomial-time) reductions that preserve satisfiability over U_d but also preserve satisfiability over operators. This was established for the special case of Boolean domains (i.e., for $d = 2$) in [2] and the same idea works for larger domains. We will need this result later in Section 8 to prove that CSPs of unbounded width admit a satisfiability gap.

Let Γ be a constraint language over U_d, let r be an integer, and let $x_1, \ldots, x_r$ be variables ranging over the domain U_d. A primitive positive formula (*pp-formula*) over Γ is a formula of the form

$$\phi(x_1, \ldots, x_r) = \exists y_1 \cdots \exists y_s (R_1(\mathbf{z}_1) \wedge \cdots \wedge R_m(\mathbf{z}_m)), \tag{10}$$

where $R_i \in \Gamma$ is a relation over U_d of arity r_i and each $\mathbf{z}_i$ is an r_i-tuple of variables from $\{x_1, \ldots, x_r\} \cup \{y_1, \ldots, y_s\}$. A relation $R \subseteq U_d^r$ is primitive positive definable (*pp-definable*) from Γ if there exists a pp-formula $\phi(x_1, \ldots, x_r)$ over Γ such that R is equal to the set of models of ϕ, that is, the set of r-tuples $(a_1, \ldots, a_r) \in U_d^r$ that make the formula ϕ true over U_d if a_i is substituted for x_i in ϕ for every $i \in [r]$.

Our goal in this section is to prove the following result.

Theorem 15. *Let Γ be a constraint language over U_d and let R be pp-definable from Γ. Then, if* $\mathrm{CSP}(\Gamma \cup \{R\})$ *has a satisfiability gap then so does* $\mathrm{CSP}(\Gamma)$.

Let $R \subseteq U_d^r$ be a pp-definable formula over Γ via the pp-formula $\phi(x_1, \ldots, x_r)$ as in (10). Given an instance $\mathcal{P} \in \mathrm{CSP}(\Gamma \cup \{R\})$ we describe a construction of an instance $\mathcal{P}' \in \mathrm{CSP}(\Gamma)$ that is, in some sense, equivalent to $\mathcal{P}$. We start with the instance $\mathcal{P}$. For every constraint $\langle \mathbf{u}, R \rangle$ of $\mathcal{P}$ with $\mathbf{u} = (u_1, \ldots, u_r)$, we introduce s fresh new variables $t_1, \ldots, t_s$ for the quantified variables in (10); furthermore, we replace $\langle \mathbf{u}, R \rangle$ by m constraints $\langle \mathbf{w}_i, R_i \rangle$, $i \in [m]$, where $\mathbf{w}_i$ is the tuple of variables obtained from $\mathbf{z}_i$ in (10) by replacing x_j by u_j, $j \in [r]$, and by replacing y_j by t_j, $j \in [s]$. The collection of variables $u_1, \ldots, u_r, t_1, \ldots, t_s$ is called the *block* of the constraint $\langle \mathbf{u}, R \rangle$ in $\mathcal{P}'$. This construction is known as the *gadget construction* in the CSP literature and it is known that $\mathcal{P}$ has a solution over U_d if and only if $\mathcal{P}'$ has a solution over U_d [12, 5]. Thus, in order to prove Theorem 15, it suffices to show the following lemma.

Lemma 16. *Let Γ be a constraint language over U_d and let R be pp-definable from Γ. Furthermore, let $\mathcal{P} \in \mathrm{CSP}(\Gamma \cup \{R\})$ and let $\mathcal{P}' \in \mathrm{CSP}(\Gamma)$ be the gadget construction replacing constraints involving R in $\mathcal{P}$. If there is a satisfying operator assignment for $\mathcal{P}$ then there is a satisfying operator assignment for $\mathcal{P}'$.*

Indeed, if $\mathrm{CSP}(\Gamma \cup \{R\})$ has a satisfiability gap then there is an unsatisfiable instance $\mathcal{P} \in \mathrm{CSP}(\Gamma \cup \{R\})$ that has a satisfying operator assignment. By the results in [12] (cf. also [5]), $\mathcal{P}'$ is unsatisfiable. By Lemma 16, $\mathcal{P}'$ has a satisfying operator assignment. Hence $\mathcal{P}'$ establishes that $\mathrm{CSP}(\Gamma)$ has a satisfiability gap, as required to prove Theorem 15.

We will frequently use (below and also in Section 8) the following observations. We give a proof of them for the sake of completeness.

Lemma 17. *Let f be a polynomial and A, B operators in a finite-dimensional Hilbert space.*

(1) If A, B commute, then so do $f(A), f(B)$.
(2) If A is normal, then so is $f(A)$.
(3) If U is a unitary operator, then $Uf(A)U^{-1} = f(UAU^{-1})$.

Proof. Let $f(x) = \sum_{i=0}^{k} \alpha_i x^i$.

(1) We have

$$f(A)f(B) = \left(\sum_{i=0}^{k} \alpha_i A^i\right)\left(\sum_{i=0}^{k} \alpha_i B^i\right) = \sum_{i,j=0}^{k} \alpha_i \alpha_j A^i B^j$$
$$= \sum_{i,j=0}^{k} \alpha_j \alpha_i B^j A^i = \left(\sum_{i=0}^{k} \alpha_i B^i\right)\left(\sum_{i=0}^{k} \alpha_i A^i\right) = f(B)f(A).$$

(2) The operator A is normal if it commutes with its conjugate transpose A^*. Since for any operators B, C it holds that $(B+C)^* = B^* + C^*$ and $(B^k)^* = (B^*)^k$, we obtain $f(A^*) = f(A)^*$. The result then follows from item (1).
(3) We have

$$Uf(A)U^{-1} = U\left(\sum_{i=0}^{k} \alpha_i A^i\right)U^{-1} = \sum_{i=0}^{k} U\alpha_i A^i U^{-1} = \sum_{i=0}^{k} \alpha_i (UAU^{-1})(UAU^{-1})\ldots(UAU^{-1})$$
$$= \sum_{i=0}^{k} \alpha_i (UAU^{-1})^i = f(UAU^{-1}).$$

□

Proof of Lemma 16. Let $\mathcal{P} = (V, U_d, \mathcal{C})$ and let $\{A_v\}_{v\in V}$ be an operator assignment that is satisfying for $\mathcal{P}$ over a finite-dimensional Hilbert space $\mathcal{H}$. We may assume that $\mathcal{H} = \mathbb{C}^p$ for some positive integer p and thus $\{A_v\}_{v\in V}$ are $p \times p$ matrices. We will construct an operator assignment that is satisfying for $\mathcal{P}'$; it will be an operator assignment over the same space and an extension of the original assignment.

Given a constraint $\langle (u_1, \ldots, u_r), R\rangle \in \mathcal{C}$, the operators $\{A_{u_i}\}_{1\leq i\leq r}$ pairwise commute by assumption. Since they are also normal, by Theorem 1 there is a unitary matrix U such that $E_i = UA_{u_i}U^{-1}$ is a diagonal matrix for each $i \in [r]$. Since $A_{u_i}^d = I$, we have $E_{u_i}^d = I$. Thus every diagonal entry $E_{u_i}(jj)$ belongs to U_d. Since $P_R(A_{u_1}, \ldots, A_{u_r}) = I$, we have $P_R(E_{u_1}, \ldots, E_{u_r}) = UP_R(A_{u_1}, \ldots, A_{u_r})U^{-1} = I$ by Lemma 17. As E_{u_i} is a diagonal matrix, we have that $P_R(E_{u_1}(jj), \ldots, E_{u_r}(jj)) = 1 = \lambda_0$ for every $j \in [p]$. Thus the tuple $(E_{u_1}(jj), \ldots, E_{u_r}(jj)) \in R$ for every $j \in [p]$. For each $j \in [p]$, let $b(j) = (b_1(j), \ldots, b_s(j)) \in U_d^s$ be a tuple of witnesses to the existentially quantified variables $y_1, \ldots, y_s$ in the formula pp-defining R in (10) when $E_{u_i}(jj)$ is substituted for x_i. For $i \in [s]$, let B_i be the diagonal matrix with $B_i(jj) = b_i(j)$ for every $j \in [d]$. Now let $C_i = U^{-1}B_iU$. Since U is unitary we have $U^{-1} = U^*$ and thus each C_i is normal: $C_iC_i^* = U^*B_iU(U^*B_iU)^* = U^*B_iUU^*B_i^*U = U^*B_iB_i^*U = U^*B_i^*B_iU = U^*B_i^*UU^*B_iU = (U^*B_iU)^*U^*B_iU = C_i^*C_i$. Since $b_i(j) \in U_d$, we have $C_i^d = I$. Also $E_1, \ldots, E_r, B_1, \ldots, B_s$ pairwise commute since they are all diagonal matrices. Hence, $A_i, \ldots, A_r, C_1, \ldots, C_s$ also pairwise commute since they all are simultaneously similar via U. As each conjunct in (10) is satisfied by the assignment sending x_i to $E_{u_i}(jj)$ and y_i to $b_i(j)$ for all $j \in [d]$, we can conclude that the matrices that are assigned to the variables in the conjuncts make the corresponding polynomial evaluate to I. But this means that the assignment to the variables in the block of the constraint $\langle (u_1, \ldots, u_r), R\rangle$ makes a satisfying operator assignment for the constraint of $\mathcal{P}'$ that has come from the conjunct. As different constraints involving R in $\mathcal{P}$ produce their own sets of fresh variables, the operator assignments do not affect each other. $\square$

We do not know whether the converse of Lemma 16 holds; this is not known even in the case of $d = 2$ [2]. The obvious idea would be to take the restriction of the operator assignment that is satisfying for $\mathcal{P}'$ but it is not clear why this should be satisfying for $\mathcal{P}$, because there is no guarantee that the operators assigned to variable in the scope of a constraint of $\mathcal{P}$ of the form $\langle \mathrm{s}, R\rangle$ commute. However, under a slight technical assumption on Γ — namely, that it includes the full binary relation[8] on U_d — one can enforce commutativity within a constraint scope and thus project an operator assignment. While this result is not needed for our main result, we include it for completeness.

Let $R_T = U_d^2$ denote the full binary relation on U_d. For an instance $\mathcal{P}'$ as defined above (and in the statement of Lemma 16), we denote by $\mathcal{P}''$ the instance obtained from $\mathcal{P}'$ by adding, for every constraint $\langle (u_1, \ldots, u_r), R\rangle$ of $\mathcal{P}$, constraints of the form $\langle (u_i, u_j), R_T\rangle$. for every $i \neq j \in [r]$.

Lemma 18. *Let Γ be a constraint language over U_d with $R_T \in \Gamma$ and let R be pp-definable from Γ. Furthermore, let $\mathcal{P} \in \mathrm{CSP}(\Gamma \cup \{R\})$ and let $\mathcal{P}'' \in \mathrm{CSP}(\Gamma)$ be defined as above. Then, we have the following:*

(1) If there is a satisfying operator assignment for $\mathcal{P}$ then there is a satisfying operator assignment for $\mathcal{P}''$.

(2) If there is a satisfying operator assignment for $\mathcal{P}''$ then there is a satisfying operator assignment for $\mathcal{P}$.

Proof. (1) follows from Lemma 16: The satisfying operator assignment for $\mathcal{P}'$ constructed in the proof of Lemma 16 is also a satisfying operator assignment or $\mathcal{P}''$. Indeed, the constraints already present in $\mathcal{P}'$ are by assumption satisfied in $\mathcal{P}''$. Regarding the extra constraints in $\mathcal{P}''$ not present in $\mathcal{P}'$, each such constraint involves the R_T relation and the two variables in the scope of the constraint come from the block of some constraint in $\mathcal{P}'$. The proof of Lemma 16 established that the constructed operators pairwise commute on these variables. Also, as $P_{R_T}(a, b) = 1$ for any $a, b \in U_d$, the polynomial constraints are satisfied as they evaluate to I.

[8]This is a special case of the so-called *commutativity gadget* [2].

For (2), take a satisfying operator assignment for $\mathcal{P}''$ and consider its restriction $\{A_v\}_{v\in V}$ onto the variables of $\mathcal{P}$. Any two operators whose variables appear within the scope of some constraint of $\mathcal{P}$ necessarily commute since the two variables are in the block of some constraint in $\mathcal{P}''$. It remains to show that the polynomial constraints of $\mathcal{P}$ are satisfied, that is, that $P_R(A_{u_1},\dots,A_{u_r}) = I$ for every constraint $\langle(u_1,\dots,u_r),R\rangle$ of $\mathcal{P}$. For this, we use Lemma 2. Let ϕ be the formula pp-defining R as in (10). We define several polynomials over variables $x_1,\dots,x_r,y_1,\dots,y_s$ that correspond to the variables in (10). For every $i\in[m]$, let Q_i be the polynomial $P_{R_i}(\mathbf{z}_i)-1$ so that the equation $Q_i = 0$ ensures $P_{R_i}(\mathbf{z}_i) = 1$, where P_{R_i} is the characteristic polynomial of R_i, and $\mathbf{z}_i$ is the tuple of variables from $x_1,\dots,x_r,y_1,\dots,y_s$ that correspond to the variables of the same name that appear in the conjunct $R_i(\mathbf{z}_i)$ of (10). Let Q be the polynomial $P_R(x_1,\dots,x_r)-1$, where P_R is the characteristic polynomial of R. By the construction of the polynomials and the choice of ϕ, every assignment over U_d that satisfies all equations $Q_1 = \cdots = Q_m = 0$ also satisfies $Q = 0$. By Lemma 2, we get $P_R(A_{u_1},\dots,A_{u_r}) - I = 0$, as required. ☐

8 Unbounded width and gaps

In this section we prove the second direction of our main result.

Theorem 19. *Let Γ be a constraint language over U_d. If $\mathrm{CSP}(\Gamma)$ does not have bounded width then $\mathrm{CSP}(\Gamma)$ has a satisfiability gap.*

The overall idea of proving Theorem 19 is to "implement" the "magic rectangle" from Section 6 in $\mathrm{CSP}(\Gamma)$ provided $\mathrm{CSP}(\Gamma)$ does not have bounded width. We achieve this in several steps via a chain of reductions that has been used since the inception of the algebraic method to the CSP [12]. While more direct constructions have been developed later, see, e.g., [5], we find this original approach to be better suited for operator CSPs.

8.1 Bounded width, Abelian groups, and the magic square

We start by introducing several definitions.

A constraint language Γ over U_d is said to be a *core language* if its every endomorphism is a permutation. This term comes from finite model theory where it is used for relational structures that do not have endomorphisms (homomorphisms to themselves) that are not automorphisms. Such structures, and therefore languages, have a number of useful properties that we will exploit later. The standard way to convert a constraint language Γ into a core language is to repeat the following procedure until the resulting language is a core language: Pick an endomorphism ϱ of Γ that is not a permutation and set

$$\varrho(\Gamma) = \{\varrho(R) \mid R\in\Gamma\},\quad \text{where } \varrho(R) = \{(\varrho(a_1),\dots,\varrho(a_n)) \mid (a_1,\dots,a_n)\in R\}.$$

There always exists an endomorphism ϱ of Γ such that $\varrho(\Gamma)$ is core [12, 5] and ϱ is *idempotent*, that is, $\varrho\circ\varrho = \varrho$. We will denote this core language by $\mathsf{core}(\Gamma)$, as it (up to an isomorphism) does not depend on the choice of ϱ. Note that the fact that ϱ is idempotent implies that it acts as identity on its image.

The language Γ is called *idempotent* if it contains all the *constant* relations, that is, relations of the form $C_a = \{(a)\}$, $a\in U_d$. For an arbitrary language Γ over U_d we use $\Gamma^* = \Gamma\cup\{C_a \mid a\in U_d\}$. A unary relation (a set) $B\subseteq U_d$ pp-definable in Γ is called a *subalgebra* of Γ. For a subalgebra B we introduce the *restriction* $\Gamma|_B$ of Γ to B defined as follows

$$\Gamma|_B = \{R\cap B^{ar(R)} \mid R\in\Gamma\}.$$

An equivalence relation θ pp-definable in Γ is said to be a *congruence* of Γ. The equivalence class of θ containing $a\in U_d$ will be denoted by $a/_\theta$, and the set of all equivalence classes, the *factor-set*, by $U_d/_\theta$.

$$\mathrm{CSP}(\Gamma) \leftrightarrow \mathrm{CSP}(\mathsf{core}(\Gamma)) \leftrightarrow \mathrm{CSP}(\mathsf{core}(\Gamma)^*) \leftarrow \mathrm{CSP}(\mathsf{core}(\Gamma)^*|_B) \leftarrow \mathrm{CSP}(\mathsf{core}(\Gamma)^*|_B/_\theta)$$

Figure 3: Reductions between CSPs corresponding to derivative languages

Congruences of a constraint language allow one to define a *factor-language* as follows. For a congruence θ of the language Γ the factor language $\Gamma/_\theta$ is the language over $U_d/_\theta$ given by

$$\Gamma/_\theta = \{R/_\theta \mid R \in \Gamma\}, \quad \text{where } R/_\theta = \{(a_1/_\theta, \dots, a_n/_\theta) \mid (a_1, \dots, a_n) \in R\}.$$

In order to fit core languages, subalgebras, and factor-languages in our framework where the domain is the set of roots of unity, we let $e = |\varrho(U_d)|$, $e = |B|$ or $e = |U_d/_\theta|$, respectively, arbitrarily choose a bijection $\pi : \varrho(U_d) \to U_e$, $\pi : B \to U_e$, and $\pi : U_d/_\theta \to U_e$, and replace $\varrho(\Gamma)$, $\Gamma|_B$, and $\Gamma/_\theta$ with $\pi(\varrho(\Gamma))$, $\pi(\Gamma|_B)$, and $\pi(\Gamma/_\theta)$, respectively.

All the languages above are connected with each other in terms of the reducibility of the corresponding CSPs, as Figure 3 and the following statements indicate.

Proposition 20 ([12, 5]). *Let Γ be a constraint language over U_d. Then*

(1) For any endomorphism ϱ of Γ, $\mathrm{CSP}(\Gamma)$ and $\mathrm{CSP}(\mathsf{core}(\Gamma))$ are polynomial-time interreducible.

(2) If Γ is a core language, $\mathrm{CSP}(\Gamma)$ and $\mathrm{CSP}(\Gamma^*)$ are polynomial-time interreducible.

(3) If B is a subalgebra of Γ then $\mathrm{CSP}(\Gamma|_B)$ is polynomial-time reducible to $\mathrm{CSP}(\Gamma)$.

(4) If θ is a congruence of Γ then $\mathrm{CSP}(\Gamma/_\theta)$ is polynomial-time reducible to $\mathrm{CSP}(\Gamma)$.

Finally, to relate the reductions above with bounded width and the magic square we apply the following result that can be extracted from the known results on the algebraic approach to CSPs [11, 3, 5].

Proposition 21 ([11, 3, 5]). *For a constraint language Γ over U_d, $\mathrm{CSP}(\Gamma)$ does not have bounded width if and only there exists a language Δ pp-definable in Γ, a subalgebra B of $\mathsf{core}(\Delta)^*$, a congruence θ of $\mathsf{core}(\Delta)^*|_B$, and an Abelian group $\mathbb{A}$ of prime order p such that $\mathsf{core}(\Delta)^*|_B/_\theta$ contains relations $R_{3,a}, R_{p+2}$ for every $a \in \mathbb{A}$ given by*

$$R_{3,a} = \{(x, y, z) \mid x + y + z = a\}, \quad \textit{and} \quad R_{p+2} = \{(a_1, \dots, a_{p+2}) \mid a_1 + \dots + a_{p+2} = 0\}.^{9}$$

8.2 Proof of Theorem 19

In this section we prove that the connections shown in Figure 3 hold in terms of satisfiability gaps. We start with a helpful observation. As any mapping on a finite set of complex numbers can be interpolated by a polynomial, we may apply such mappings to operators as well (we assume that such an interpolating polynomial is of the lowest degree possible, and so is unique). A polynomial ϱ is said to *interpolate* a set $B \subseteq U_d$ if $\varrho(\lambda) = 1$ if $\lambda \in B$ and $\varrho(\lambda) = 0$ if $\lambda \in U_d - B$.

Lemma 22. *Let $d, e \in \mathbb{N}$, $e \le d$. Let $\pi : U_e \to U_d$ be an injective mapping and ϱ a unary polynomial that interpolates $B = \mathsf{Im}\ (\pi)$. Let $A_1, \dots, A_k$ be pairwise commuting normal operators of order e. Then $C_i = \pi(A_i)$, $i \in [k]$, are pairwise commuting normal operators of order d, and $\varrho(C_i) = I$. Conversely, let $C_1, \dots, C_k$ be pairwise commuting normal operators of order d such that $\varrho(C_i) = I$. Then for*

[9] The relations $R_{3,a}, R_{p+2}$ are chosen here because they are needed for our purpose. In fact, they can be replaced with any relations expressible by linear equations over $\mathbb{A}$.

$A_i = \pi^{-1}(C_i)$, $i \in [k]$, it holds that the A_i's are pairwise commuting normal operators of order e, and the eigenvalues of the C_i's belong to B.

Proof. That the C_i's are normal and pairwise commute follow from Lemma 17. Let the A_i's be ℓ-dimensional and U a unitary operator guaranteed by Theorem 1 such that UA_iU^{-1} is diagonal for all $i \in [k]$, and let

$$UA_iU^{-1} = \begin{pmatrix} \mu_{i1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mu_{i\ell} \end{pmatrix}.$$

Then,

$$\begin{aligned} C_i^d &= (\pi(A_i))^d = U^{-1}U(\pi(A_i))^dU^{-1}U = U^{-1}(\pi(UA_iU^{-1}))^dU \\ &= U^{-1}\begin{pmatrix} (\pi(\mu_{i1}))^d & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & (\pi(\mu_{i\ell}))^d \end{pmatrix}U = U^{-1}\begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{pmatrix}U = I, \end{aligned}$$

because $\pi(\mu_{ij}) \in U_d$ and by Lemma 17. In a similar way, as $\pi(\mu_{ij}) \in B$,

$$\varrho(C_i) = U^{-1}\begin{pmatrix} \varrho(\mu_{i1}) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \varrho(\mu_{i\ell}) \end{pmatrix}U = I.$$

For the second part of the claim we first need verify that all the eigenvalues of the C_i's belong to B. Let U be a unitary operator that diagonalizes the C_i's and

$$UC_iU^{-1} = \begin{pmatrix} \mu_{i1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mu_{i\ell} \end{pmatrix},$$

assuming that the C_i's are ℓ-dimensional. Then we have

$$I = \varrho(C_i) = U^{-1}U\varrho(C_i)U^{-1}U = U^{-1}\varrho(UC_iU^{-1})U = U^{-1}\begin{pmatrix} \varrho(\mu_{i1}) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \varrho(\mu_{i\ell}) \end{pmatrix}U,$$

implying $\varrho(\mu_{ij}) = 1$. Then we proceed as in the first part of the claim. $\square$

Next we consider the four connections from Figure 3 one by one and prove that each of them preserves the satisfiability gap.

Step 1 (Reductions to a core). Let Γ be a constraint language over the set U_d and $\varrho : U_d \to U_d$ an idempotent endomorphism of Γ. Then CSP(Γ) has a satisfiability gap if and only if CSP($\varrho(\Gamma)$) does.

Suppose that $|\mathsf{Im}\ (\varrho)| = e$, let $\pi' : \mathsf{Im}\ (\varrho) \to U_e$ be any bijection between U_e and $\mathsf{Im}\ (\varrho)$, and $\pi = \pi' \circ \varrho$. Let $\Delta = \{\pi(R) \mid R \in \Gamma\}$, we show that CSP($\Delta$) has a satisfiability gap if and only if CSP(Γ) does.

Let $\mathcal{P} = (V, U_e, \mathcal{C})$ be a gap instance of CSP(Δ), and let $\mathcal{P}^\pi = (V, U_d, \mathcal{C}^\pi)$ be the corresponding instance of CSP(Γ), where for each $\langle \mathbf{s}, R\rangle \in \mathcal{C}$ the set $\mathcal{C}^\pi$ includes $\langle \mathbf{s}, Q\rangle$ with $Q \in \Gamma$ and $\pi(Q) = R$. As is easily seen, $\mathcal{P}^\pi$ has no solution, because for any solution φ of $\mathcal{P}^\pi$ the mapping $\pi \circ \varphi$ is a solution of $\mathcal{P}$. Let

$\{A_v \mid v \in V\}$ be an ℓ-dimensional satisfying operator assignment for $\mathcal{P}$. We prove that $\{\pi'^{-1}(A_v) \mid v \in V\}$ is a satisfying operator assignment for $\mathcal{P}^\pi$. Let $C_v = \pi'^{-1}(A_v)$.

By Lemma 22, the C_v's are normal, $C_v^d = I$, $v \in V$, and for any constraint $\langle \mathbf{s}, R\rangle \in \mathcal{C}$ and any $v, w \in \mathbf{s}$, C_v, C_w commute. Now, let $\langle \mathbf{s}, R\rangle \in \mathcal{C}$, $\mathbf{s} = (v_1, \dots, v_k)$, $\langle \mathbf{s}, Q\rangle$ be the corresponding constraint of $\mathcal{P}^\pi$, and f_R, f_Q be the polynomials representing R, Q over U_e, U_d respectively. We have $f_R(A_{v_1}, \dots, A_{v_k}) = I$, and we need to show that $f_Q(C_{v_1}, \dots, C_{v_k}) = I$. Let U diagonalize $A_{v_1}, \dots, A_{v_k}$ and

$$UA_{v_i}U^{-1} = \begin{pmatrix} \mu_{i1} & \dots & 0 \\ & \ddots & \\ 0 & \dots & \mu_{i\ell} \end{pmatrix}.$$

Then $(\mu_{1j}, \dots, \mu_{kj}) \in R$ for $j \in [\ell]$, because

$$I = f_R(A_{v_1}, \dots, A_{v_k}) = f_R(UA_{v_1}U^{-1}, \dots, UA_{v_k}U^{-1}) = \begin{pmatrix} f_R(\mu_{11}, \dots, \mu_{k1}) & \dots & 0 \\ & \ddots & \\ 0 & \dots & f_R(\mu_{1\ell}, \dots, \mu_{k\ell}) \end{pmatrix}.$$

Therefore, $(\pi'^{-1}(\mu_{1j}), \dots, \pi'^{-1}(\mu_{kj})) \in \varrho(Q)$ for $j \in [\ell]$, and, as ϱ is an endomorphism, $(\pi'^{-1}(\mu_{1j}), \dots, \pi'^{-1}(\mu_{kj})) \in Q$ and so $f_Q(\pi'^{-1}(\mu_{1j}), \dots, \pi'^{-1}(\mu_{kj})) = \lambda_0 = 1$. Since

$$UC_{v_i}U^{-1} = U\pi'^{-1}(A_{v_i})U^{-1} = \pi'^{-1}(UA_{v_i}U^{-1}) = \begin{pmatrix} \pi'^{-1}(\mu_{i1}) & \dots & 0 \\ & \ddots & \\ 0 & \dots & \pi'^{-1}(\mu_{i\ell}) \end{pmatrix},$$

we also have $f_Q(C_{v_1}, \dots, C_{v_k}) = \lambda_0 = 1$.

Now, let $\mathcal{P}^\pi = (V, U_d, \mathcal{C}^\pi)$ be a gap instance of CSP(Γ), $\{C_v \mid v \in V\}$ an ℓ-dimensional satisfying operator assignment for $\mathcal{P}^\pi$, and $\mathcal{P} = (V, U_e, \mathcal{C})$, $\mathcal{C} = \{\langle \mathbf{s}, R\rangle \mid \langle \mathbf{s}, Q\rangle \in \mathcal{C}^\pi, R = \pi(Q)\}$, the corresponding instance of CSP(Δ). Then again $\mathcal{P}$ has no solution over U_e, and it remains to prove that $\{A_v \mid v \in V\}$, $A_v = \pi(C_v)$ is a satisfying operator assignment for $\mathcal{P} = (V, \mathcal{C})$. For $\langle \mathbf{s}, R\rangle \in \mathcal{C}$, $\mathbf{s} = (v_1, \dots, v_k)$, let f_R, f_Q be polynomials representing R and $Q \in \Gamma$ with $\pi(Q) = R$, respectively.

First, we show that the C_v's can be replaced with $\varrho(C_v)$. Since for any $\langle \mathbf{s}, Q\rangle \in \mathcal{C}^\pi$, $\mathbf{s} = (v_1, \dots, v_k)$, and any $a_1, \dots, a_k \in U_d$, we have that $f_Q(\varrho(a_1), \dots, \varrho(a_k)) = 1$ whenever $f_Q(a_1, \dots, a_k) = 1$, by Lemma 2 $f_Q(\varrho(C_{v_1}), \dots, \varrho(C_{v_k})) = I$ whenever $f_Q(C_{v_1}, \dots, C_{v_k}) = I$. Thus, we assume $A_v = \pi'(C_v)$ for $v \in V$.

That A_v is normal, $A_v^e = I$, $v \in V$, and the A_v's locally commute follows from Lemma 22. Let $f_Q^\pi(x_1, \dots, x_k) = f_Q(\pi'^{-1}(x_1), \dots, \pi'^{-1}(x_k))$. As is easily seen, for any $a_1, \dots, a_k \in U_e$, if $f_Q^\pi(a_1, \dots, a_k) = 1$ then $f_R(a_1, \dots, a_k) = 1$. Hence, by Lemma 2 $f_R(A_{v_1}, \dots, A_{v_k}) = I$ whenever $f_Q^\pi(A_{v_1}, \dots, A_{v_k}) = I$. Finally, we prove that $\pi'^{-1}(A_v) = C_v$; this implies that $f_Q^\pi(A_{v_1}, \dots, A_{v_k}) = I$ completing the proof. Let U diagonalize C_v and

$$UC_vU^{-1} = \begin{pmatrix} \mu_1 & \dots & 0 \\ & \ddots & \\ 0 & \dots & \mu_\ell \end{pmatrix}.$$

Then

$$\begin{aligned} \pi'^{-1}(A_v) &= \pi'^{-1}(\pi'(C_v)) = U^{-1}U\pi'^{-1}(\pi'(C_v))U^{-1}U = U^{-1}\pi'^{-1}(\pi'(UC_vU^{-1}))U \\ &= U^{-1}\begin{pmatrix} \pi'^{-1}(\pi'(\mu_1)) & \dots & 0 \\ & \ddots & \\ 0 & \dots & \pi'^{-1}(\pi'(\mu_\ell)) \end{pmatrix}U = U^{-1}\begin{pmatrix} \mu_1 & \dots & 0 \\ & \ddots & \\ 0 & \dots & \mu_\ell \end{pmatrix}U = C_v. \end{aligned}$$

Step 2 (Adding constant relations). Let Γ be a core language. Then $\mathrm{CSP}(\Gamma)$ has a satisfiability gap if and only if $\mathrm{CSP}(\Gamma^*)$ does.

Since $\Gamma \subseteq \Gamma^*$, if $\mathrm{CSP}(\Gamma)$ has a satisfiability gap, so does $\mathrm{CSP}(\Gamma^*)$. We prove that if $\mathrm{CSP}(\Gamma^*)$ has a satisfiability gap then $\mathrm{CSP}(\Gamma)$ has a satisfiability gap.

We will use the following relation R_Γ that is known to be pp-definable in Γ [29]:

$$R_\Gamma = \{(\varrho(\lambda_0), \ldots, \varrho(\lambda_{d-1})) \mid \varrho \text{ is an endomorphism of } \Gamma\}.$$

As Γ is a core language, for any $(a_1, \ldots, a_d) \in R_\Gamma$ it holds that $\{a_1, \ldots, a_d\} = U_d$. By Theorem 15 we may assume that $R_\Gamma \in \Gamma$.

Let $\mathcal{P} = (V, U_d, \mathcal{C})$ be a gap instance of $\mathrm{CSP}(\Gamma^*)$. We construct an instance $\mathcal{P}' = (V', U_d, \mathcal{C}')$ of $\mathrm{CSP}(\Gamma)$ as follows.

- $V' = V \cup \{v_a | a \in U_d\}$;
- $\mathcal{C}'$ consists of three parts: $\{C = \langle \mathbf{s}, R\rangle \in \mathcal{C} \mid R \in \Gamma\}$, $\{\langle (v_{a_1}, \ldots, v_{a_n}), R_\Gamma\rangle\}$, and $\{\langle (v, v_a), =_d\rangle \mid \langle (v), C_a\rangle \in \mathcal{C}\}$, where $=_d$ denotes the equality relation on U_d.

It is known [29] that $\mathcal{P}'$ has a classic solution if and only if $\mathcal{P}$ has one. If $\varphi : V \to U_d$ is a solution of $\mathcal{P}$ then we can extend it to a solution of $\mathcal{P}'$ by mapping v_a to a. Conversely, let $\varphi : V' \to U_d$ be a solution of $\mathcal{P}'$. The restriction of φ to V may not be a solution of $\mathcal{P}$, because for some constraint $\langle (v), C_a\rangle \in \mathcal{C}$ it may be the case that $\varphi(v) = \varphi(v_a) \neq a$. This can be fixed as follows. As φ is a solution, $(\varphi(v_{\lambda_0}), \ldots, \varphi(v_{\lambda_{d-1}})) \in R_\Gamma$, hence, the mapping $\varrho : U_d \to U_d$ given by $\varrho(a) = \varphi(v_a)$ is an endomorphism of Γ. As Γ is a core language, ϱ is a permutation on U_d and ϱ^s is the identity permutation for some s. Then ϱ^{s-1} is the inverse ϱ^{-1} of ϱ and is also an endomorphism of Γ. Therefore $\varphi' = \varrho^{-1} \circ \varphi$ is also a solution of $\mathcal{P}'$ and $\varphi'(v_a) = a$ for $a \in U_d$. Thus, $\varphi'|_V$ is a solution of $\mathcal{P}$.

Now, suppose that $\{A_v \mid v \in V\}$ is an ℓ-dimensional satisfying operator assignment for $\mathcal{P}$. First, we observe that if $\mathcal{C}$ contains a constraint $\langle (v), C_a\rangle$ then A_v is the scalar operator aI. Indeed, let $f_a(x)$ be a polynomial representing C_a, that is, $f_a(a) = \lambda_0$ and $f_a(b) = \lambda_1$ for $b \in U_d - \{a\}$. Let also U be a unitary operator that diagonalizes A_v and $\mu_1, \ldots, \mu_\ell$ the eigenvalues of A_v. Then, as $f_a(A_v) = I$ we obtain

$$I = UIU^{-1} = Uf_a(A_v)U^{-1} = f_a(UA_vU^{-1}) = \begin{pmatrix} f_a(\mu_1)) & \ldots & 0 \\ & \ddots & \\ 0 & \ldots & f_a(\mu_\ell)) \end{pmatrix}$$

implying that $f_a(\mu_i) = \lambda_0$ for $i \in [\ell]$. Thus, a is the only eigenvalue of A_v and

$$A_v = U^{-1}aIU = aI.$$

All such operators pairwise commute regardless of the value of a. Therefore v_a can be assigned aI for $a \in U_d$, and the resulting assignment is a satisfying operator assignment for $\mathcal{P}'$.

Step 3 (Satisfiability gap from subalgebras). Let Γ be a constraint language over the set U_d and let B be its subalgebra. Then if $\mathrm{CSP}(\Gamma|_B)$ has a satisfiability gap then so does $\mathrm{CSP}(\Gamma)$.

Let $\Delta = \Gamma|_B$. Then by Theorem 15 we may assume $\Delta \subseteq \Gamma$ and $B \in \Gamma$. Let $e = |B|$ and $\pi : U_e \to U_d$ a bijection between U_e and B.

Let $\mathcal{P} = (V, U_e, \mathcal{C})$ be a gap instance of $\mathrm{CSP}(\pi^{-1}(\Delta))$ and the instance $\mathcal{P}^\pi = (V, U_d, \mathcal{C}^\pi)$ constructed as follows. For every $\langle \mathbf{s}, R\rangle \in \mathcal{C}$ the instance $\mathcal{P}^\pi$ contains $\langle \mathbf{s}, \pi(R)\rangle$. As is easily seen, $\mathcal{P}^\pi$ has no classic solution. Therefore, it suffices to show that for any ℓ-dimensional satisfying operator assignment $\{A_v \mid v \in V\}$ for $\mathcal{P}$, the assignment $C_v = \pi(A_v)$ is a satisfying operator assignment for $\mathcal{P}^\pi$.

By Lemma 22, the C_v's are normal, satisfy the condition $C_v^d = I$, and locally commute. For $\langle \mathbf{s}, R\rangle \in \mathcal{C}$, $\mathbf{s} = (v_1, \dots, v_k)$, let $f_R^\pi(x_1, \dots, x_k) = f_R(\pi^{-1}(x_1), \dots, \pi^{-1}(x_k))$. As in **Step 1**, it can be shown that $\pi^{-1}(C_v) = A_v$, and therefore $f_R^\pi(C_{v_1}, \dots, C_{v_k}) = I$. For any $a_1, \dots, a_k \in U_d$, if $(a_1, \dots, a_k) \in \pi(R)$ then $a_1, \dots, a_k \in B$. Therefore, if $f_R^\pi(a_1, \dots, a_k) = 1$ then $f_{\pi(R)}(a_1, \dots, a_k) = 1$. By Lemma 2 this implies $f_{\pi(R)}(C_{v_1}, \dots, C_{v_k}) = I$.

Step 4 (Satisfiability gap from homomorphic images). Let Γ be a constraint language over the set U_d and θ a congruence of Γ. Then if $\mathrm{CSP}(\Gamma/_\theta)$ has a satisfiability gap then so does $\mathrm{CSP}(\Gamma)$.

Let $\varrho : U_d \to U_d/_\theta$ be the natural mapping $a \mapsto a/_\theta$ and $\pi' : U_d/_\theta \to U_e$, where $e = |U_d/_\theta|$, a bijection. Finally, let $\pi = \pi' \circ \varrho$ and $\Delta = \pi(\Gamma)$. For $R \in \Delta$ let $\pi^{-1}(R)$ be the full preimage of R under π. Since θ is pp-definable in Γ, so is $\pi^{-1}(R)$ for any $R \in \Delta$. Indeed, if $R = \pi(Q)$ for some $Q \in \Gamma$, then

$$\pi^{-1}(R)(x_1, \dots, x_k) = \exists y_1, \dots, y_k \;\; Q(y_1, \dots, y_k) \wedge \bigwedge_{i\in[k]} \theta(x_i, y_i).$$

Using Theorem 15 we may assume that $\pi^{-1}(R) \in \Gamma$ for $R \in \Delta$. Let $\pi^* : U_e \to U_d$ assign to $a \in U_e$ a representative of the θ-class ${\pi'}^{-1}(a)$. Thus, in a certain sense, π^* is an inverse of π.

Suppose that $\mathcal{P} = (V, U_e, \mathcal{C})$ is a gap instance of $\mathrm{CSP}(\Delta)$ and let $\mathcal{P}^\pi = (V, U_d, \mathcal{C}^\pi)$ be given by $\mathcal{C}^\pi = \{\langle \mathbf{s}, \pi^{-1}(R)\rangle \mid \langle \mathbf{s}, R\rangle \in \mathcal{C}\}$. We prove that $\mathcal{P}^\pi$ is a gap instance of $\mathrm{CSP}(\Gamma)$. Firstly, observe that $\mathcal{P}^\pi$ has no classic solution, because for any solution φ of $\mathcal{P}^\pi$ the mapping $\pi \circ \varphi$ is a solution of $\mathcal{P}$. Let $\{A_v \mid v \in V\}$ be a satisfying operator assignment for $\mathcal{P}$. We set $C_v = \pi^*(A_v)$ and prove that $\{C_v \mid v \in V\}$ is a satisfying operator assignment for $\mathcal{P}^\pi$. By Lemma 22, the C_v's are normal, satisfy the condition $C_v^d = I$, and locally commute. For $\langle \mathbf{s}, R\rangle \in \mathcal{C}$, $\mathbf{s} = (v_1, \dots, v_k)$, let $f_R^\pi(x_1, \dots, x_k) = f_R(\pi(x_1), \dots, \pi(x_k))$. As before, it is easy to see that $\pi(C_v) = A_v$ for $v \in V$, and therefore $f_R^\pi(C_{v_1}, \dots, C_{v_k}) = I$. Finally, for any $(a_1, \dots, a_k) \in U_d$, if $f_R^\pi(a_1, \dots, a_k) = f_R(\pi(a_1), \dots, \pi(a_k)) = 1$, then $(\pi(a_1), \dots, \pi(a_k)) \in R$, and so $(a_1, \dots, a_k) \in \pi^{-1}(R)$ and $f_{\pi^{-1}(R)}(a_1, \dots, a_k) = 1$. By Lemma 2 this implies that $f_{\pi^{-1}(R)}(C_{v_1}, \dots, C_{v_k}) = I$.

Proof of Theorem 19. Suppose that $\mathrm{CSP}(\Gamma)$ does not have bounded width. Then by Proposition 21 there exists a language Δ pp-definable in Γ, a subalgebra B of $\mathsf{core}(\Delta)^*$, a congruence θ of $\mathsf{core}(\Delta)^*|_B$, and an Abelian group $\mathbb{A}$ of prime order p such that $\mathsf{core}(\Delta)^*|_B/_\theta$ contains relations $R_{3,a}, R_{p+2}$ for every $a \in \mathbb{A}$ given by

$$R_{3,a} = \{(x, y, z) \mid x + y + z = a\}, \quad \text{and} \quad R_{p+2} = \{(a_1, \dots, a_{p+2}) \mid a_1 + \dots + a_{p+2} = 0\}.$$

By what is proved above it suffices to prove that $\mathrm{CSP}(\Delta_p)$, $\Delta_p = \{R_{3,a} \mid a \in \mathbb{A}\} \cup \{R_{p+2}\}$, has a satisfiability gap. However, if $p = 2$, the Mermin-Peres magic square from [44] provides a gap instance of $\mathrm{CSP}(\{R_{3,1}, R_{3,-1}\})$, and the construction from Proposition 14 provides a gap instance for $\mathrm{CSP}(\Delta_p)$ when $p > 2$. Note that in both cases the group $\mathbb{A}$ is the multiplicative group of roots of unity. ☐

References

[1] Padmanabhan K Aravind. Bell's theorem without inequalities and only two distant observers. *Found. Phys.*, 15:397–405, 2002. doi:10.1023/A:1021272729475.

[2] Albert Atserias, Phokion G. Kolaitis, and Simone Severini. Generalized satisfiability problems via operator assignments. *J. Comput. Syst. Sci.*, 105:171–198, 2019. arXiv:1704.01736, doi:10.1016/J.JCSS.2019.05.003.

[3] Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. ACM*, 61(1):3:1–3:19, 2014. doi:10.1145/2556646.

[4] Libor Barto and Marcin Kozik. Robustly solvable constraint satisfaction problems. *SIAM J. Comput.*, 45(4):1646–1669, 2016. arXiv:1512.01157, doi:10.1137/130915479.

[5] Libor Barto, Andrei Krokhin, and Ross Willard. Polymorphisms, and how to use them. In Andrei Krokhin and Stanislav Živný, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2017. doi:10.4230/DFU.Vol7.15301.1.

[6] John S Bell. On the problem of hidden variables in quantum mechanics. *Rev. Mod. Phys.*, 38(3):447, 1966. doi:10.1103/RevModPhys.38.447.

[7] Manuel Bodirsky. *Complexity of infinite-domain constraint satisfaction*, volume 52. Cambridge University Press, 2021.

[8] Manuel Bodirsky and Jakub Rydval. On the descriptive complexity of temporal constraint satisfaction problems. *J. ACM*, 70(1):2:1–2:58, 2023. doi:10.1145/3566051.

[9] Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. SDPs and robust satisfiability of promise CSP. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC'23)*, pages 609–622. ACM, 2023. arXiv:2211.08373, doi:10.1145/3564246.3585180.

[10] Jonah Brown-Cohen and Prasad Raghavendra. Combinatorial optimization algorithms via polymorphisms. *CoRR*, abs/1501.01598, 2015. arXiv:1501.01598.

[11] Andrei Bulatov. Bounded relational width. Unpublished manuscript, 2009. URL: https://www2.cs.sfu.ca/~abulatov/papers/relwidth.pdf.

[12] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005. doi:10.1137/S0097539700376676.

[13] Andrei A. Bulatov. A Dichotomy Theorem for Constraints on a Three-Element Set. In *Proc. 43rd Symposium on Foundations of Computer Science (FOCS'02)*, pages 649–658. IEEE Computer Society, 2002. doi:10.1109/SFCS.2002.1181990.

[14] Andrei A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006. doi:10.1145/1120582.1120584.

[15] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proc. 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*, pages 319–330, 2017. arXiv:1703.03021, doi:10.1109/FOCS.2017.37.

[16] Richard Cleve, Peter Høyer, Benjamin Toner, and John Watrous. Consequences and limits of nonlocal strategies. In *Proc. 19th Annual IEEE Conference on Computational Complexity (CCC'04)*, pages 236–249. IEEE Computer Society, 2004. doi:10.1109/CCC.2004.1313847.

[17] Richard Cleve, Li Liu, and William Slofstra. Perfect commuting-operator strategies for linear system games. *J. Math. Phys.*, 58(1), 2017. arXiv:1606.02278, doi:10.1063/1.4973422.

[18] Richard Cleve and Rajat Mittal. Characterization of binary constraint system games. In *Proc. 41st International Colloquium on Automata, Languages, and Programming (ICALP'14)*, volume 8572 of *Lecture Notes in Computer Science*, pages 320–331. Springer, 2014. doi:10.1007/978-3-662-43948-7_27.

[19] Eric Culf, Hamoon Mousavi, and Taro Spirig. Approximation algorithms for noncommutative constraint satisfaction problems. 2023. arXiv:2312.16765.

[20] Víctor Dalmau and Andrei A. Krokhin. Robust Satisfiability for CSPs: Hardness and Algorithmic Results. *ACM Trans. Comput. Theory*, 5(4):15:1–15:25, 2013. doi:10.1145/2540090.

[21] Víctor Dalmau and Justin Pearson. Closure functions and width 1 problems. In *Proc. 4th International Conference on Principles and Practice of Constraint Programming (CP'99)*, volume 1713 of *Lecture Notes in Computer Science*, pages 159–173. Springer, 1999. doi:10.1007/978-3-540-48085-3_12.

[22] Romuald Debruyne and Christian Bessière. Some Practicable Filtering Techniques for the Constraint Satisfaction Problem. In *Proc. 15th International Joint Conference on Artificial Intelligence (IJCAI'97)*, pages 412–417. Morgan Kaufmann, 1997.

[23] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766.

[24] Michel X. Goemans and David P. Williamson. Approximation algorithms for Max-3-Cut and other problems via complex semidefinite programming. *J. Comput. Syst. Sci.*, 68(2):442–470, 2004. doi:10.1016/J.JCSS.2003.07.012.

[25] Venkatesan Guruswami and Yuan Zhou. Tight Bounds on the Approximability of Almost-Satisfiable Horn SAT and Exact Hitting Set. *Theory Comput.*, 8(1):239–267, 2012. doi:10.4086/TOC.2012.V008A011.

[26] Paul R Halmos. *Introduction to Hilbert space and the theory of spectral multiplicity*. Courier Dover Publications, 2017.

[27] Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *J. Comb. Theory, Ser. B*, 48(1):92–110, 1990. doi:10.1016/0095-8956(90)90132-J.

[28] Pavol Hell and Jaroslav Nešetřil. *Graphs and homomorphisms*, volume 28 of *Oxford Lecture Series in Mathematics and its Applications*. OUP Oxford, 2004.

[29] Peter Jeavons, David A. Cohen, and Marc Gyssens. How to determine the expressive power of constraints. *Constraints*, 4(2):113–131, 1999. doi:10.1023/A:1009890709297.

[30] Peter G. Jeavons, David A. Cohen, and Marc Gyssens. Closure properties of constraints. *J. ACM*, 44(4):527–548, 1997. doi:10.1145/263867.263489.

[31] Zhengfeng Ji. Binary constraint system games and locally commutative reductions. 2013. arXiv:1310.3794.

[32] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP*=RE. Technical report, 2020. arXiv:2001.04383.

[33] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP* = RE. *Commun. ACM*, 64(11):131–138, 2021. doi:10.1145/3485628.

[34] Simon Kochen and E. P. Specker. The problem of hidden variables in quantum mechanics. *J. Mathematics and Mechanics*, 17(1):59–87, 1967. URL: http://www.jstor.org/stable/24902153.

[35] Phokion G. Kolaitis and Moshe Y. Vardi. On the expressive power of datalog: Tools and a case study. *J. Comput. Syst. Sci.*, 51(1):110–134, 1995. doi:10.1006/jcss.1995.1055.

[36] Phokion G. Kolaitis and Moshe Y. Vardi. Conjunctive-query containment and constraint satisfaction. *J. Comput. Syst. Sci.*, 61(2):302–332, 2000. doi:10.1006/jcss.2000.1713.

[37] Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. The complexity of general-valued CSPs. *SIAM J. Comput.*, 46(3):1087–1110, 2017. arXiv:1502.07327, doi:10.1137/16M1091836.

[38] Marcin Kozik. Solving CSPs Using Weak Local Consistency. *SIAM J. Comput.*, 50(4):1263–1286, 2021. arXiv:1605.00565, doi:10.1137/18M117577X.

[39] Marcin Kozik, Andrei Krokhin, Matt Valeriote, and Ross Willard. Characterizations of several Maltsev conditions. *Algebra universalis*, 73(3):205–224, 2015. doi:10.1007/s00012-015-0327-2.

[40] Benoit Larose and Lázló Zádori. Bounded width problems and algebras. *Algebra Univers.*, 56:439–466, 2007. doi:10.1007/s00012-007-2012-6.

[41] Laura Mančinska and David E. Roberson. Quantum isomorphism is equivalent to equality of homomorphism counts from planar graphs. In *Proc. 61st IEEE Annual Symposium on Foundations of Computer Science (FOCS'20)*, pages 661–672. IEEE, 2020. doi:10.1109/FOCS46700.2020.00067.

[42] Laura Mančinska and David E. Roberson. Quantum homomorphisms. *J. Comb. Theory, Ser. B*, 118:228–267, 2016. doi:10.1016/J.JCTB.2015.12.009.

[43] Miklós Maróti and Ralph McKenzie. Existence theorems for weakly symmetric operations. *Algebra Univers.*, 59(3-4):463–489, 2008. doi:10.1007/s00012-008-2122-9.

[44] N. David Mermin. Simple unified form for the major no-hidden-variables theorems. *Phys. Rev. Lett.*, 65(27):3373, 1990. doi:10.1103/PhysRevLett.65.3373.

[45] N. David Mermin. Hidden variables and the two theorems of John Bell. *Rev. Mod. Phys.*, 65(3):803, 1993. doi:10.1103/RevModPhys.65.803.

[46] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[47] Connor Paddock and William Slofstra. Satisfiability problems and algebras of boolean constraint system games. 2023. arXiv:2310.07901.

[48] Asher Peres. Incompatible results of quantum measurements. *Phys. Lett.*, 151(3-4):107–108, 1990. doi:10.1016/0375-9601(90)90172-K.

[49] E.L. Post. *The two-valued iterative systems of mathematical logic*, volume 5 of *Annals of Mathematical Studies*. Princeton University Press, 1941. doi:10.2307/2268608.

[50] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC'08)*, pages 245–254. ACM, 2008. doi:10.1145/1374376.1374414.

[51] Thomas Schaefer. The complexity of satisfiability problems. In *Proc. 10th Annual ACM Symposium on the Theory of Computing (STOC'78)*, pages 216–226, 1978. doi:10.1145/800133.804350.

[52] William Slofstra. Tsirelson's problem and an embedding theorem for groups arising from non-local games. *J. Am. Math. Soc.*, 33(1):1–56, 2020. doi:10.1090/jams/929.

[53] Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. *J. ACM*, 63(4):37:1–37:33, 2016. arXiv:1210.2987, doi:10.1145/2974019.

[54] Johan Thapper and Stanislav Živný. The power of Sherali–Adams relaxations for general-valued CSPs. *SIAM J. Comput.*, 46(4):1241–1279, 2017. arXiv:1606.02577, doi:10.1137/16M1079245.

[55] Johan Thapper and Stanislav Živný. The limits of SDP relaxations for general-valued CSPs. *ACM Trans. Comput. Theory*, 10(3):12:1–12:22, 2018. arXiv:1612.01147, doi:10.1145/3201777.

[56] Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In *Proc. 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*, pages 331–342, 2017. arXiv:1704.01914, doi:10.1109/FOCS.2017.38.

[57] Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *J. ACM*, 67(5):30:1–30:78, 2020. arXiv:1704.01914, doi:10.1145/3402029.