Satisfiability of Commutative vs. Non-Commutative CSPs

Andrei A. Bulatov
School of Computing Science, Simon Fraser University, Burnaby, Canada

Stanislav Živný
Department of Computer Science, University of Oxford, 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 0-Valid-SAT, 1-Valid-SAT, 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, these notions differ as there are gaps, i.e., there are unsatisfiable instances that are satisfiable via operators on Hilbert spaces.

We generalize their result to CSPs on *arbitrary finite* domains and give an almost complete classification: First, we show that NP-hard CSPs admit a separation between classical satisfiability and satisfiability via operators on finite- and infinite-dimensional Hilbert spaces. Second, we show that tractable CSPs of bounded width have no satisfiability gaps of any kind. Finally, we show that tractable CSPs of unbounded width can simulate, in a satisfiability-gap-preserving fashion, linear equations over an Abelian group of prime order p; for such CSPs, we obtain a separation of classical satisfiability and satisfiability via operators on infinite-dimensional Hilbert spaces. Furthermore, if $p = 2$, such CSPs also have gaps separating classical satisfiability and satisfiability via operators on finite- and infinite-dimensional Hilbert spaces.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Problems, reductions and completeness

Keywords and phrases constraint satisfaction, quantum CSP, operator CSP

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.37

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2404.11709` [18]

Funding *Andrei A. Bulatov*: NSERC Discovery Grant.
Stanislav Živný: UKRI EP/X024431/1.

Acknowledgements We thank the anonymous reviewers for their feedback and William Slofstra for useful discussions.

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 [16, 66], infinite-domain CSPs [7, 8], optimization CSPs [61], as well as approximable CSPs [57, 10],

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).
Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis
Article No. 37; pp. 37:1–37:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 [22, 21, 2, 53, 46].

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.

Back in 1978, Schaefer famously classified all Boolean CSPs as solvable in polynomial time or NP-hard [58].[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 [34]. 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 [33]. 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 [27].

Following the so-called algebraic approach to CSPs, pioneered by Jeavons, Cohen and Gyssens [35] and Bulatov, Jeavons, and Krokhin [12], the conjecture was resolved in the affirmative in 2017 by Bulatov [16] and, independently, by Zhuk [65, 66]. 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 [31, 24, 4], classifying the complexity of exact optimization CSPs [61, 42], 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].

[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.

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 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 [50, 51] and the construction is now know as the Mermin-Peres magic square [54]. This construction proves the Bell-Kochen-Specker theorem on the impossibility to explain quantum mechanics via hidden variables [6, 39].

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 if the instance is satisfiable [22]. The Mermin-Peres construction [50, 54] 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 [50, 54] 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 [20]. The game was studied systematically for Boolean CSP by Cleve and Mittal [22] and further studied by Cleve, Liu, and Slofstra [21].

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 Hilbert space 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 should pairwise commute.

Ji showed that for Boolean CSPs corresponding to 2-SAT there is no difference between (classical) satisfiability and satisfiability via operators [36]. Later, Atserias, Kolaitis, and Severini established a complete classification for all Boolean CSPs parameterized by the set of allowed constraint relations. In particular, using the substitution method [22] they showed that that only CSPs whose relations are 0-valid, 1-valid, or 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 (over both finite- and infinite-dimensional Hilbert spaces) [2]. For all other Boolean CSPs, Atserias et al. [2] showed that there are satisfiability gaps in the following sense: There are instances that are not (classically) satisfiable but are satisfiable via operators on finite-dimensional Hilbert spaces, relying on the Mermin-Peres magic square – gaps of the *first kind*. This immediately implies that there are instances that are not (classically) satisfiable but are satisfiable via operators on infinite-dimensional Hilbert spaces – gaps of the *second kind*. Finally, there are instances that are not satisfiable via operators on finite-dimensional Hilbert spaces but are satisfiable via operators on infinite-dimensional Hilbert spaces, relying on the breakthrough result of Slofstra who proved this for linear equations [59] – gaps of the *third kind*. The result is established in [2] 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 [58], 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, [23, 30, 56], 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 is a partial classification of satisfiability gaps for CSPs on arbitrary finite domains, with the remaining cases being related to a well-known open problem in the area, which is the existence of gap instances of the first kind for linear equations over $\mathbb{Z}_p$ with $p > 2$.

In detail, as our first result we prove that *NP-hard* CSPs do have gaps of all three kinds. Second, we show that CSPs of *bounded width* (on arbitrary finite domains) do not have a satisfiability gap (of any kind), meaning that classical satisfiability is equivalent to satisfiability via operators on finite- and infinite-dimensional Hilbert spaces. Third, all other CSPs (i.e., tractable CSPs of unbounded width) can simulate, in a precise technical sense, linear equations over an Abelian group of prime order p. We prove that these simulations preserve satisfiability gaps. Thus, we obtain a satisfiability gap of the second kind by the result from an upcoming paper by Slofstra and Zhang [60]. Moreover, for $p = 2$, we obtain a satisfiability gap of the first kind (and thus also of the second kind) from the Mermin-Peres magic square [50, 51, 54], and a satisfiability gap of the third kind from Slofstra's result [59]. Finally, the reductions will also establish the first claim, gaps of all three kinds for NP-hard CSPs via [50, 51, 54, 59].

[3] For finite-dimensional Hilbert spaces, a linear operator is just, up to a choice of basis, a matrix with complex entries, and the adjoint A^* of A is the conjugate transpose of A, also denoted by A^*. For infinite-dimensional Hilbert spaces, the notions are more involved, cf. Section 2 for all details.

▶ **Theorem 1** (Main result, informal statement). *Let Γ be an arbitrary finite set of relations on a finite domain.*

1. *If* $\mathrm{CSP}(\Gamma)$ *is NP-hard then* $\mathrm{CSP}(\Gamma)$ *has gaps of all three kinds.*
2. *Otherwise,* $\mathrm{CSP}(\Gamma)$ *is solvable in polynomial time and*
   a. *either Γ has bounded width and* $\mathrm{CSP}(\Gamma)$ *has no gaps of any kind,*
   b. *or* $\mathrm{CSP}(\Gamma)$ *does not have bounded width and* $\mathrm{CSP}(\Gamma)$ *can simulate linear equations over an Abelian group of prime order p in a satisfiability-gap-preserving fashion. In this case,* $\mathrm{CSP}(\Gamma)$ *has a gap of the second kind. Furthermore, if $p = 2$ then* $\mathrm{CSP}(\Gamma)$ *has gaps of all three kinds.*

We note that Theorem 1 implies a classification of Boolean CSPs, thus recovering the result by Atserias et al. [2]. Indeed, a Boolean language Γ that is 0-valid, 1-valid, or comes from 2-SAT, Horn-SAT, or Dual Horn-SAT has bounded width; also, for a Boolean Γ one gets $p = 2$ in Theorem 1 (2b).

Theorem 1 also implies a classification for CSPs on graphs (such CSPs are also known as H-Coloring problems): If $\Gamma = \{R\}$, where R is a binary symmetric relation, then either R is bipartite or not. In the former case, $\mathrm{CSP}(\Gamma)$ has bounded width [17] and thus has no gaps of any kind. In the latter case, $\mathrm{CSP}(\Gamma)$ is NP-hard [33, 14] and thus has gaps of all three kinds.

▶ **Corollary 2.** *Let $G = (V, E)$ be a graph. If G is bipartite,* $\mathrm{CSP}(\{E\})$ *has no satisfiability gap of any kind. Otherwise it has satisfiability gaps of all three kinds.*

We note that the remaining cases not covered by Theorem 1 (i.e., $\mathrm{CSP}(\Gamma)$ that are tractable but have unbounded width and can simulate linear equations over an Abelian group of prime order $p > 2$ but not of order 2) relate to the known difficulties of establishing satisfiability gaps of the first kind for linear equations over $\mathbb{Z}_p$ with $p > 2$, e.g., there are results showing that such an instance cannot be obtained from generalized Pauli matrices [56] and beyond [29].[4]

The proof of Theorem 1 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.[5] 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 [43]. Thirdly, to prove that NP-hard CSPs and certain 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.

We note 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,

[4] We remark that our results show that one can obtain gap instances from homomorphic preimages of already established gap instance (cf. Step 4 in [18]).

[5] [49, Theorem 6.4] considers subdivisions, a special case of conjunctive definitions, which in turn are a special case of primitive positive definitions (without existential quantifiers). Our result shows that [49, Theorem 6.4] extends to conjunctive definitions.

while the complexity of Boolean CSPs was established by Schaefer in 1978 [58] and the complexity of CSPs on three-element domains was established by Bulatov in 2002 [13, 15], the dichotomy for all finite domains was only established in 2017 [16, 65, 66]. 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 [55]), 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 might be 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. Indeed, if satisfiability gaps of the first kind exist for linear equations over $\mathbb{Z}_p$ with $p > 2$ then our main result implies that all CSPs of unbounded width admit satisfiability gaps of all three kinds. This is yet another result indicating the fundamental nature of bounded width, which captures not only the power of the local consistency algorithm [45, 48, 11, 3] as conjectured in [27] with links to Datalog, pebble games, and logic [27, 41], but also robust solvability of CSPs [4], exact solvability of valued CSPs by LP [62] and SDP [63] relaxations, and now also possibly satisfiability via operators.

Related work

Paddock and Slofstra [53] streamlined the results from [2], and also gave an overview of other notions of satisfiability and their relationship, including the celebrated MIP*=RE result of Ji et al. [37, 38]. Further, Culf, Mousavi, and Spirig recently studied approximability of operator CSPs [23]. While we used the magic square to construct a gap of the first kind, other systems based on linear equations could be used instead, e.g. the pentagram and the BCS in [59], cf. also [64]. Finally, we note that the notion of quantum homomorphism from the work of Mančinska and Roberson [47], as well as the recent work of Ciardo [19], is different from ours except when $d = 2$. In the case of $d = 2$, as considered in [2], operator assignments have to be self-adjoint and can be transformed to quantum isomorphisms. When $d \geq 3$, the straightforward translation through Fourier transform in cyclic groups result in projectors that are not self-adjoint, and therefore do not give rise to a quantum isomorphism. However, the overall results are somewhat similar, e.g., Ciardo shows [19] that CSPs of bounded width have no quantum advantage (over finite-dimensional spaces) by establishing a so-called minion homomorphism – relying crucially on idempotency – to the minion capturing the basic SDP relaxation, which is known to solve CSPs of bounded width by the work of Barto and Kozik [4]. The methods used in [19] rely, unlike methods and results in this paper, on being in the finite-dimensional case.

2 Preliminaries

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 CSP(Γ) 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 [27]. 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 [45, 48, 11, 3, 44],[6] we will rely on the result of Kozik [43] 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 $\mathcal{P} \in$ CSP(Γ) into an equivalent subinstance $\mathcal{P}' \in$ CSP(Γ) that satisfies the AC property. Intuitively, given an instance $\mathcal{P} = (V, D, \mathcal{C}) \in$ CSP(Γ), 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 [27, 25].

Equivalently, Arc-Consistency can be described in terms of a Datalog program [40]. In general, a Datalog program derives facts about an instance $\mathcal{P} \in$ CSP(Γ) 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

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

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.

[6] One characterization implies that checking whether a constraint language has bounded width is decidable [44].

Algorithm 1 SLAC.

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** $D_v = D$
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** $\mathcal{P}'$

The Singleton Arc-Consistency (SAC) algorithm is a modification of the AC algorithm [26]. 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 [43] and Zhuk [66]) that uses the Linear AC algorithm rather than the AC algorithm. Kozik has shown that SLAC solves all CSPs of bounded width [43]. 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 Algorithm 1, does not change the instance.

Multi-dimensional Fourier transform

Let U_d be the set of d-th roots of unity, that is, $U_d = \{\lambda_k = e^{\frac{2\pi i}{d}k} \mid 0 \le k < d\}$. 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 [52].

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. 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 a polynomial P. We say that operators A and B *commute* if $AB = BA$. A collection of linear operators $A_1, \ldots, A_n$ *pairwise commute* if $A_i A_j = A_j A_i$ for every $i, j \in [n]$. If $A_1, \ldots, A_n$ pairwise commute then $P(A_1, \ldots, A_n)$ is well defined. $\mathbb{C}[x_1, \ldots, x_n]$ denotes the ring of polynomials with complex coefficients and commuting variables $x_1, \ldots, x_n$. A linear operator is *bounded* if it maps bounded subsets to bounded subsets. Let A be a densely defined linear operator, i.e., A is defined almost everywhere. We denote by A^* its *adjoint* [28] and call A *normal* if A commutes with its adjoint, i.e., $AA^* = A^*A$. A linear operator U is called *unitary* if $UU^* = U^*U = I$, the identity operator.

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. Thus, for Hilbert spaces of finite dimension 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, the adjoint operator A^* is the conjugate transpose. Recall that $(AB)^* = B^*A^*$. We will use the following form of the so-called *strong spectral theorem.*

▶ **Theorem 3** ([32]). *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 $i \in [r]$.*

In the infinite-dimensional case, the Strong Spectral Theorem is more complicated. Firstly, it involves general L^2- and L^∞-spaces. Let $(\Omega, \mathcal{M}, \mu)$ be a measure space, i.e., Ω is a set, $\mathcal{M}$ is a σ-algebra on the set Ω (i.e. $\mathcal{M}$ is a nonempty collection of subsets of Ω containing Ω and closed under complements, countable intersections, and countable unions), and μ is a measure on $(\Omega, \mathcal{M})$ (i.e. μ is nonnegative and countably additive). Then $L^2(\Omega, \mu)$ denotes the collection of square integrable measurable functions, up to almost everywhere equality, and $L^\infty(\Omega, \mu)$ denotes the collection of essentially bounded measurable functions, up to almost everywhere equality; all measure-theoretic terms in these definitions refer to μ, cf. [28] for details. Instead of diagonal matrices, the General Strong Spectral Theorem uses *multiplication operators* on a $L^2(\Omega, \mu)$-space. Let V be a complex vector space of functions mapping an index set X to $\mathbb{C}$. A multiplication operator of V is a linear operator whose value at a function $f : X \to \mathbb{C}$ in V is given by pointwise multiplication by a fixed function $a : X \to \mathbb{C}$. In other words, the multiplication operator given by a is

$$(T_a(f))(x) = a(x)f(x) \qquad \text{for each } x \in X.$$

▶ **Theorem 4** (General Strong Spectral Theorem [28]). *Let $A_1, \ldots, A_r$ be normal bounded linear operators on a Hilbert space $\mathcal{H}$. If $A_1, \ldots, A_r$ pairwise commute then there exist measure space $(\Omega, \mathcal{M}, \mu)$, a unitary map $U : \mathcal{H} \to L^2(\Omega, \mu)$, and functions $a_1, \ldots, a_r \in L^\infty(\Omega, \mu)$ such that $A_i = U^{-1}T_{a_i}U$ 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 on 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 $\mathcal{P}$ on a Hilbert space $\mathcal{H}$ is a mapping that assigns to every variable from V an operator A_v on $\mathcal{H}$ such that

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

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

We call an operator assignment $\{A_v\}$ *satisfying for* $\mathcal{P}$ if $P_R(A_{v_1}, \dots, A_{v_r}) = I$ for every constraint $\langle (v_1, \dots, v_r), R \rangle \in \mathcal{C}$. Let $\mathcal{P}$ be a CSP instance. Following the terminology of Atserias et al. [2], we say that $\mathcal{P}$ has a *satisfiability gap of the first kind* if $\mathcal{P}$ is not satisfiable over U_d but is satisfiable by an operator assignment on a finite-dimensional Hilbert space. Similarly, we say that $\mathcal{P}$ has a *satisfiability gap of the second kind* if $\mathcal{P}$ is not satisfiable over U_d but is satisfiable by an operator assignemnt on an infinite-dimensional Hilbert space. Finally, we say that $\mathcal{P}$ has a *satisfiability gap of the third kind* if $\mathcal{P}$ is not satisfiable on finite-dimensional Hilbert spaces but is satisfiable by an operator assignment on an infinite-dimensional Hilbert space. We say that CSP(Γ) has a satisfiablity gap of the i-th kind, $i = 1, 2, 3$, if there is at least one instance $\mathcal{P} \in$ CSP(Γ) witnessing such a gap. By definition, a gap of the first kind implies a gap of the second kind. Also, a gap of the third kind implies a gap of the second kind. Put differently, if Γ has no gap of the second kind then it has no gap of the first or third kind either.

We shall repeatedly use the following simple lemma, proved in Appendix A.

▶ **Lemma 5.** *Let $x_1, \dots, x_r$ be variables, let $Q_1, \dots, Q_m, Q$ be polynomials in $\mathbb{C}[x_1, \dots, x_r]$, and let $\mathcal{H}$ be a Hilbert space. If every assignment over U_d that satisfies the equations $Q_1 = \dots = Q_m = 0$ also satisfies the equation $Q = 0$, then every fully commuting operator assignment on $\mathcal{H}$ that satisfies the equations $Q_1 = \dots = Q_m = 0$ also satisfies the equation $Q = 0$.*

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 full version of the paper [18].

Bounded width

One part of our main result is the following.

▶ **Theorem 6.** *Let Γ be a constraint language over U_d. If* CSP(Γ) *has bounded width then it has no satisfiability gap of any kind.*

The main idea behind the proof of Theorem 6 is to simulate the inference provided by SLAC (see Algorithm 1) by inference in polynomial equations. Let $\mathcal{S}$ be a SLAC-program solving CSP(Γ). In order to prove Theorem 6 we take an instance $\mathcal{P} = (V, U_d, \mathcal{C})$ of CSP(Γ) that has no solution, and therefore is not SLAC-consistent, as CSP(Γ) 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 CSP(Γ) 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 $(x \in S) \wedge R(x, y, z_1, \dots, z_r)) \to (y \in S')$ 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 $S \subseteq U_d$, the unary constraint restricting the domain of a variable x to the set S is represented by the polynomial

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

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

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

To give an idea of how Theorem 6 is proved, we sketch the proof of the following.

▶ **Lemma 7.** *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 5 that any operator assignment is a zero of $Rule_{S,R,S'}$, see [18] for details. This can be used to establish the claim for $i = 2$ [18]. 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, then multiply (3) by $(Dom_{\overline{S}_1}(A_{v_1}) - I)$ on the left and subtract. The problem is, however, that

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

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. ◀

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

Figure 1 Reductions between CSPs corresponding to derivative languages.

To complete the proof of Theorem 6 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 7, 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.

NP-hard CSPs and unbounded width

The second part of our main result boils down to the following.

▶ **Theorem 8.** *Let Γ be a constraint language over U_d. If* $\mathrm{CSP}(\Gamma)$ *does not have bounded width then* $\mathrm{CSP}(\Gamma)$ *can simulate linear equations over an Abelian group of prime order p in a satisfiability-gap-preserving fashion. Moreover, if* $\mathrm{CSP}(\Gamma)$ *is NP-hard then one can take $p = 2$.*

Assume that $\mathrm{CSP}(\Gamma)$ does not have bounded width. The "implementation" of linear equations in $\mathrm{CSP}(\Gamma)$ that preserves gaps will be achieved 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 1. 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 is either the binary equality relation on U_d or $R_i \in \Gamma$ 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 9.** *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 of the first (second, third) kind then so does* $\mathrm{CSP}(\Gamma)$.

Weaker forms of Theorem 9 have appeared in the literature. This includes for example subdivisions from Theorem 6.4 of [49].

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 (and similarly for the binary equality relation, cf. [18] for details). 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 9, it suffices to show the following; the proof is a simple generalization of the $d = 2$ case proved in [2].

▶ **Lemma 10.** *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 (finite or infinite dimensional) satisfying operator assignment for $\mathcal{P}$ then there is a (respectively, finite or infinite dimensional) satisfying operator assignment for $\mathcal{P}'$.*

If $\mathrm{CSP}(\Gamma \cup \{R\})$ has a satisfiability gap of the first kind 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 10, $\mathcal{P}'$ has a satisfying operator assignment. Hence $\mathcal{P}'$ establishes that $\mathrm{CSP}(\Gamma)$ has a satisfiability gap, and Theorem 9 is proved. This argument also extends to gaps of the second and third kind.

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 1 indicates. In the full version [18], we show that all arrows in Figure 1 preserve satisfiability gaps. To relate these reductions with bounded width and magic squares we use the following result.

▶ **Proposition 11** ([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 $\Gamma' = \mathsf{core}(\Delta)^*|_B/_\theta$ 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 $R_{p+2} = \{(a_1, \ldots, a_{p+2}) \mid a_1 + \cdots + a_{p+2} = 0\}$. If $\mathrm{CSP}(\Gamma)$ is NP-hard, then Δ can be chosen to contain $R_{3,a}, R_{p+2}$ for $p = 2$.*[8]

To prove Theorem 8, suppose that $\mathrm{CSP}(\Gamma)$ does not have bounded width. By Proposition 11 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 $\Gamma' = \mathsf{core}(\Delta)^*|_B/_\theta$ contains relations $R_{3,a}, R_{p+2}$ for every $a \in \mathbb{A}$. Our goal is to show that if $\mathrm{CSP}(\Gamma')$ has a satisfiability gap then so does $\mathrm{CSP}(\Gamma)$.

[8] 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 $\mathbb{A}$.

To give an idea of the gap-preservation proofs, we sketch how a satisfiability gap is preserved. Let Γ be a constraint language over the set U_d and let B be its subalgebra. We show that if CSP($\Gamma|_B$) has a satisfiability gap then so does CSP(Γ). We show this for a gap of the first kind (over finite-dimensional Hilbert spaces); the infinite-dimensional case is more complicated, cf. [18] for details. Let $\Delta = \Gamma|_B$. Then by Theorem 9 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 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 [18], 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 5 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 3 to simultaneously diagonalize the restriction of an operator assignment to the scope of a constraint, Lemma 5 that relates polynomial equations over U_d and operators, and the above mentioned result on preservation of operator assignments by certain polynomials. The infinite-dimensional case is more delicate, relying on the General Strong Spectral Theorem (Theorem 4).

To finish the proof of Theorem 1, assume that CSP(Γ) does not have bounded width and Γ' from Proposition 11 is over a group of prime order p. As reductions preserve satisfiability gaps, it suffices that CSP(Δ_p), where $\Delta_p = \{R_{3,a} \mid a \in \mathbb{A}\} \cup \{R_{p+2}\}$, has a satisfiability gap. The result of Slofstra and Zhang [60] provides a gap instance of CSP($\{R_{3,1}, R_{3,-1}\}$) of the second kind for any prime p, and thus the same holds for CSP(Γ). For $p = 2$ the Mermin-Peres magic square from [50] provides a gap instance of CSP($\{R_{3,1}, R_{3,-1}\}$) of the first kind, and same holds for CSP(Γ). Finally, the result by Slofstra [59] provides a gap instance of CSP($\{R_{3,1}, R_{3,-1}\}$) of the third kind, and we get the same for CSP($\Gamma \cup \{R_T\}$), where R_T is the full binary relation on U_d. Finally, if CSP(Γ) is NP-hard then Γ can simulate any constraint language [12], and thus in particular Δ_2.

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. 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. 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. 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. H-coloring dichotomy revisited. *Theor. Comput. Sci.*, 349(1):31–39, 2005. doi:10.1016/J.TCS.2005.09.028.

15 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.

16 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. doi:10.1109/FOCS.2017.37.

17 Andrei A. Bulatov. Constraint satisfaction problems: complexity and algorithms. *ACM SIGLOG News*, 5(4):4–24, 2018. doi:10.1145/3292048.3292050.

18 Andrei A. Bulatov and Stanislav Živný. Satisfiability of commutative vs. non-commutative CSPs. *CoRR*, 2024. arXiv:2404.11709.

19 Lorenzo Ciardo. Quantum advantage and CSP complexity. In *Proc. 39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'24)*, pages 23:1–23:15. ACM, 2024. doi:10.1145/3661814.3662118.

20 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.

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

22 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.

23 Eric Culf, Hamoon Mousavi, and Taro Spirig. Approximation algorithms for noncommutative constraint satisfaction problems. *CoRR*, 2023. doi:10.48550/arXiv.2312.16765.

24 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.

25 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.

26 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.

27 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.

28 Gerald B. Folland. *A Course in Abstract Harmonic Analysis*. Studies in Advanced Mathematics. Taylor & Francis, 1994.

29 Markus Frembs and Cihan Okay amd Ho Yiu Chung. No state-independent contextuality can be extracted from contextual measurement-based quantum computation with qudits of odd prime dimension. *CoRR*, 2022. arXiv:2209.14018.

30 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.

31 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.

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

33 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.

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

35 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.

36 Zhengfeng Ji. Binary constraint system games and locally commutative reductions. *CoRR*, 2013. arXiv:1310.3794.

37 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP*=RE. *CoRR*, 2020. arXiv:2001.04383.

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

39 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.

40 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.

41 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.

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

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

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

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

46 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.

47 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.

48 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`.

49 Kieran Mastel and William Slofstra. Two prover perfect zero knowledge for MIP. In *Proc. 56th Annual ACM Symposium on Theory of Computing (STOC'24)*, pages 991–1002. ACM, 2024. `doi:10.1145/3618260.3649702`.

50 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`.

51 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`.

52 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

53 Connor Paddock and William Slofstra. Satisfiability problems and algebras of boolean constraint system games. *CoRR*, 2023. `arXiv:2310.07901`.

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

55 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`.

56 Hammam Qassim and Joel J Wallman. Classical vs quantum satisfiability in linear constraint systems modulo an integer. *J. Phys. A*, 53(38):385304, 2020. `doi:10.1088/1751-8121/aba306`.

57 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`.

58 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`.

59 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`.

60 William Slofstra and Luming Zhang. Small cancellation for solution groups. In preparation.

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

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

63 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. `doi:10.1145/3201777`.

64 Xingjian Zhang, Zhaokai Pan, and Guoding Liu. Unconditional quantum MAGIC advantage in shallow circuit computation. *CoRR*, 2024. `arXiv:2402.12246`.

65 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. `doi:10.1109/FOCS.2017.38`.

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

A Proof of Lemma 5

▶ **Lemma 12** (Lemma 5 restated). *Let $x_1, \dots, x_r$ be variables, let $Q_1, \dots, Q_m, Q$ be polynomials in $\mathbb{C}[x_1, \dots, x_r]$, and let $\mathcal{H}$ be a Hilbert space. If every assignment over U_d that satisfies the equations $Q_1 = \dots = Q_m = 0$ also satisfies the equation $Q = 0$, then every fully commuting operator assignment on $\mathcal{H}$ that satisfies the equations $Q_1 = \dots = Q_m = 0$ 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.

Finite-dimensional case. 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 3 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 $Q(A_1, \ldots, A_r) = U^{-1}Q(E_1, \ldots, E_r)U = 0$.

Infinite-dimensional case. 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 operators are assigned to $X_1, \ldots, X_r$. Then, since $A_1, \ldots, A_r$ are normal and commute, by Theorem 4 there exist a measure space $(\Omega, \mathcal{M}, \mu)$, a unitary map $U : \mathcal{H} \to L^2(\Omega, \mu)$, and functions $a_1, \ldots, a_r \in L^\infty(\Omega, \mu)$ such that, for the multiplication operators $E_i = T_{a_i}$ of $L^2(\Omega, \mu)$, the equalities $A_i = U^{-1}E_iU$ hold for $i \in [r]$. This implies $UA_iU^{-1} = E_i$. As $A_i^d = I$ we also have $E_i^d = I$. Therefore $a_i(\omega)^d = 1$ for almost all $\omega \in \Omega$, or more formally $\mu(\{\omega \in \Omega \mid (a_i(\omega)^d \neq 1\}) = 0$. Hence $a_i(\omega) \in U_d$ for almost all $\omega \in \Omega$. For every $\ell \in [m]$ we have $Q_\ell(E_1, \ldots, E_r) = UQ_\ell(A_1, \ldots, A_r)U^{-1}$, implying by the conditions of the lemma that $Q_\ell(E_1, \ldots, E_r) = 0$. Since E_i is the multiplication operator given by the function a_i, it holds that $Q_\ell(a_1(\omega), \ldots, a_r(\omega)) = 0$ for almost all $\omega \in \Omega$. As for almost all $\omega \in \Omega$, it holds that the value $a_i(\omega) \in U_d$ for each $i \in [r]$, we have $Q(a_1(\omega), \ldots, a_r(\omega)) = 0$ for almost all $\omega \in \Omega$. Therefore $Q(E_1, \ldots, E_r) = 0$ and this implies $Q(A_1, \ldots, A_r) = 0$ as before. ◀