Limitations of Affine Integer Relaxations for Solving Constraint Satisfaction Problems

Moritz Lichter
RWTH Aachen University, Germany

Benedikt Pago
University of Cambridge, UK

Abstract

We show that various recent algorithms for finite-domain constraint satisfaction problems (CSP), which are based on solving their affine integer relaxations, do not solve all tractable and not even all Maltsev CSPs. This rules them out as candidates for a universal polynomial-time CSP algorithm. The algorithms are $\mathbb{Z}$-affine k-consistency, BLP+AIP, BA^k, and CLAP. We thereby answer a question by Brakensiek, Guruswami, Wrochna, and Živný [10] whether a constant level of BA^ksolves all tractable CSPs in the negative: Indeed, not even a sublinear level k suffices. We also refute a conjecture by Dalmau and Opršal [20] (LICS 2024) that every CSP is either solved by $\mathbb{Z}$-affine k-consistency or admits a Datalog reduction from 3-colorability. For the cohomological k-consistency algorithm, that is also based on affine relaxations, we show that it correctly solves our counterexample but fails on an NP-complete template.

2012 ACM Subject Classification Theory of computation → Finite Model Theory; Theory of computation → Problems, reductions and completeness; Theory of computation → Complexity theory and logic

Keywords and phrases constraint satisfaction, affine relaxation, promise CSPs, $\mathbb{Z}$-affine k-consistency, cohomological k-consistency algorithm, Tseitin, graph isomorphism

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.166

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: `https://arxiv.org/abs/2407.09097` [28]

Funding *Moritz Lichter*: The author received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (SymSim: grant agreement No. 101054974).Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.
Benedikt Pago: Funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee: grant number EP/X028259/1.

Acknowledgements We thank a number of people for helpful discussions and valuable input at various stages of this work, especially also for acquainting us with the problem: We are grateful to Anuj Dawar, Martin Grohe, Andrei Krokhin, Adam Ó Conghaile, Jakub Opršal, Standa Živný, and Dmitriy Zhuk. We are especially indebted to Michael Kompatscher, who kindly provided us with the proof of Theorem 2 (3).

1 Introduction

Constraint satisfaction problems (CSPs) provide a general framework that encompasses a huge variety of different problems, from solving systems of linear equations over Boolean satisfiability to variants of the graph isomorphism problem. We view CSPs as homomorphism problems. A CSP is defined by a relational structure $\mathbf{A}$ called the *template* of the CSP. An *instance* is a structure $\mathbf{B}$ of matching vocabulary and the question is whether there is a homomorphism from $\mathbf{B}$ to $\mathbf{A}$. We only consider *finite-domain* CSPs, i.e., the template $\mathbf{A}$ is always finite. It had long been conjectured by Feder and Vardi [23] that every finite-domain CSP is NP-complete or in P. In 2017, the conjecture was confirmed independently by Bulatov [11] and Zhuk [33]. The complexity of a CSP is determined by the polymorphisms ("higher-dimensional symmetries") of its template. If the template has no a so-called *weak near-unanimity* polymorphism, then the corresponding CSP is NP-complete. For the other case, Bulatov and Zhuk presented sophisticated polynomial-time algorithms. A less involved algorithm had been known earlier for templates with a *Maltsev* polymorphism [12]. None of these algorithms is *universal* in the sense that on input $(\mathbf{B}, \mathbf{A})$ they decide whether $\mathbf{B}$ maps homomorphically into $\mathbf{A}$ in time polynomial in both $|\mathbf{B}|$ and $|\mathbf{A}|$. Instead, these are *families* of algorithms, one for each template. The question whether there is a universal, and ideally "simple", algorithm for all tractable CSPs, or even just for all Maltsev CSPs, is still open.

One natural approach towards universal algorithms is via affine relaxations of systems of linear equations over $\{0, 1\}$: Given a template $\mathbf{A}$, an instance $\mathbf{B}$, and possibly a width parameter k, the existence of a homomorphism $\mathbf{B} \to \mathbf{A}$ is encoded into a system of linear equations. If the domain of the variables is relaxed from $\{0, 1\}$ to $\mathbb{Z}$, the system can be solved in polynomial time [27, 29], and the transformation of the CSP into the equation system is also computationally easy. Thus, if this integer relaxation was exact for all tractable CSPs, or at least an interesting subclass thereof, such as all Maltsev CSPs, then computing and solving it would constitute a universal polynomial-time algorithm for that class. Several algorithms based on this idea have been developed in recent years, motivated specifically by the study of *Promise CSPs* [10, 9, 17, 20]. This is a relatively new variant of CSPs which generalize for example the approximate graph coloring problem and are still not very well understood. The algorithms can be applied just the same to classical CSPs, and not even for these, much is known about their power. In the present paper, we prove strong limitations for all these algorithms and show that even for Maltsev CSPs, none of them is universal: We construct a template $\mathbf{A}$ whose CSP is *not solved* by these algorithms by providing instances $\mathbf{B}$ that admit no homomorphism to $\mathbf{A}$ but which are accepted by the algorithms. This also refutes a conjecture by Dalmau and Opršal [20], that we expand upon below. Our result is in stark contrast to the situation for *valued CSPs*, an optimization version of CSP. For these, a surprisingly simple linear-algebraic algorithm solves all tractable cases optimally [31].

Let us briefly introduce the algorithms that are addressed by our construction. All of them make use of (slightly) different systems of equations, which can all be reduced to the *width-k affine relaxation*. Given a template structure $\mathbf{A}$, an instance $\mathbf{B}$, and a width $k \in \mathbb{N}$, the variables of the equation system are indexed with partial homomorphisms from induced size-k substructures of $\mathbf{B}$ to $\mathbf{A}$. A solution to the width-k affine relaxation is thus an assignment of numerical values to partial homomorphisms. The equations enforce a consistency condition, i.e., express that partial homomorphisms with overlapping domains receive values that fit together. This is related to, but stronger than, the *k-consistency* method: The k-consistency algorithm is a well-studied simple combinatorial procedure that checks for inconsistencies between local solutions and propagates these iteratively. This solves the *bounded width* CSPs (see e.g. [23, 4, 2]) but is not powerful enough to deal with *all* tractable CSPs [1]. The consistency conditions of the width-k affine relaxation are stronger in the sense that they enforce a *global* notion of consistency rather than a local one. The algorithms that fail to solve our counterexample are the following:

The **$\mathbb{Z}$-affine k-consistency algorithm** [20] (Section 5.1) runs the k-consistency procedure. All non-k-consistent partial homomorphisms are removed from the width-k affine relaxation. The algorithm accepts the instance $\mathbf{B}$ if and only if this modified version of the width-k affine relaxation has an integral solution. Dalmau and Opršal [20] conjectured that for all finite structures $\mathbf{A}$, CSP($\mathbf{A}$) is either Datalog$^{\cup}$-reducible to CSP($\mathbb{Z}$) and thus solved by $\mathbb{Z}$-affine k-consistency for a fixed k, or 3-colorability is Datalog$^{\cup}$-reducible to CSP($\mathbf{A}$) (see Conjecture 17). Assuming P $\neq$ NP, the conjecture implies that every tractable finite-domain CSP is solved by $\mathbb{Z}$-affine k-consistency.

The **BLP+AIP algorithm** by Brakensiek, Guruswami, Wrochna, and Živný [10] (Section 5.2) first solves the width-k affine relaxation over the non-negative rationals, where k is the arity of the template. Next, the integral width-k affine relaxation is checked for a solution, but every variable is set to 0 that is set to 0 by every rational solution. The **BA^{k}-algorithm** proposed by Ciardo and Živný [16] (Section 5.2) generalizes BLP+AIP: The width k is not fixed to be the arity of the template but is a parameter of the algorithm, like in $\mathbb{Z}$-affine k-consistency. In [16], it is shown that there is an *NP-complete* (promise) CSP on which the algorithm fails, but no tractable counterexample had been known until now.

The **CLAP algorithm**, due to Ciardo and Živný [17] (Section 5.3), tests in the first step, for each partial homomorphism f, whether f can receive weight exactly 1 in a non-negative rational solution of the width-k affine relaxation, where k is the arity of the template. If not, it is discarded. This is repeated until the process stabilizes. Then the width-k affine relaxation is solved over the integers, where all discarded partial homomorphisms are forced to 0.

▶ **Theorem 1.** *There is a Maltsev template with 7 elements that is neither solved by*
1. *$\mathbb{Z}$-affine k-consistency, for every $k \in o(n)$, where n is the instance size,*
2. *BLP+AIP,*
3. *BA^{k}, for every $k \in o(n)$, nor*
4. *the CLAP algorithm.*

Hence, none of the algorithms solves all tractable CSPs.

In particular, this answers a question of Brakensiek, Guruswami, Wrochna, and Živný [10] whether a constant level of the BA^{k} hierarchy solves all tractable CSPs in the negative: Indeed, not even a sublinear level suffices. It also refutes the aforementioned Conjecture 17 regarding the power of the $\mathbb{Z}$-affine k-consistency relaxation [20], under the assumption that P $\neq$ NP. But we actually show a stronger statement: Namely, 3-colorability is not Datalog$^{\cup}$-reducible to the CSP that we use in the proof of the above theorem (Lemma 19). This is shown via a known inexpressibility result for *rank logic* [26] and disproves the conjecture unconditionally.

To prove Theorem 1, we construct and analyze instances. Our template is a combination of systems of linear equations over the Abelian groups $\mathbb{Z}_2$ and $\mathbb{Z}_3$, but the template itself is not a group. Since the affine algorithms reduce CSPs to a problem over the infinite Abelian group $(\mathbb{Z}, +)$, we investigate for which finite groups this is possible: we study what we call *group coset-CSPs* (to distinguish them from equation systems over groups). The template of a coset-CSP consists of a finite group Γ, and its relations are cosets of powers of subgroups of Γ. They always have a Maltsev polymorphism [6]. Coset-CSPs have been studied as "group-CSPs" by Berkholz and Grohe [6, 8] or as "subgroup-CSPs" by Feder and Vardi [23].

▶ **Theorem 2.** *For each of the algorithms $\mathbb{Z}$-affine k-consistency, BLP+AIP, BA^{k}, and CLAP, the following is true:*
1. *Every coset-CSP over a finite Abelian group is solved (for $\mathbb{Z}$-affine k-consistency, k must be at least the arity of the template structure).*
2. *There exists a non-Abelian coset-CSP that is not solved, namely over S_{18}, the symmetric group on 18 elements (for any constant or even sublinearly growing k).*

3. *There are non-Abelian coset-CSPs that are solved, namely over any 2-nilpotent group of odd order. For example, there are non-Abelian 2-nilpotent semidirect products $\mathbb{Z}_{p^2} \rtimes \mathbb{Z}_p$ of order p^3 for each odd prime p.*

While Assertion 1 is easily derived from the literature [19, 3], it turns out somewhat surprisingly that Abelian groups are not the border of tractability for the affine algorithms: They also work over certain 2-nilpotent groups; these are in a sense the non-Abelian groups that are closest possible to being Abelian. Assertion 2 is shown with a construction that is "semantically equivalent" to the one that we use for Theorem 1, but whose template is a coset-CSP. However, the analysis of the instances is technically much more involved. The construction in Theorem 1 is simpler and yields a smaller template. We show that our first counterexample can be expressed as instances of the *graph isomorphism* problem with *bounded color class size*, that is, the isomorphism problem of vertex-colored graphs, in which each color is only used for a constant number of vertices. This problem is expressible as a coset-CSP over the symmetric group [8]. This also shows that the affine CSP algorithms cannot be adapted to solve the graph isomorphism problem. They fail already on the bounded color class version, which is known to be in P [24]. There exists another highly interesting affine CSP algorithm that we have not addressed so far. This is the *cohomological k-consistency* algorithm due to Ó Conghaile [19] (see Section 5.4). As it turns out, this algorithm is actually able to solve our counterexample correctly. Hence, for all we know, it may be possible that there is a $k \in \mathbb{N}$ such that cohomological k-consistency is a universal polynomial-time algorithm for Maltsev or even all tractable CSPs. However, we can show *without* complexity-theoretic assumptions that it fails on NP-complete CSPs. Recent work by Chan and Ng [14] independently shows a similar result, but with a different technique: They prove that random instances of certain NP-complete templates are not solved by cohomological k-consistency, even for $k \in o(n)$, but they do not know a *tractable* counterexample, either.

▶ **Theorem 3.** *The CSP on which the algorithms in Theorem 1 fail is solved by cohomological k-consistency, for every $k \geq 4$. There exists an NP-complete CSP such that for every constant $k \in \mathbb{N}$, cohomological k-consistency fails to solve it.*

Our Techniques. Our proof of Theorem 1 combines results due to Berkholz and Grohe [8] with a new *homomorphism or-construction* that encodes the disjunction of two CSPs. For a system of linear equations to have an integral solution, it suffices to have a rational *p-solution* and a rational *q-solution* (for p and q coprime), in which all non-zero values are of the form p^z, with $z \in \mathbb{Z}$, or q^z, respectively. Thus, it suffices to design the instances in such a way that these two co-prime rational solutions exist. For the algorithms that involve a width-parameter k, the additional challenge is to make the construction robust so that it works against any choice of k (in our case it works even if k grows with the instance size). The *Tseitin contradictions* [32] over *expander graphs* (see Section 4) achieve this robustness. It is known that these cannot be solved by "local" algorithms, e.g., the k-consistency method, for any constant k [1]. Berkholz and Grohe showed that the width-k relaxation for unsatisfiable Tseitin contradictions over $\mathbb{Z}_p$, for a prime p, still has a p-solution. We combine two unsatisfiable Tseitin systems over $\mathbb{Z}_2$ and $\mathbb{Z}_3$ in the aforementioned homomorphism or-construction (Section 3). This yields an unsatisfiable CSP instance whose width-k relaxation has a 2- and a 3-solution and thereby also an integral solution. The reason why this approach fails for the cohomological algorithm (Theorem 3) is that it solves the width-k relaxation when a partial homomorphism is fixed. This fixing of local solutions reduces the homomorphism or-construction to just solving equations over $\mathbb{Z}_2$ and $\mathbb{Z}_3$, respectively, which the affine relaxation can do. To prove the second part of Theorem 3, we modify the homomorphism or-construction so that cohomology no longer solves it, but this also makes the template NP-complete.

2 Preliminaries

We write $[k]$ for $\{1,\dots,k\}$. For $k \in \mathbb{N}$ and a set N, let $\binom{N}{\leq k}$ be the set of all subsets of N of size at most k. A *relational vocabulary* τ is a set of relation symbols $\{R_1,\dots,R_k\}$ with associated arities $\mathrm{ar}(R_i)$. A *relational τ-structure* is a tuple $\mathbf{A} = (A, R_1^{\mathbf{A}},\dots,R_k^{\mathbf{A}})$ of a *universe* A and interpretations of the relation symbols such that $R_i^{\mathbf{A}} \subseteq A^{\mathrm{ar}(R_i)}$ for all $i \in [k]$. We use letters $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ for finite relational structures. Their universes are denoted A, B, and C, respectively. If $\mathbf{A}$ is a structure and $X \subseteq A$, then $\mathbf{A}[X]$ denotes the induced substructure with universe X. For two τ-structures $\mathbf{A}$ and $\mathbf{B}$, we write $\mathrm{Hom}(\mathbf{A},\mathbf{B})$ for the set of *homomorphisms* $\mathbf{A} \to \mathbf{B}$. A *graph* $G = (V,E)$ is a binary $\{E\}$-structure, where we denote its *vertex set* by $V(G)$ and its *edge set* by $E(G)$. Group operations are written as multiplication, for Abelian groups we use additive notation.

CSPs and Polymorphisms. For a finite τ-structure $\mathbf{A}$, denote by $\mathrm{CSP}(\mathbf{A})$ the *CSP with template* $\mathbf{A}$, i.e., the class of finite τ-structures $\mathbf{B}$ such that there is a homomorphism $\mathbf{B} \to \mathbf{A}$. We call a structure $\mathbf{B}$ a $\mathrm{CSP}(\mathbf{A})$-instance if $\mathbf{B}$ has the same vocabulary as $\mathbf{A}$. The complexity of $\mathrm{CSP}(\mathbf{A})$, and also the applicability of certain algorithms, is determined by the *polymorphisms* of the τ-structure $\mathbf{A}$. An ℓ-ary polymorphism is a map $p\colon A^\ell \to A$ such that for every $R \in \sigma$ of arity $r = \mathrm{ar}(R)$ and all $\bar{a}_1,\dots,\bar{a}_\ell \in R^{\mathbf{A}}$, the tuple $(p(a_{11},a_{21},\dots,a_{\ell 1}),\dots,p(a_{1r},a_{2r},\dots,a_{\ell r}))$ is also in $R^{\mathbf{A}}$ (where a_{ij} denotes the j-th entry of the tuple $\bar{a}_i$). The polymorphisms of a structure are closed under composition. A ternary operation p is *Maltsev* if it satisfies the identity $p(x,x,y) = p(y,x,x) = y$ for all inputs. For a group Γ the map $f(x,y,z) = xy^{-1}z$ is a typical example of a Maltsev operation. The templates with Maltsev polymorphisms form a subclass of all tractable CSPs [12]. For more background on the algebraic approach to CSPs, see for example [5].

Logics, Interpretations, and Reductions. A logic L defines $\mathrm{CSP}(\mathbf{A})$ if there exists an L-formula F such that each instance $\mathbf{B}$ satisfies F if and only if $\mathbf{B} \in \mathrm{CSP}(\mathbf{A})$. *Inflationary fixed-point logic* (IFP) is the extension of first-order logic by an operator defining inflationary fixed-points. IFP defines connected components of graphs, which is not possible in pure first-order logic. More details are not needed and we refer to [22, Chapter 8.1]. A *logical interpretation* is a (partial) map from σ-structures to τ-structures defined by logical formulas in the following way. For a σ-structure, a formula using vocabulary σ with k free variables defines the set of all k-tuples of the structure satisfying the formula. An interpretation consists of a formula defining the new universe, and for each relation symbol $R \in \tau$ of a formula defining the relation R in the new structure. Formal details are not needed in this paper, and for more details we refer to [22, Section 11.2] or the full version [28]. Such interpretations can be used as reductions between decision problems. Of particular interest in the context of CSPs are Datalog-interpretations, which can be expressed as IFP-interpretations (again see [22, Theorem 9.1.4]). Dalmau and Opršal [20] consider a variant of these reductions called *Datalog$^\cup$-reductions*. We omit a definition and only note that Datalog$^\cup$-reductions can be expressed as IFP-interpretations, too.

The k-Consistency Algorithm. A well-known heuristic for CSPs is the k-consistency algorithm. For a template $\mathbf{A}$ and an instance $\mathbf{B}$, the k-consistency algorithm computes a map $\kappa_k^{\mathbf{A}}[\mathbf{B}]$ assigning to each $X \in \binom{B}{\leq k}$ a set of partial homomorphisms $\mathbf{B}[X] \to \mathbf{A}$: it is the unique greatest fixed-point that satisfies the following properties for all $Y \subset X \in \binom{B}{\leq k}$.

Forth-Condition: Every $f \in \kappa_k^{\mathbf{A}}[\mathbf{B}](Y)$ extends to some $g \in \kappa_k^{\mathbf{A}}[\mathbf{B}](X)$, that is, $g|_Y = f$.
Down-Closure: For every $g \in \kappa_k^{\mathbf{A}}[\mathbf{B}](X)$, we have $g|_Y \in \kappa_k^{\mathbf{A}}[\mathbf{B}](Y)$.
If $\kappa_k^{\mathbf{A}}[\mathbf{B}](X) = \emptyset$ for some $X \in \binom{B}{\leq k}$, then the algorithm rejects $\mathbf{B}$, otherwise it accepts.

CSP-Relaxation via Affine Systems of Linear Equations. We introduce a system of linear equations due to Berkholz and Grohe [8], which will be used to (approximately) solve CSPs. We transfer hardness results for this system to other systems used in the different algorithms. Let $\mathbf{A}$ be a template structure and $\mathbf{B}$ be an instance. The *width-k affine relaxation* $\mathsf{L}_{\mathsf{CSP}}^{k,\mathbf{A}}(\mathbf{B})$ aims to encode (approximately) whether $\mathbf{B}$ is in CSP($\mathbf{A}$).

$\mathsf{L}_{\mathsf{CSP}}^{k,\mathbf{A}}(\mathbf{B})$: variables $x_{X,f}$ for all $X \in \binom{B}{\leq k}$ and all $f \in \mathrm{Hom}(\mathbf{B}[X], \mathbf{A})$
$$\sum_{\substack{f \in \mathrm{Hom}(\mathbf{B}[X],\mathbf{A}),\\ f\|_{X\setminus\{b\}}=g}} x_{X,f} = x_{X\setminus\{b\},g} \quad \text{for all } X \in \tbinom{B}{\leq k}, b \in X, g \in \mathrm{Hom}(\mathbf{B}[X\setminus\{b\}],\mathbf{A})$$ $$x_{\emptyset,\emptyset} = 1$$

Here $\emptyset$ is the unique homomorphism $\mathbf{B}[\emptyset] \to \mathbf{A}$. If k is at least the arity of $\mathbf{A}$, then $\mathbf{B} \in$ CSP($\mathbf{A}$) if and only if $\mathsf{L}_{\mathsf{CSP}}^{k,\mathbf{A}}(\mathbf{B})$ has a nonnegative integral solution (and actually a $\{0,1\}$-solution) [8]. We will be mainly interested in integral solutions of $\mathsf{L}_{\mathsf{CSP}}^{k,\mathbf{A}}(\mathbf{B})$, so without the non-negativity restriction. To show the existence of these solutions, we will also consider special rational solutions:

▶ **Definition 4.** *For $p \in \mathbb{Z}$, a* p-solution *of a system of linear equations L with variables* $\mathrm{Var}(\mathsf{L})$ *is a solution $\Phi\colon \mathrm{Var}(\mathsf{L}) \to \mathbb{Q}$ of L such that, for all $x \in \mathrm{Var}(\mathsf{L})$, $\Phi(x) = 0$ or $\Phi(x) = p^i$ for some $i \in \mathbb{Z}$.*

▶ **Lemma 5** ([8])**.** *If p and q are coprime integers and a system L of linear equations over $\mathbb{Q}$ has a p-solution and a q-solution, then L has an integral solution, which is only non-zero for variables on which the p-solution or the q-solution is non-zero.*

Group Coset-CSPs. Let Γ be a finite group. In a Γ-*coset-CSP* [6, 23] variables range over Γ and the constraints are of the following form. For an r-tuple of variables $\bar{x} = (x_1, \dots, x_r)$, an r-ary Γ-*coset-constraint* is the constraint $\bar{x} \in \Delta\delta$, where $\Delta \leq \Gamma^r$ is a subgroup of Γ^r and $\delta \in \Gamma^r$, that is, $\Delta\delta$ is a right coset of Γ^r. With the term *coset-CSP* we refer to a Γ-coset-CSP in this sense. For every finite group Γ and every arity r, there is a structure $\Gamma^{[r]}$ such that every r-ary Γ-coset-CSP is a $\Gamma^{[r]}$-instance and CSP($\Gamma^{[r]}$) contains all solvable r-ary Γ-coset-CSPs. CSP($\Gamma^{[r]}$) is always tractable [23]. In fact, being a coset-CSP in this sense is equivalent to having the Maltsev polymorphism $f(x, y, z) = xy^{-1}z$.

▶ **Lemma 6.** *For every finite template $\mathbf{A} = (A, R_1^{\mathbf{A}}, ..., R_m^{\mathbf{A}})$ and every binary operation $\cdot\colon A \times A \to A$ such that $\Gamma = (A, \cdot)$ is a group,*
- *the map $f : \Gamma^3 \to \Gamma$ defined by $f(x, y, z) = xy^{-1}z$ is a polymorphism of $\mathbf{A}$ if and only if*
- *each relation $R_i^{\mathbf{A}}$ is a coset of a subgroup of Γ^r for some $r \in \mathbb{N}$.*

Thus, coset-CSPs are a natural class to study. In particular, being Maltsev, they are tractable even if Γ is non-Abelian. By contrast, solving *systems of equations* is NP-complete if (and only if) Γ is non-Abelian [25]. Systems of equations over an *Abelian* group Γ can however always be viewed as a Γ-coset-CSP. Hence, when we consider equation systems over Abelian groups in Section 4, we can treat them uniformly as coset-CSPs.

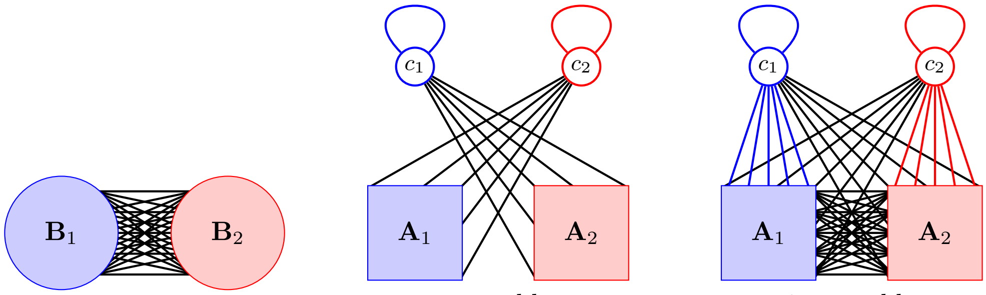

Figure 1 The different homomorphism or-constructions: The picture assumes that the two vocabularies τ_1 and τ_2 are binary and contain a single relation each (blue and red). It shows the instance $\mathbf{OR}[\mathbf{B}_1, \mathbf{B}_2]$. and the tractable and intractable template construction. The new S-relation is drawn in black, where the edges are all oriented from left to right. Pairs added to the relation of τ_1 or τ_2 are drawn in blue or red, respectively.

3 Homomorphism OR-Construction

For $i \in [2]$, let $\mathbf{A}_i$ and $\mathbf{B}_i$ be nonempty τ_i-structures and τ_1 and τ_2 be disjoint. The $\mathbf{A}_i$ are templates and the $\mathbf{B}_i$ the corresponding instances, and we assume that their universes are disjoint. We aim to define τ-structures $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{B} \in \mathrm{CSP}(\mathbf{A})$ if and only if $\mathbf{B}_i \in \mathrm{CSP}(\mathbf{A}_i)$ for some $i \in [2]$. Let S be a fresh binary relation symbol and $\tau := \tau_1 \cup \tau_2 \cup \{S\}$. The arities are inherited from τ_1 and τ_2. The instance $\mathbf{B} = \mathbf{OR}[\mathbf{B}_1, \mathbf{B}_2]$ with universe $B := B_1 \cup B_2$ has relations $S^{\mathbf{B}} := B_1 \times B_2$ and $R^{\mathbf{B}} := R^{\mathbf{B}_i}$ for all $i \in [2]$ and $R \in \tau_i$. We provide two variants of $\mathbf{A}$ with different properties.

The Tractable Case. The first variant of the template will preserve tractability of the $\mathrm{CSP}(\mathbf{A}_i)$ and hence is called the *tractable homomorphism or-construction*. The template $\mathbf{A} = \mathbf{OR}_{\mathrm{T}}[\mathbf{A}_1, \mathbf{A}_2]$ is defined as follows. Let c_1 and c_2 be two fresh vertices.

$$\begin{aligned} A &:= A_1 \cup A_2 \cup \{c_1, c_2\} \\ R^{\mathbf{A}} &:= R^{\mathbf{A}_i} \cup \{c_i\}^{\mathrm{ar}(R)} && \text{for all } i \in [2], R \in \tau_i \\ S^{\mathbf{A}} &:= \big(A_1 \times \{c_2\}\big) \cup \big(\{c_1\} \times A_2\big) \end{aligned}$$

The construction is depicted in Figure 1. Intuitively, a homomorphism $\mathbf{B}_i \to \mathbf{A}_i$ induces a homomorphism $\mathbf{B} \to \mathbf{A}$ by mapping all vertices of $\mathbf{B}_{3-i}$ to c_{3-i}. The relation S ensures that every homomorphism of $\mathbf{B} \to \mathbf{A}$ is of this form, which proves the following lemma:

▶ **Lemma 7.** $\mathbf{OR}[\mathbf{B}_1, \mathbf{B}_2] \in \mathrm{CSP}(\mathbf{OR}_T[\mathbf{A}_1, \mathbf{A}_2])$ *if and only if* $\mathbf{B}_i \in \mathrm{CSP}(\mathbf{A}_i)$ *for some* $i \in [2]$.

The next lemmas summarize the properties of the construction, which are required later: the tractable homomorpism or-construction preserves k-consistency of the $\mathbf{B}_i$ and inherits solutions of the width-k affine relations from the $\mathbf{B}_i$.

▶ **Lemma 8.** *Let* $\mathbf{A} = \mathbf{OR}_T[\mathbf{A}_1, \mathbf{A}_2]$, $\mathbf{B} = \mathbf{OR}[\mathbf{B}_1, \mathbf{B}_2]$, $k \in \mathbb{N}$, $i \in [2]$, $X \in \binom{B}{\leq k}$, *and* $f \in \mathrm{Hom}(\mathbf{B}[X], \mathbf{A})$. *If* $f(X \cap B_{3-i}) = \{c_{3-i}\}$ *and* $f|_{X \cap B_i} \in \kappa_k^{\mathbf{A}_i}[\mathbf{B}_i](X \cap B_i)$, *then* $f \in \kappa_k^{\mathbf{A}}[\mathbf{B}](X)$.

▶ **Lemma 9.** *Let* $\mathbf{A} = \mathbf{OR}_T[\mathbf{A}_1, \mathbf{A}_2]$*,* $\mathbf{B} = \mathbf{OR}[\mathbf{B}_1, \mathbf{B}_2]$*,* $i \in [2]$*, and* Φ *be a solution to* $\mathsf{L}^{k,\mathbf{A}_i}_{\mathsf{CSP}}(\mathbf{B}_i)$*. Then there exists a solution* Ψ *to* $\mathsf{L}^{k,\mathbf{A}}_{\mathsf{CSP}}(\mathbf{B})$ *defined, for all* $X \in \binom{B}{\leq k}$ *and all* $f \in \mathrm{Hom}(\mathbf{B}[X], \mathbf{A})$*, by* $\Psi(x_{X,f}) = \Phi(x_{X \cap B_i}, f|_{X \cap B_i})$ *if* $f(X \cap B_{3-i}) = \{c_{3-i}\}$ *and* $\Psi(x_{X,f}) = 0$ *otherwise. In particular,* Ψ *is a p-solution or integral, if* Φ *is a p-solution or integral, respectively.*

▶ **Lemma 10.** *If* $\mathbf{A}_1$ *and* $\mathbf{A}_2$ *have a Maltsev polymorphism, then* $\mathbf{OR}_T[\mathbf{A}_1, \mathbf{A}_2]$ *has one.*

Also if the $\mathbf{A}_i$ are not Maltsev, tractability is preserved.

▶ **Lemma 11.** *If* $\mathrm{CSP}(\mathbf{A}_1)$ *and* $\mathrm{CSP}(\mathbf{A}_2)$ *are tractable, then* $\mathrm{CSP}(\mathbf{OR}_T[\mathbf{A}_1, \mathbf{A}_2])$ *is tractable.*

For the proof of this lemma, we provide a polynomial-time algorithm for $\mathrm{CSP}(\mathbf{OR}_\mathrm{T}[\mathbf{A}_1, \mathbf{A}_2])$ in [28]. Roughly speaking, the algorithm proceeds as follows: If the instance $\mathbf{B}$ has more than one connected component, they can be treated individually. So we can assume that $\mathbf{B}$ is connected. The algorithm divides the universe B into two sets B_1 and B_2 such that elements in B_i are contained in tuples of a τ_i-relation. If one is contained in both a τ_1- and a τ_2-relation, then $\mathbf{B}$ is a no-instance. Because $\mathbf{B}$ is connected, B_1 and B_2 are connected by the relation S (not necessarily by a biclique). By the construction of the tractable or-construction, for any potential homomorphism there is an $i \in [2]$ such that B_i is mapped to c_i and B_{3-i} to $\mathbf{A}_{3-i}$. This can be tested by the algorithms for $\mathrm{CSP}(\mathbf{A}_1)$ and $\mathrm{CSP}(\mathbf{A}_2)$. Note that this algorithm essentially only computes connected components and calls the decision algorithms of the $\mathrm{CSP}(\mathbf{A}_i)$. Hence, it can be expressed in logics that are at least expressive as IFP.

▶ **Corollary 12.** *Let L be a logic that is at least expressive as inflationary fixed-point logic. If* $\mathrm{CSP}(\mathbf{A}_1)$ *and* $\mathrm{CSP}(\mathbf{A}_2)$ *are L-definable, then* $\mathrm{CSP}(\mathbf{OR}_T[\mathbf{A}_1, \mathbf{A}_2])$ *is L-definable.*

The tractable homomorphism or-construction has a drawback: forbidding a single partial homomorphism mapping vertices of $\mathbf{B}_i$ to c_i resolves the or-construction for the width-k affine relaxation: every such solution to the width-k affine relaxation of $\mathbf{OR}_\mathrm{T}[\mathbf{B}_1, \mathbf{B}_2]$ induces a solution for $\mathbf{B}_i$.

▶ **Lemma 13.** *Let* $k \geq 2$*,* $i \in [2]$*, and* Φ *be a solution to* $\mathsf{L}^{\mathbf{A},k+1}_{\mathsf{CSP}}(\mathbf{B})$*. If there is a set* $X \in \binom{B_i}{\leq k}$ *such that for* $f\colon X \to \{c_i\}$ *it holds that* $\Phi(x_{X,f}) = 0$*, then* $\Phi|_{B_i}$ *is a solution to* $\mathsf{L}^{\mathbf{A}_i,k}_{\mathsf{CSP}}(\mathbf{B}_i)$*.*

In some situations this will cause problems later. To remedy this problem, we make the construction more flexible at the cost of giving up tractability.

The Intractable Case. The second variant, which is called the *intractable homomorphism or-construction*, $\mathbf{A} = \mathbf{OR}_\mathrm{NPC}[\mathbf{A}_1, \mathbf{A}_2]$ has the same universe as before, but interprets relations differently:

$$R^{\mathbf{A}} := R^{\mathbf{A}_i} \cup \Big((A_i \cup \{c_i\})^{\mathrm{ar}(R)} \setminus A_i^{\mathrm{ar}(R)} \Big) \qquad \text{for all } i \in [2], R \in \tau_i$$
$$S^{\mathbf{A}} := \big(A_1 \times (A_2 \cup \{c_2\})\big) \cup \big((A_1 \cup \{c_1\} \times A_2\big)$$

The construction is shown in Figure 1. The definition of S still enforces that a homomorphism $\mathbf{B} \to \mathbf{A}$ has to induce a homomorphism $\mathbf{B}_i \to \mathbf{A}_i$ for one i. Only for one i, the extra vertex c_i can be in the image of a homomorphism. Thus, this variant provides again a homomorphism-or construction, but now two *partial* homomorphisms $\mathbf{B}_1 \to \mathbf{A}_1$ and $\mathbf{B}_2 \to \mathbf{A}_2$ can be combined:

▶ **Lemma 14.** *Let* $\mathbf{A} = \mathbf{OR}_{NPC}[\mathbf{A}_1, \mathbf{A}_2]$*,* $\mathbf{B} = \mathbf{OR}[\mathbf{B}_1, \mathbf{B}_2]$*,* $X_i \subseteq B_i$ *for each* $i \in [2]$*, and consider homomorphisms* $f_i \in \mathrm{Hom}(\mathbf{B}_i[X_i], \mathbf{A}_i)$ *for both* $i \in [2]$*. The map* $f\colon X_1 \cup X_2 \to A$ *induced by* f_1 *and* f_2 *satisfies* $f \in \mathrm{Hom}(\mathbf{B}[X_1 \cup X_2], \mathbf{A})$*.*

Also the intractable construction inherits solutions of the width-k affine relaxation and preserves k-consistency similar to Lemmas 8 and 9, but now for the combined partial homomorphisms. Instead of requiring that $f(X \cap B_{3-i}) = \{c_{3-i}\}$, we only require that $f|_{X \cap B_{3-i}} \in \mathrm{Hom}(\mathbf{B}_{3-i}[X], \mathbf{A}[A_{3-i} \cup \{c_{3-i}\}])$. Already for easy templates the CSP of the intractable homomorphism-or construction is NP-complete. The reason is that for general instances the S-relation does not have to be a biclique and hence a yes-instance does not have to induce a homomorphism for one $\mathbf{B}_i$ at all. Even for choosing $\mathbf{A}_1$ and $\mathbf{A}_2$ to be the structure with a single element and one nonempty ternary relation, the intractable OR-construction yields an NP-complete CSP (e.g. shown by a reduction from 3-SAT). Many other CSPs can be reduced to this case, for example group-coset-CSPs.

4 Tseitin Formulas over Abelian Groups and Expanders

Our construction is based on graphs with high connectivity, so-called *expanders*. We do not give the standard definition of this notion because we only need one property of expanders, namely this: We say that a family of 2-connected 3-regular graphs $(G_n)_{n \in \mathbb{N}}$ is a family of *expander graphs* if there is a constant $c > 0$ (the *expansion constant*) such that for all G_n and $X \subseteq E(G_n)$, there is a set $\hat{X} \supseteq X$ of size $|\hat{X}| \leq c|X|$ such that $E(G) \setminus \hat{X}$ is empty or the edge set of a 2-connected subgraph of G. The existence of such families is shown in [7, Corollary A.4]. Let G be a 2-connected 3-regular graph. Fix an orientation H of G, i.e., a directed graph with one direction of each edge of G. Let $V := V(G), E := E(G)$ in this section. For a set $W \subseteq V$, denote by $\delta_-(W) \subseteq E$ the set of all $uv \in E$ such that $(u, v) \in E(H) \cap (V \setminus W) \times W$. Analogously, $\delta_+(W) \subseteq E$ is the set of all edges leaving W, and $\delta(W) := \delta_+(W) \cup \delta_-(W)$. Fix a finite Abelian group Γ. Let $\lambda\colon V \to \Gamma$. Define the Γ-coset-CSP $\mathcal{C}^{H,\Gamma,\lambda}$, or $\mathcal{C}^\lambda$ for short, with variable set $\{y_e \mid e \in E\}$ and linear equations

$$\sum_{e \in \delta_+(v)} y_e - \sum_{e \in \delta_-(v)} y_e = \lambda(v) \qquad \text{for all } v \in V.$$

In the case $\Gamma = \mathbb{Z}_2$, we obtain the classic Tseitin contradictions [32]. The CSP $\mathcal{C}^\lambda$ is solvable if and only if $\sum_{v \in V} \lambda(v) = 0$ [8]. For all sets $W \subseteq V$, the CSP $\mathcal{C}^\lambda$ implies the constraint $C(W)$ defined via

$$\sum_{e \in \delta_+(W)} y_e - \sum_{e \in \delta_-(W)} y_e = \sum_{v \in W} \lambda(v).$$

▶ **Definition 15** (Robustly Consistent Assignments [8])**.** *For* $\lambda\colon V \to \Gamma$ *and a set* $X \subseteq E$*, a partial assignment* $f\colon X \to \Gamma$ *for* $\mathcal{C}^\lambda$ *is* ℓ-consistent*, if for every* $W \in \binom{V}{\leq \ell}$ *such that* $\delta(W) \subseteq X$*, the assignment* f *satisfies the constraint* $C(W)$*. Note that* f *is a partial solution if it is* 1*-consistent. We call* f robustly consistent *if it is* $n/3$*-consistent.*

We review facts about robustly consistent assignments for $\mathcal{C}^\lambda$. The detailed statements and proofs can be found in the full version and in [8]. The key idea is that the expansion property of G ensures that $\mathcal{C}^\lambda$ is always locally satisfiable, on subinstances of size up to $k = o(|E|)$. This is because the inconsistency can be "shifted around" the graph to any equation outside of the local scope. Thus, for every set X of at most k variables, there is at least one robustly consistent assignment with domain X. Robustly consistent assignments are also not discarded by the k-consistency procedure. In particular, k-consistency always accepts $\mathcal{C}^\lambda$ even if it has no solution.

▶ **Lemma 16** (consequence of [8, Lemma 4.6]). *If $k \in o(|E|)$ and Γ is a p-group (i.e., $|\Gamma|$ is a power of p), and $\lambda\colon V \to \Gamma$, then there is a p-solution of $\mathsf{L}_{\mathsf{CSP}}^{k,\Gamma^{[3]}}(\mathcal{C}^\lambda)$ such that non-robustly consistent partial assignments are set to 0, and each robustly consistent partial solution is mapped to $1/p^\ell$ for some $\ell \in \mathbb{N}$.*

The above lemma can also be refined so that the resulting p-solution assigns the value 1 to $x_{X,f}$ for a single robustly consistent partial homomorphism $f\colon X \to \Gamma$ of our choice.

5 Limitations of the Affine Algorithms

All of the affine algorithms are *sound*: they accept all yes-instances. This section shows that many of them are not *complete* on tractable CSPs: they do not reject all no-instances, and thus do not solve the CSP. We consider the tractable homomorphism or-construction $\mathbf{OR}_{\mathrm{T}}[\mathbb{Z}_2^{[3]}, \mathbb{Z}_3^{[3]}]$ of the ternary $\mathbb{Z}_2$-coset-CSP and the ternary $\mathbb{Z}_3$-coset-CSP.

5.1 $\mathbb{Z}$-Affine k-Consistency Relaxation

The $\mathbb{Z}$-*affine k-consistency relaxation* [20] solves the following system of affine linear equations over the integers. Let $\mathbf{A}$ be a template, $\mathbf{B}$ be an instance, and κ be a map assigning to every set $X \in \binom{B}{\leq k}$ a set of partial homomorphisms $\mathbf{B}[X] \to \mathbf{A}$. Define the system $\mathsf{L}_{\mathbb{Z}\text{-aff}}^{k,\mathbf{A}}(\mathbf{B}, \kappa)$:

$\mathsf{L}_{\mathbb{Z}\text{-aff}}^{k,\mathbf{A}}(\mathbf{B}, \kappa)$: variables $z_{X,f}$ for all $X \in \binom{B}{\leq k}$ and $f \in \kappa(X)$

$$z_{X,f} \in \mathbb{Z} \qquad \text{for all } X \in \tbinom{B}{\leq k} \text{ and } f \in \kappa(X)$$

$$\sum_{f \in \kappa(X)} z_{X,f} = 1 \qquad \text{for all } X \in \tbinom{B}{\leq k}$$

$$\sum_{f \in \kappa(X), f|_Y = g} z_{X,f} = z_{Y,g} \qquad \text{for all } Y \subset X \in \tbinom{B}{\leq k} \text{ and } g \in \kappa(Y)$$

For a fixed positive integer k, a template structure $\mathbf{A}$, and an instance $\mathbf{B}$, the $\mathbb{Z}$-affine k-consistency relaxation runs the k-consistency algorithm to compute $\kappa_k^{\mathbf{A}}[\mathbf{B}]$, the function that maps each set $X \in \binom{B}{\leq k}$ to the set of k-consistent partial homomorphisms $\mathbf{B}[X] \to \mathbf{A}$. The instance $\mathbf{B}$ is accepted by the algorithm if $\mathsf{L}_{\mathbb{Z}\text{-aff}}^{k,\mathbf{A}}(\mathbf{B}, \kappa_k^{\mathbf{A}}[\mathbf{B}])$ has an integral solution and rejects $\mathbf{B}$ otherwise. Dalmau and Opršal [20] conjectured the following on the power of the $\mathbb{Z}$-affine k-consistency relaxation:

▶ **Conjecture 17** ([20]). *For every finite structure $\mathbf{A}$, either $\mathrm{CSP}(K_3)$ is Datalog$^\cup$-reducible to $\mathrm{CSP}(\mathbf{A})$ or $\mathrm{CSP}(\mathbf{A})$ is Datalog$^\cup$-reducible to $\mathrm{CSP}(\mathbb{Z})$, where K_3 denotes the triangle.*

Being Datalog$^\cup$-reducible to $\mathrm{CSP}(\mathbb{Z})$ implies that $\mathrm{CSP}(\mathbf{A})$ is solved by $\mathbb{Z}$-affine k-consistency for some constant k [20], which we show is not the case for our counterexample.

▶ **Theorem 18.** *For every $k \geq 1$, the $\mathbb{Z}$-affine k-consistency relaxation does not solve $\mathbf{OR}_T[\mathbb{Z}_2^{[3]}, \mathbb{Z}_3^{[3]}]$. This is even true if k is not a constant, but an at most sublinear function in the instance size.*

Proof. Let $(G_n)_{n\in\mathbb{N}}$ be a family of 3-regular 2-connected expander graphs and fix $G := G_n$, for a large enough $n \in \mathbb{N}$, such that $|V(G)|$ is sufficiently larger than k. Let H be an orientation of G. Let $p_1 := 2$ and $p_2 := 3$. For $i \in [2]$, let $\Gamma_i := \mathbb{Z}_{p_i}$, and $\lambda_i : V(G) \to \Gamma_i$ be 0 everywhere except at one vertex $v^* \in V(G)$, where we set $\lambda_i(v^*) := 1$. For each

$i \in [2]$, we consider the 3-ary Γ_i-coset-CSP $\mathbf{B}_i := \mathcal{C}^{H,\Gamma_i,\lambda_i}$. Let $\mathbf{B} := \mathbf{OR}[\mathbf{B}_1, \mathbf{B}_2]$ and $\mathbf{A} := \mathbf{OR}_{\mathrm{T}}[\mathbb{Z}_2^{[3]}, \mathbb{Z}_3^{[3]}]$ be the corresponding tractable homomorphism or-template. From $\sum_{v \in V(G)} \lambda_i(v) \neq 0$ it follows $\mathbf{B}_i \notin \mathrm{CSP}(\Gamma_i^{[3]})$ for both $i \in [2]$ and thus $\mathbf{B} \notin \mathrm{CSP}(\mathbf{A})$ by Lemma 7. For a set $X \subseteq B$, a partial homomorphism $f \colon \mathbf{B}[X] \to \mathbf{A}$ is robustly consistent if $f|_{B_i}$ is a robustly consistent partial homomorphism $\mathbf{B}_i[X \cap B_i] \to \mathbf{A}_i$ for some $i \in [2]$ and $f(X \cap B_{3-i}) = \{c_{3-i}\}$, where c_{3-i} is the fresh vertex of the or-construction. Robustly consistent partial solutions of the Tseitin systems are not discarded by k-consistency, and by Lemma 8 this is also true for the robustly consistent partial homomorphisms of the or-instance. Then $\mathsf{L}_{\mathsf{CSP}}^{k,\mathbf{A}_i}(\mathbf{B}_i)$ has a p_i-solution for both $i \in [2]$ by Lemma 16, which is only non-zero for variables indexed by robustly consistent partial homomorphisms. By Lemma 9, $\mathsf{L}_{\mathsf{CSP}}^{k,\mathbf{A}}(\mathbf{B})$ has a p_i-solution in which only robustly consistent partial homomorphisms are non-zero, too. By Lemma 5, there is an integral solution to $\mathsf{L}_{\mathsf{CSP}}^{k,\mathbf{A}}(\mathbf{B})$, which is only non-zero for robustly consistent partial solutions of $\mathbf{B}$. Such solutions to $\mathsf{L}_{\mathsf{CSP}}^{k,\mathbf{A}}(\mathbf{B})$ imply solutions to $\mathsf{L}_{\mathbb{Z}\text{-aff}}^{k,\mathbf{A}}(\mathbf{B}, \kappa_k^{\mathbf{A}}[\mathbf{B}])$. Hence, the $\mathbb{Z}$-affine k-consistency relaxation wrongly accepts $\mathbf{B}$. ◀

▶ **Lemma 19.** $\mathrm{CSP}(K_3)$ *is not Datalog*$^\cup$*-reducible to* $\mathrm{CSP}(\mathbf{OR}_T[\mathbb{Z}_2^{[3]}, \mathbb{Z}_3^{[3]}])$.

Theorem 18 and Lemma 19 disprove Conjecture 17. To show the lemma, we show that $\mathrm{CSP}(\mathbb{Z}_p^{[r]})$ is not Datalog$^\cup$-reducible to $\mathrm{CSP}(\mathbf{OR}_{\mathrm{T}}[\mathbb{Z}_2^{[3]}, \mathbb{Z}_3^{[3]}])$ for a prime $p \notin \{2,3\}$. This suffices because $\mathrm{CSP}(\mathbb{Z}_p^{[r]})$ is Datalog$^\cup$-reducible to $\mathrm{CSP}(K_3)$ [20]. We use results from finite model theory: $\{2,3\}$-rank logic [21] extends IFP by a rank operator for definable matrices over $\mathbb{Z}_2$ and $\mathbb{Z}_3$. The logic defines $\mathrm{CSP}(\mathbb{Z}_2^{[3]})$ and $\mathrm{CSP}(\mathbb{Z}_3^{[3]})$ and so also $\mathrm{CSP}(\mathbf{OR}_{\mathrm{T}}[\mathbb{Z}_2^{[3]}, \mathbb{Z}_3^{[3]}])$ by Corollary 12. If $\mathrm{CSP}(\mathbb{Z}_p^{[r]})$ is Datalog$^\cup$-reducible to $\mathrm{CSP}(\mathbf{OR}_{\mathrm{T}}[\mathbb{Z}_2^{[3]}, \mathbb{Z}_3^{[3]}])$, then $\{2,3\}$-rank logic solves $\mathrm{CSP}(\mathbb{Z}_p^{[r]})$ because IFP expresses Datalog$^\cup$-reductions. But $\mathbb{Z}_p$-equation systems are only solvable in $\{2,3\}$-rank logic if $p \in \{2,3\}$ [26].

5.2 BLP+AIP and BA^k

We introduce another well-studied system of equations for CSPs [3, 10] parameterized by the size of partial solutions [15]. Let k be a positive integer, $\mathbf{A}$ a template τ-structure and $\mathbf{B}$ a $\mathrm{CSP}(\mathbf{A})$-instance. We define the system $\mathsf{L}_{\mathsf{IP}}^{k,\mathbf{A}}(\mathbf{B})$ with variable set $\mathcal{V}^{k,\mathbf{A}}(\mathbf{B})$.

$\mathsf{L}_{\mathsf{IP}}^{k,\mathbf{A}}(\mathbf{B})$: variables $\lambda_{X,f}$ for all $X \in \binom{B}{\leq k}$ and $f \colon X \to A$, and variables $\mu_{R,\bar{b},\bar{a}}$ for all $R \in \tau$, $\bar{b} \in R^{\mathbf{B}}$, and $\bar{a} \in R^{\mathbf{A}}$

$$\sum_{f \colon X \to A} \lambda_{X,f} = 1 \qquad \text{for all } X \in \tbinom{B}{\leq k},$$

$$\sum_{\substack{f \colon X \to A, \\ f|_Y = g}} \lambda_{X,f} = \lambda_{Y,g} \qquad \text{for all } Y \subset X \in \tbinom{B}{\leq k}, g \colon Y \to A,$$

$$\sum_{\bar{a} \in R^{\mathbf{A}}, a_{\bar{i}} = \bar{a}'} \mu_{R,\bar{b},\bar{a}} = \lambda_{X(\bar{b}_{\bar{i}}), \bar{b}_{\bar{i}} \mapsto \bar{a}'} \qquad \text{for all } R \in \tau, \bar{a}' \in A^k, \bar{b} \in R^{\mathbf{B}}, \bar{i} \in [\mathrm{ar}(R)]^k,$$

where $a_{\bar{i}}$ and $b_{\bar{i}}$ are the k-tuples $(a_{\bar{i}_1}, \ldots, a_{\bar{i}_k})$ and $(b_{\bar{i}_1}, \ldots, b_{\bar{i}_k})$, respectively, $X(\bar{b}_{\bar{i}})$ is the set of entries of $\bar{b}_{\bar{i}}$, and $\bar{b}_{\bar{i}} \mapsto \bar{a}'$ is the function $X(\bar{b}_{\bar{i}}) \to A$ mapping $\bar{b}_{\bar{i}}$ to $\bar{a}'$.

Different domains of the variables (see [10]) are of interest: If we restrict the variables to $\{0,1\}$, then $\mathsf{L}_{\mathsf{IP}}^{1,\mathbf{A}}(\mathbf{B})$ is solvable if and only if $\mathbf{B} \in \mathrm{CSP}(\mathbf{A})$. The relaxation of $\mathsf{L}_{\mathsf{IP}}^{k,\mathbf{A}}(\mathbf{B})$ to nonnegative rationals is the *k-basic linear programming (BLP)* relaxation $\mathsf{L}_{\mathsf{BLP}}^{k,\mathbf{A}}(\mathbf{B})$[1]. The affine relaxation

[1] The literature [10, 16] only calls $\mathsf{L}_{\mathsf{BLP}}^{1,\mathbf{A}}(\mathbf{B})$ the basic linear programming relaxation. For convenience and uniformity, we extend the notion in this paper to arbitrary k.

of $\mathsf{L}_{\mathsf{IP}}^{k,\mathbf{A}}(\mathbf{B})$ to all integers is the *k-affine integer programming (AIP)* relaxation $\mathsf{L}_{\mathsf{AIP}}^{k,\mathbf{A}}(\mathbf{B})$. By increasing the parameter k, the BLP and AIP relaxations result in the Sherali-Adams LP hierarchy [30] and the affine integer programming hierarchy [15] of the $\{0,1\}$-system, respectively. In contrast to the $\mathbb{Z}$-affine k-consistency relaxation, the BLP+AIP algorithm is not parameterized by the size of partial solutions k. Ciardo and Živný [18, 16] proposed this parameterized version called BA^k, where BA^1 is just the BLP+AIP algorithm.

$\mathsf{BA}^k(\mathbf{A})$-algorithm: input a CSP($\mathbf{A}$)-instance $\mathbf{B}$

1. Compute a relative interior point $\Phi\colon \mathcal{V}^{k,\mathbf{A}}(\mathbf{B}) \to \mathbb{Q}$ in the polytope defined by $\mathsf{L}_{\mathsf{BLP}}^{k,\mathbf{A}}(\mathbf{B})$. The solution Φ has in particular the property that for each variable $x \in \mathcal{V}^{k,\mathbf{A}}(\mathbf{B})$ there is a solution Ψ to $\mathsf{L}_{\mathsf{BLP}}^{\mathbf{A}}(\mathbf{B})$ such that $\Psi(x) \neq 0$ if and only if $\Phi(x) \neq 0$. If such a point does not exist, reject.
2. Refine $\mathsf{L}_{\mathsf{AIP}}^{k,\mathbf{A}}(\mathbf{B})$ by adding the constraints $x = 0$ whenever $\Phi(x) = 0$ for all $x \in \mathcal{V}^{k,\mathbf{A}}(\mathbf{B})$.
3. If the refined system is feasible (over $\mathbb{Z}$), then accept, otherwise reject.

The original presentation of BA^k [18] uses a slightly different system of equations but one can easily verify that our presentation is equisatisfiable. The system in [18] does not have variables $\lambda_{X,f}$ but uses variables $\lambda_{R_k,\bar{b},\bar{a}}$ instead, where R_k is the full k-ary relation. We deviate from the presentation in [18] to keep it consistent with the systems for the other algorithms. We show that BA^k fails on the counterexample provided for $\mathbb{Z}$-affine k-consistency.

▶ **Theorem 20.** *For every integer k, the algorithm $\mathsf{BA}^k(\mathbf{A})$ does not solve $\mathbf{OR}_T[\mathbb{Z}_2^{[3]}, \mathbb{Z}_3^{[3]}]$. This is even true if k is not a constant but an at most sublinear function in the instance size.*

The theorem is proved using the same construction as for Theorem 18. One shows, for k at least the arity of $\mathbf{A}$, that a non-negative or integral solution of $\mathsf{L}_{\mathsf{CSP}}^{k,\mathbf{A}}(\mathbf{B})$ implies a solution for $\mathsf{L}_{\mathsf{BLP}}^{k,\mathbf{A}}(\mathbf{B})$ or $\mathsf{L}_{\mathsf{AIP}}^{k,\mathbf{A}}(\mathbf{B})$, respectively. The solution to $\mathsf{L}_{\mathsf{BLP}}^{k,\mathbf{A}_i}(\mathbf{B}_i)$, where $\mathbf{B}_i$ is the Tseitin-system over $p_i = i+1$, is non-zero and non-negative exactly for the robustly consistent partial homomorphisms. This carries over to the notion of robustly consistent partial homomorphisms for the or-instance $\mathbf{B} = \mathbf{OR}[\mathbf{B}_1, \mathbf{B}_2]$. This implies that the interior point computed in Step 1 is non-zero for robustly consistent partial homomorphisms. So they are not set to zero in Step 2 and there is an integral solution.

5.3 The CLAP Algorithm

The CLAP algorithm [17] combines the BLP and the AIP relaxation. It first iteratively reduces the solution space using BLP by fixing partial solutions to 1 and discarding those for which this refined BLP is not solvable. Then BLP+AIP is run on the restricted solution space, where again a partial solution is fixed:

$\mathsf{CLAP}(\mathbf{A})$-algorithm: input a CSP($\mathbf{A}$)-instance $\mathbf{B}$

1. Maintain, for each relation symbol $R \in \tau$ and each tuple $\bar{b} \in R^{\mathbf{B}}$, a set $S_{\bar{b},R} \subseteq R^{\mathbf{A}}$ of possible images of $\bar{b}$ under a homomorphism. Initialize $S_{\bar{b},R} := R^{\mathbf{A}}$.
2. Repeat until no set $S_{\bar{b},R}$ changes anymore: For each $R \in \tau$, $\bar{b} \in R^{\mathbf{B}}$, and $\bar{a} \in S_{\bar{b},R}$, solve $\mathsf{L}_{\mathsf{BLP}}^{1,\mathbf{A}}(\mathbf{B})$ together with the following additional constraints:
$$\begin{aligned} \mu_{R,\bar{b},\bar{a}} &= 1, \\ \mu_{R',\bar{b}',\bar{a}'} &= 0 \qquad \text{for all } R' \in \tau, \bar{b}' \in R'^{\mathbf{B}}, \bar{a}' \notin S_{\bar{b}',R'}. \end{aligned}$$
If this system is not feasible, remove $\bar{a}$ from $S_{\bar{b},R}$.

3. If there are $R \in \tau$ and $\bar{b} \in R^{\mathbf{B}}$ such that $S_{\bar{b},R} = \emptyset$, then reject.
4. For each $R \in \tau$, $\bar{b} \in R^{\mathbf{B}}$, and $\bar{a} \in S_{\bar{b},R}$, execute $\mathsf{BA}^1(\mathbf{A})$ (which is BLP+AIP) on $\mathbf{B}$, where we additionally fix
$$\mu_{R,\bar{b},\bar{a}} = 1,$$
$$\mu_{R',\bar{b}',\bar{a}'} = 0 \qquad \text{for all } R' \in \tau, \bar{b}' \in R'^{\mathbf{B}}, \bar{a}' \notin S_{\bar{b}',R'}$$
in Step 1 of $\mathsf{BA}^1(\mathbf{A})$. If $\mathsf{BA}^1(\mathbf{A})$ accepts, then accept.
5. If $\mathsf{BA}^1(\mathbf{A})$ rejects all inputs in the step before, then reject.

Note that the algorithm accepts in Step 4 if for *one* $R \in \tau$, one $\bar{b} \in R^{\mathbf{B}}$, and one $\bar{a} \in S_{\bar{b},R}$ BLP+AIP accepts when fixing the partial solution $\bar{b} \mapsto \bar{a}$ (in contrast to the stronger requirement that for each $R \in \tau$ and $\bar{b} \in R^{\mathbf{B}}$, there is one $\bar{a} \in S_{\bar{b},R}$ for which BLP+AIP accepts). We show that this weaker requirement can actually by omitted and does not change whether CLAP solves CSP(A). This is crucial to show that CLAP fails on the same counterexample.

▶ **Theorem 21.** $\mathsf{CLAP}(\mathbf{A})$ *does not solve* $\mathrm{CSP}(\mathbf{OR}_T[\mathbb{Z}_2^{[3]}, \mathbb{Z}_3^{[3]}])$.

We prove this theorem again with the construction from the proof of Theorem 18, where we pick k to be the arity of $\mathbf{A}$. Again we have to argue that the pruning steps never remove robustly consistent partial homomorphisms. Here we also need the property of the Tseitin systems that their relaxation admits a p_i-solution even if we require a given partial homomorphism to receive value 1. Theorems 18, 20, and 21 **prove Theorem 1**. Our arguments do not exploit that CLAP uses $\mathsf{L}^{1,\mathbf{A}}_{\mathsf{BLP}}(\mathbf{B})$ and $\mathsf{BA}^1(\mathbf{A})$ and is not parametrized by a width k. Hence, a parameterized version of CLAP will fail on the same counterexample.

5.4 The Cohomological k-Consistency Algorithm

The cohomological k-consistency algorithm due to Ó Conghaile [19] was originally presented in categorical and cohomological notions but computing cohomology groups is nothing else than solving a system of linear equations over the integers. Therefore, we can state the algorithm in a way similar to the other algorithms seen so far.

Cohomological k-consistency algorithm: input a CSP($\mathbf{A}$)-instance $\mathbf{B}$

1. Maintain, for each $X \in \binom{B}{\leq k}$, a set $\mathcal{H}(X) \subseteq \mathrm{Hom}(\mathbf{B}[X], \mathbf{A})$. Initialize them to be $\mathcal{H}(X) := \mathrm{Hom}(\mathbf{B}[X], \mathbf{A})$.
2. Repeat until none of the sets $\mathcal{H}(X)$ changes anymore:
   a. Run the k-consistency algorithm on $\mathcal{H}$ and remove from each $\mathcal{H}(X)$ the partial homomorphisms that fail the forth-condition or down-closure property.
   b. For each $X \in \binom{B}{\leq k}$ and $f \in \mathcal{H}(X)$, check whether $\mathsf{L}^{k,\mathbf{A}}_{\mathbb{Z}\text{-aff}}(\mathbf{B}, \mathcal{H})$ has a solution that satisfies $x_{X,f} = 1$ and $x_{X,f'} = 0$ for every $f' \in \mathcal{H}(X) \setminus \{f\}$. If it does not, then remove f from $\mathcal{H}(X)$ for the next iteration of the loop.
3. If $\mathcal{H}(X) = \emptyset$ for some $X \in \binom{B}{\leq k}$, then reject; otherwise accept.

Step 2b of the algorithm approximates whether there is a global homomorphism whose restriction to X is equal to f via solving the $\mathsf{L}^{k,\mathbf{A}}_{\mathbb{Z}\text{-aff}}(\mathbf{B}, \mathcal{H})$ in which we set $x_{X,f} = 1$ and $x_{X,f'} = 0$ for all other f'. At least for the template $\mathrm{CSP}(\mathbf{OR}_{\mathrm{T}}[\mathbb{Z}_2^{[3]}, \mathbb{Z}_3^{[3]}])$, the cohomological k-consistency algorithm is strictly more powerful than the previous ones because it correctly rejects the instances:

▶ **Theorem 22.** *If $\mathbf{A}_1, \mathbf{A}_2$ are templates of Abelian coset-CSPs and $k \geq \mathrm{ar}(A_i) + 1$ for both $i \in [2]$, then the cohomological k-consistency algorithm solves* $\mathrm{CSP}(\mathbf{OR}_T[\mathbf{A}_1, \mathbf{A}_2])$.

The proof exploits Lemma 13: When a partial homomorphism of $\mathbf{B}_i$ is fixed as in Step 2(b) of the algorithm, then the tractable homomorphism or-construction is resolved: $\mathsf{L}^{k,\mathbf{A}}_{\mathbb{Z}\text{-aff}}(\mathbf{B}, \mathcal{H})$ (with the additional constraints) has an integral solution only if $\mathsf{L}^{k,\mathbf{A}_i}_{\mathbb{Z}\text{-aff}}(\mathbf{B}_i, \mathcal{H})$ has an integral solution. But for Abelian coset-CSPs, $\mathsf{L}^{k,\mathbf{A}_i}_{\mathbb{Z}\text{-aff}}(\mathbf{B}_i, \mathcal{H})$ has an integral solution if and only if $\mathbf{B}_i \in \mathrm{CSP}(\mathbf{A}_i)$ (see Section 6). This means that there is no integral solution and the algorithm rejects $\mathbf{B}$. Actually, showing that the cohomological algorithm correctly solves all instances also follows the algorithm for Lemma 11. While the algorithm correctly solves the tractable homomorphism or-construction, it fails on the *intractable* one. This proves *without* complexity-theoretic assumptions like $\mathrm{P} \neq \mathrm{NP}$ that this polynomial-time algorithm does not solve all finite-domain CSPs.

▶ **Theorem 23.** *There is an NP-complete template structure $\mathbf{A}$ such that for every constant k, the cohomological k-consistency algorithm does not solve* $\mathrm{CSP}(\mathbf{A})$.

The theorem is proved using the same setup as in the counterexample for the other algorithms, but we use the intractable homomorphism or-construction. This makes the crucial difference that, by Lemma 14, two partial homomorphisms of both $\mathbf{B}_1 \to \mathbf{A}_1$ and $\mathbf{B}_2 \to \mathbf{A}_2$ can be combined into a partial homomorphism $\mathbf{B} \to \mathbf{A}$. If both partial homomorphisms are robustly consistent, then also their combination is not ruled out by k-consistency. When a partial solution is fixed in Step 2, there is still a p_i-solution for both $\mathbf{B}_i$. Hence there is an integral solution by Lemma 5. Thus, all robustly consistent assignments survive Step 2. As indicated in Section 3, $\mathrm{CSP}(\mathbf{A})$ is NP-complete. Theorems 22 and 23 **prove Theorem 3**.

6 Affine Algorithms and Coset-CSPs

The counterexample we have used so far is not a coset-CSP itself, but a combination of two Abelian coset-CSPs in the homomorphism or-construction. We now set out to explore the power of the affine algorithms on coset-CSPs. It follows from [3] that solving $\mathsf{L}^{k,\mathbf{A}}_{\mathsf{AIP}}(\mathbf{B})$ for $k = 1$ (often just called AIP) suffices to solve all Abelian coset-CSPs. Since all algorithms considered are at least as powerful as AIP, this **proves Theorem 2 (1)**. But there are non-Abelian groups for which AIP still works: These are 2-nilpotent groups Γ of odd order, for example certain non-Abelian semidirect products $\mathbb{Z}_{p^2} \rtimes \mathbb{Z}_p$ for each odd prime p. For these groups, the Maltsev operation $f(x, y, z) = x - y + z$ for a certain Abelian group Δ is a polymorphism of the non-Abelian template. It follows again from [3] that AIP solves the Γ-coset-CSP. We thank Michael Kompatscher for this proof idea. This **proves Theorem 2 (3)**.

It remains to **show Theorem 2 (2)**, i.e., that the affine algorithms studied in Section 5 also fail on coset-CSPs. The key idea is to translate our counterexample from Section 5 into a coset-CSP such that hardness for the algorithms is preserved. This is achieved via a series of reduction steps: We start again with Tseitin systems $\mathbf{B}_1$ and $\mathbf{B}_2$ over $\mathbb{Z}_2$ and $\mathbb{Z}_3$, respectively.

These systems, like every coset-CSP, can be reduced, by a variant of the well-known CFI construction [13], to the *graph isomorphism problem for graphs of bounded color class size* [8]. These are vertex-colored graphs in which only a constant number of vertices have the same color. The reduction gives for each $i \in [2]$ two colored graphs $\mathbf{G}^{\mathbf{B}_i}_{\mathbb{Z}_{p_i}}$ and $\widetilde{\mathbf{G}}^{\mathbf{B}_i}_{\mathbb{Z}_{p_i}}$ that are isomorphic if and only if $\mathbf{B}_i \in \mathrm{CSP}(\mathbb{Z}^{[3]}_{p_i})$. Next, we use an *isomorphism or-construction*, similar to the one in [8]. This yields two colored graphs which are isomorphic if and only if for at least one $i \in [2]$ we have $\mathbf{G}^{\mathbf{B}_i}_{\mathbb{Z}_{p_i}} \cong \widetilde{\mathbf{G}}^{\mathbf{B}_i}_{\mathbb{Z}_{p_i}}$. Finally, this isomorphism problem is expressed as a binary S_d-coset-CSP [8], where d is the size of the color classes, which is 18 in our case. This S_{18}-coset CSP has a solution if and only if at least one of the initial two Tseitin instances has a solution – which by construction is not the case.

The technical difficulty is showing that the hardness of the instances is preserved via the reduction steps. One can indeed show that the translations from group-coset-CSPs to bounded color class size graph isomorphism and back as well as the isomorphism or-construction preserve solutions of the width-k affine relaxation and k-consistency in a similar way as we did this for the homomorphism or-construction in Section 3. In the end, we are essentially in the same setting as in the proofs in Section 5. We find a suitable notion of robustly consistent partial homomorphisms which are not ruled out by k-consistency. We also get p_i-solutions to the width-k affine relaxation of the S_{18}-coset-CSP, which are only non-zero for robustly consistent partial homomorphisms. Thus, using Lemma 5, we obtain an integral solution. By following the same reasoning as in Section 5, we can show that neither the $\mathbb{Z}$-affine k-consistency relaxation, BA^k, nor CLAP solve $\mathrm{CSP}(S_{18}^{[2]})$. We provide a full proof including a description of the reduction steps in the full version.

7 Conclusion

Regarding the question for a universal polynomial-time CSP algorithm, we conclude that most of the affine algorithms from recent years are not powerful enough. Only *cohomological k-consistency* remains as a candidate because it sets local solutions to 1 when solving the affine relaxation. We are aware of another algorithm with this feature: This is C(BLP+AIP), a variation of CLAP mentioned in [17] and defined more explicitly in [14]. This algorithm also involves solving the integer relaxation where additionally a local solution is set to 1. We expect that it solves our counterexample, too. In [14], it is shown that C(BLP+AIP) fails on certain intractable templates but it remains an intriguing open question to find a *tractable* counterexample for C(BLP+AIP) and cohomological k-consistency. Another question that we have not addressed is the relationship between the different algorithms. It is obvious from the definitions that cohomological k-consistency subsumes $\mathbb{Z}$-affine k-consistency, and that CLAP subsumes BLP+AIP. How the k-consistency based methods compare to the BLP-based ones remains unanswered; it may be that they are incomparable. In particular, we would like to know if cohomological k-consistency strictly subsumes all the other algorithms. In light of our results, this seems likely, but since the cohomological algorithm does not use the BLP, it is not obvious how it compares to, say, BA^k.

References

1 Albert Atserias, Andrei A. Bulatov, and Víctor Dalmau. On the power of k-consistency. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings*, volume 4596 of *Lecture Notes in Computer Science*, pages 279–290. Springer, 2007. doi:10.1007/978-3-540-73420-8_26.

2 Libor Barto. The collapse of the bounded width hierarchy. *Journal of Logic and Computation*, 26(3):923–943, November 2014. doi:10.1093/logcom/exu070.

3 Libor Barto, Jakub Bulín, Andrei A. Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. *J. ACM*, 68(4):28:1–28:66, 2021. doi:10.1145/3457606.

4 Libor Barto and Marcin Kozik. Constraint satisfaction problems of bounded width. In *2009 50th Annual IEEE symposium on foundations of computer science*, pages 595–603. IEEE, 2009. doi:10.1109/FOCS.2009.32.

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 Christoph Berkholz and Martin Grohe. Limitations of algebraic approaches to graph isomorphism testing. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 155–166. Springer, 2015. doi:10.1007/978-3-662-47672-7_13.

7 Christoph Berkholz and Martin Grohe. Linear Diophantine equations, group CSPs, and graph isomorphism. *CoRR*, abs/1607.04287, 2016. arXiv:1607.04287.

8 Christoph Berkholz and Martin Grohe. Linear Diophantine equations, group CSPs, and graph isomorphism. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 327–339. SIAM, 2017. doi:10.1137/1.9781611974782.21.

9 Joshua Brakensiek and Venkatesan Guruswami. An algorithmic blend of LPs and ring equations for promise CSPs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 436–455. SIAM, 2019. doi:10.1137/1.9781611975482.28.

10 Joshua Brakensiek, Venkatesan Guruswami, Marcin Wrochna, and Stanislav Živný. The power of the combined basic LP and affine relaxation for promise CSPs. *Electron. Colloquium Comput. Complex.*, TR20-004, 2020. URL: https://eccc.weizmann.ac.il/report/2020/004.

11 Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, 2017. doi:10.1109/FOCS.2017.37.

12 Andrei A. Bulatov and Víctor Dalmau. A simple algorithm for Mal'tsev constraints. *SIAM J. Comput.*, 36(1):16–27, 2006. doi:10.1137/050628957.

13 Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992. doi:10.1007/BF01305232.

14 Siu On Chan and Hiu Tsun Ng. How random CSPs fool hierarchies: II. *Electron. Colloquium Comput. Complex.*, TR25-026, 2025. To appear at STOC 2025. URL: https://eccc.weizmann.ac.il/report/2025/026.

15 Lorenzo Ciardo and Stanislav Živný. Approximate graph colouring and crystals. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2256–2267. SIAM, 2023. doi:10.1137/1.9781611977554.CH86.

16 Lorenzo Ciardo and Stanislav Živný. Approximate graph colouring and the hollow shadow. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 623–631. ACM, 2023. doi:10.1145/3564246.3585112.

17 Lorenzo Ciardo and Stanislav Živný. CLAP: A new algorithm for promise CSPs. *SIAM J. Comput.*, 52(1):1–37, 2023. doi:10.1137/22M1476435.

18 Lorenzo Ciardo and Stanislav Živný. Hierarchies of minion tests for PCSPs through tensors. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 568–580. SIAM, 2023. doi:10.1137/1.9781611977554.CH25.

19 Adam Ó Conghaile. Cohomology in constraint satisfaction and structure isomorphism. In Stefan Szeider, Robert Ganian, and Alexandra Silva, editors, *47th International Symposium on Mathematical Foundations of Computer Science, MFCS 2022, August 22-26, 2022, Vienna, Austria*, volume 241 of *LIPIcs*, pages 75:1–75:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.MFCS.2022.75.

20 Víctor Dalmau and Jakub Opršal. Local consistency as a reduction between constraint satisfaction problems. In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 29:1–29:15. ACM, 2024. doi:10.1145/3661814.3662068.

21 Anuj Dawar, Martin Grohe, Bjarki Holm, and Bastian Laubner. Logics with rank operators. In *24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, 11-14 August 2009, Los Angeles, CA, USA*, pages 113–122. IEEE Computer Society, 2009. doi:10.1109/LICS.2009.24.

22 Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Perspectives in Mathematical Logic. Springer, Berlin, Heidelberg, 1995. doi:10.1007/978-3-662-03182-7.

23 Tomás Feder and Moshe Y. Vardi. Monotone monadic SNP and constraint satisfaction. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 612–622, 1993. doi:10.1145/167088.167245.

24 Merrick L. Furst, John E. Hopcroft, and Eugene M. Luks. Polynomial-time algorithms for permutation groups. In *21st Annual Symposium on Foundations of Computer Science, Syracuse, New York, USA, 13-15 October 1980*, pages 36–41. IEEE Computer Society, 1980. doi:10.1109/SFCS.1980.34.

25 Mikael Goldmann and Alexander Russell. The complexity of solving equations over finite groups. *Information and Computation*, 178(1):253–262, 2002. doi:10.1006/inco.2002.3173.

26 Erich Grädel and Wied Pakusa. Rank logic is dead, long live rank logic! *J. Symb. Log.*, 84(1):54–87, 2019. doi:10.1017/jsl.2018.33.

27 Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, 1979. doi:10.1137/0208040.

28 Moritz Lichter and Benedikt Pago. Limitations of affine integer relaxations for solving constraint satisfaction problems, 2025. doi:10.48550/arXiv.2407.09097.

29 Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.

30 Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990. doi:10.1137/0403036.

31 Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 695–704, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2488608.2488697.

32 Grigori Samuilovitsch Tseitin. *On the Complexity of Derivation in Propositional Calculus*, pages 466–483. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983. doi:10.1007/978-3-642-81955-1_28.

33 Dmitriy Zhuk. The proof of CSP dichotomy conjecture. *Journal of the ACM*, 67, April 2017. doi:10.1145/3402029.