ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-37

Finite Unary Relations and Qualitative Constraint Satisfaction

Peter Jonsson[1]

Abstract. Extending qualitative CSPs with the ability of restricting selected variables to finite sets of possible values has been proposed as an important research direction with important applications. Complexity results for this kind of formalisms have appeared in the literature but they focus on concrete examples and not on general principles. We propose three general methods. The first two methods are based on analysing the given CSP from a model-theoretical perspective, while the third method is based on directly analysing the growth of the representation of solutions. We exemplify our methods on temporal and spatial formalisms including Allen's algebra and RCC5.

1 INTRODUCTION

Qualitative reasoning has a long history in artificial intelligence and the combination of qualitative reasoning and constraint reasoning has been a very productive field. A large number of constraint-based formalisms for qualitative reasoning have been invented, most notably within temporal and spatial reasoning, and they have been investigated from many different angles. Recently, there has been a strong interest in combining different qualitative CSPs. Wolter and Zakharyaschev [45] refer to temporal and spatial reasoning when they write the following motivation.

> The next apparent and natural step would be to combine these two kinds of reasoning.
> The importance of such a step for both theory and applications is beyond any doubt.

It has also been noted that another (but related) line of research is highly relevant. Cohn and Renz [17] write the following.

> One problem with this [constraint-based] approach is that spatial entities are treated as variables which have to be instantiated using values of an infinite domain. How to integrate this with settings where some spatial entities are known or can only be from a small domain is still unknown and is one of the main future challenges of constraint-based spatial reasoning.

That is, they regard the question of how to extend constraint formalisms (in particular, spatial formalisms) with constants and other unary relations[2] as being very important; the same observation has been made in a wider context by Kreutzmann and Wolter [32]. Unfortunately, this question has not received the same amount of attention as the question of how to handle combined formalisms. Let us consider finite-domain CSPs for a moment so let D denote a finite domain of values and let $D_f = \{U \mid U \subseteq D\}$, i.e. the finite set of unary relations over D. For every finite constraint language Γ over D, the computational complexity of CSP($\Gamma \cup D_f$) is known due to results by Bulatov [10]. This is an important complexity results in finite-domain constraint satisfaction and it it has been reproven several times using different methods [1, 11]. The situation is radically different when considering infinite-domain CSPs where similar powerful results are not known. This can, at least partly, be attributed to the fact that infinite-domain CSPs constitute a much richer class of problems than finite-domain CSPs: for every computational problem X, there is an infinite-domain constraint language Γ_X such that X and CSP(Γ_X) are polynomial-time Turing equivalent [3]. For finite domain CSPs, we know that the problem is in NP and that the majority of computational problems cannot be captured by finite-domain CSPs.

Nevertheless, there are concrete examples where interesting qualitative and/or infinite-domain CSPs have been extended with finite unary relations. A very early example is the article by Jonsson & Bäckström [28] where several temporal formalisms (including the point algebra and Allen's interval algebra) are extended by unary relations (and also other relations). A more recent example is the article by Li et al. [35] where the point algebra and Allen's algebra are once again considered, together with the cardinal relation algebra, and RCC-5 and RCC-8 with two-dimensional polygonal regions, The results for the temporal formalisms by Jonsson & Bäckström are not completely comparable with the results by Li et al.: Jonsson & Bäckström's approach is based on linear programming while Li et al. use methods based on enforcing consistency. Consistency-enforcing methods have certain advantages such as lower time complexity and easier integration with existing constraint solving methods. At the same time, the linear programming method allows for more expressive extensions with retained tractability. Both consistency-based and LP-based methods have attracted attention lately, cf. Giannakopoulou et al. [22] and Kreutzmann and Wolter [32], respectively, and generalisations of the basic concepts have been proposed and analysed by de Leng and Heintz [18].

Our approach is different: instead of studying concrete examples, we study basic principles and aim at providing methods that are applicable to various constraint formalisms. We present three different methods. The first two methods are based on analysing the given CSP from a model-theoretical perspective, i.e. we investigate properties such as model-completeness and homogeneity. The third method is more of a toolbox for proving that the size of solutions grows in a controlled way, and that problems consequently are in NP. We illustrate the methods on both temporal and spatial formalisms (including Allen's algebra and RCC-5). The reader may find it strange that we

[1] Department of Computer and Information Science, Linköping University, SE-58183 Linköoping, Sweden. Email: `peter.jonsson@liu.se`

[2] Finite unary relations are sometimes referred to as *landmarks* in the AI literature. We will use the standard mathematical term throughout the paper.

mostly consider extensions with constant relations. The explanation is the close connection between problems extended with constants and with finite unary relations: if one of them is in NP, then both are in NP (see Lemma 3). Most problems under consideration become NP-hard when adding unary relations containing at least three elements: for example, if the constraint language contains the disequality relation $\neq$, then NP-hardness follows from a straightforward reduction from 3-COLOURABILITY. However, this is not always true if we only add constants to the language. Thus, we can extract more information by considering constants instead of finite unary relations. The same viewpoint is taken by, for instance, Li et al. [35].

The paper has the following structure. We introduce the basic concepts from CSPs and logic together with some information about homomorphisms in Section 2. The three different methods are presented in Sections 3, 4, and 5, respectively. We conclude the paper with a brief discussion in Section 6.

2 PRELIMINARIES

This section is divided into three parts where we consider constraint satisfaction, logic, and automorphisms of relational structures, respectively.

2.1 Constraint satisfaction problems

We begin by presenting CSPs in terms of *homomorphisms*. This view is the most common in the literature on finite-domain CSP and it will provide us with certain advantages: some of the properties that we consider later on are inherently based on homomorphisms. One should note, however, that there is no fundamental difference with the more common AI viewpoint that constraint satisfaction is about assigning values to variables in a way that satisfy certain constraints. In fact, we will use both viewpoints in the sequel.

A *relational signature* τ is a set of *relation symbols* R_i with an associated *arities* $k_i \in \mathbb{N}$. A *(relational) structure* Γ *over relational signature* τ (also called τ*-structure*) is a set D_Γ (the *domain*) together with a relation $R_i^\Gamma \subseteq D_\Gamma^{k_i}$ for each relation symbol R_i of arity k_i. If the reference to the structure Γ is clear, we may omit the superscript in R_i^Γ. We sometimes use the shortened notation $\overline{x}$ for a vector $(x_1, \ldots, x_n)$ of any length.

Let Γ and Δ be τ-structures. A *homomorphism* from Γ to Δ is a function f from D_Γ to D_Δ such that for each n-ary relation symbol R in τ and each n-tuple $\overline{a} = (a_1, \ldots, a_n)$, if $\overline{a} \in R^\Gamma$, then $(f(a_1), \ldots, f(a_n)) \in R^\Delta$.

Let Γ be a (possibly infinite) structure with a (possibly infinite) relational signature τ. Then the *constraint satisfaction problem (CSP)* for Γ is the following computational problem.

$\mathrm{CSP}(\Gamma)$
INSTANCE: A τ-structure Δ.
QUESTION: Is there a homomorphism from Δ to Γ?

In the homomorphism perspective on CSPs, the structure Γ is typically called the *template* of the constraint satisfaction problem $\mathrm{CSP}(\Gamma)$. The reader should be aware that several different names are used in the literature; *constraint language* is probably the most common within AI.

A homomorphism from a given τ-structure Δ to Γ is called a *solution* of Δ for $\mathrm{CSP}(\Gamma)$. It is in general not clear how to represent solutions for $\mathrm{CSP}(\Gamma)$ on a computer; however, for the definition of the problem $\mathrm{CSP}(\Gamma)$ we do not need to represent solutions since we only have to decide the *existence* of solutions. To represent an input structure Δ of $\mathrm{CSP}(\Gamma)$, we need to fix a suitable representation of the relation symbols in the signature τ. We will see in the forthcoming sections that the choice of representation is very important. Given a particular representation of relation symbols, we let $||\Delta||$ denote the size of an input structure Δ.

Example 1 (k-COLOURABILITY). For $k \geq 1$, the k-COLOURABILITY problem is the computational problem of deciding for a given finite graph G whether the vertices can be coloured by k colours or not such that adjacent vertices get different colours. It is well-known that the k-colouring problem is NP-hard for $k \geq 3$ and tractable when $k \leq 2$. For $k \geq 1$, let K_k denote the complete loop-free graph on k vertices. We view undirected graphs as τ-structures where τ contains a single binary relation symbol E which denotes a symmetric and anti-reflexive relation. Then the k-COLOURABILITY problem can be viewed as $\mathrm{CSP}(\{K_k\})$.

Example 2 (Digraph acyclicity). Consider the problem CSP($\{<\}$) where $<$ is the binary order relation of the set $\mathbb{Q}$ of rational numbers. Let $G = (V, A)$ be a directed graph. It is easy to see that there is a homomorphism from G to $(\mathbb{Q}; <)$ if and only if G contains no directed cycle. Thus, CSP($\{<\}$) is solvable in polynomial time since cycle detection in directed graphs can be carried out in polynomial (in fact, linear) time.

Clearly, we can equivalently define the instances of the CSP(Γ) problem as a tuple (V, C) where V is a set of variables and C is a set of constraints of the form $R(x_{i_1}, \ldots, x_{i_k})$ where $R \in \Gamma$, k is the arity of R, and $x_{i_1}, \ldots, x_{i_k} \subseteq V$. In this case, a solution is a function from V to the domain of Γ satisfying $(f(x_{i_1}, \ldots, f(x_{i_k})) \in R$ for every $R(x_{i_1}, \ldots, x_{i_k}) \in C$.

Let D be a value domain with a particular representations and let $||d||$ denote the size of the representation of $d \in D$. We let $D_c = \{\{d\} \mid d \in D\}$ and $D_f = \{D' \subseteq D \mid D' \text{ is finite}\}$. Given a representation of the elements in D, we always represent the members of D_f as sets of elements in D and we may assume that the size of D_f is linear in the sizes of its elements. Other ways of representing D_f are possible but they are outside the scope of this paper. If Γ is a constraint language with domain D, then CSP($\Gamma \cup D_c\}$ is the problem CSP(Γ) extended with constants and CSP($\Gamma \cup D_f\}$ is the problem CSP(Γ) extended with finite unary relations. The next lemma is basically Proposition 1(iii) in Li et al. [35] extended to arbitrary constraint languages.

Lemma 3 *CSP($\Gamma \cup D_c$) is in NP if and only if CSP($\Gamma \cup D_f$) is in NP.*

Proof. There is a trivial polynomial-time reduction from CSP($\Gamma \cup D_c$) to CSP($\Gamma \cup D_f$) so we consider the other direction. Let $I = (V, C)$ be an arbitrary instance of CSP($\Gamma \cup D_f$). Assume I has a solution $s : V \to D$. Each constraint $U(x) \in C$ with $U \in D_f$ can be replaced by the constraint $\{s(v)\}(v)$. The resulting instance I' is an instance of CSP($\Gamma \cup D_c$), it is satisfiable, and $||I'|| \leq ||I||$. The problem CSP($\Gamma \cup D_c$) is in NP so the satisfiability of I' can be polynomial-time verified by a certificate X. A polynomial-time verifiable certificate for I is thus the tuple (I', X). □

Lemma 3 allows us to, for example, concentrate on CSP($\Gamma \cup D_c$) instead of CSP($\Gamma \cup D_f$) when proving membership in NP.

2.2 Logic

First-order formulas φ over the signature τ (or, in short, τ-formulas) are as usual inductively defined using the logical symbols of universal and existential quantification, disjunction, conjunction, negation, equality, bracketing, variable symbols and the symbols from τ. The semantics of a first-order formula over some τ-structure is defined in the ordinary Tarskian style. A τ-formula without free variables is called a τ-sentence. We write $\Gamma \models \varphi$ if and only if the τ-structure Γ is a model for the τ-sentence φ, that is, satisfies φ; this notation is lifted to sets of sentences in the usual way.

One can use first-order formulas over the signature τ to define relations over a given τ-structure Γ: for a formula $\varphi(x_1, \ldots, x_k)$ where $x_1, \ldots, x_k$ are the free variables of φ the corresponding relation R is the set of all k-tuples $(t_1, \ldots, t_k) \in D_\Gamma^k$ such that $\varphi(t_1, \ldots, t_k)$ is true in Γ. In this case we say that R is *first-order definable* over Γ. Note that our definitions are always parameter-free, i.e. we do not allow the use of domain elements in them. We say that the τ-structure Γ admits *quantifier elimination* if every relation with a first-order definition in Γ has a quantifier-free definition in Γ. We also say that a set of formulas T admit quantifier elimination if each $F \in T$ has a logically equivalent quantifier-free formula.

A first-order τ-formula $\phi(x_1, \ldots, x_n)$ is called *existential* if it is of the form

$$\exists x_{n+1}, \ldots, x_m . \psi$$

where ψ is a quantifier-free first-order formula. A subset of existential formulas is of particular interest to us: a first-order τ-formula $\phi(x_1, \ldots, x_n)$ is called *primitive positive* if it is of the form

$$\exists x_{n+1}, \ldots, x_m . \psi_1 \wedge \cdots \wedge \psi_l$$

where $\psi_1, \ldots, \psi_l$ are *atomic τ-formulas*, i.e., formulas of the form

1. $R(y_1, \ldots, y_k)$ with $R \in \tau$ and $y_i \in \{x_1, \ldots, x_m\}$ or
2. $y = y'$ for $y, y' \in \{x_1, \ldots, x_m\}$.

If the relation R has a primitive positive definition in Γ, then we say that R is *pp-definable* in Γ, and we define $\langle \Gamma \rangle$ to be the set of relations that are pp-definable in Γ. It is well-known [27] (and not hard to prove) that if Γ is a structure and a relation R is pp-definable in Γ, then there is a polynomial-time reduction from CSP($\Gamma \cup \{R\}$) to CSP(Γ). This explains why pp-definitions are important when studying the complexity of CSP problems. To exemplify, consider the constraint language $\Gamma = \{\{1, 2, 3, 4\}, \neq\}$ with the natural numbers as its domain. We see that the binary relation $K_4 = \{(x, y) \in \{1, 2, 3, 4\}^2 \mid x \neq y\}$ (from Example 1) is pp-definable in Γ since

$$K_4(x, y) \Leftrightarrow \{1, 2, 3, 4\}(x) \wedge \{1, 2, 3, 4\}(y) \wedge x \neq y.$$

and it follows that CSP(Γ) is NP-hard.

It is worth mentioning that many of the operations in relational algebra can be viewed as pp-definitions. Let R and S denote binary relations. Then, the converse $R^\smile$ has the pp-definition $R^\smile(x, y) \Leftrightarrow R(y, x)$, the intersection $R \cap S$ has the pp-definition $(R \cap S)(x, y) \Leftrightarrow R(x, y) \wedge S(x, y)$, and the composition $R \circ S$ has the pp-definition $(R \circ S)(x, y) \Leftrightarrow \exists z . R(x, z) \wedge S(z, y)$.

2.3 Automorphisms

Keeping the homomorphism definition of CSPs in mind may be helpful in the rest of this section. Let Γ and Δ denote two relational τ-structures. An injective homomorphism that additionally preserve the complement of each relation is called an *embedding*. Homomorphisms from Γ to Γ are called *endomorphisms* of Γ. An *automorphism* of Γ is a bijective endomorphism whose inverse is also an endomorphism; that is, they are bijective embeddings of Γ into Γ. The set containing all endomorphisms of Γ is denoted $\mathrm{End}(\Gamma)$ while the set of all automorphisms is denoted $\mathrm{Aut}(\Gamma)$.

Example 4 Let $R_+ = \{(x, y, z) \in \mathbb{Z}^3 \mid x + y = z\}$. For arbitrary $a \in \mathbb{Z}$, let $e_a : \mathbb{Z} \to \mathbb{Z}$ be defined as $e_a(n) = a \cdot n$. Let $e : \mathbb{Z} \to \mathbb{Z}$ be an arbitrary endomorphism of $(Z; R_+)$; e is a homomorphism so $(e(x), e(y), e(z)) \in R_+$ whenever $(x, y, z) \in R_+$ and, more generally, $e(\sum_{i=1}^k x_i) = \sum_{i=1}^k e(x_i)$ when $x_1, \ldots, x_k \in \mathbb{Z}$. Arbitrarily choose $n \in \mathbb{Z}$ and note that

$$e(n) = e(\underbrace{1 + \cdots + 1}_{n \text{ times}}) = n \cdot e(1).$$

It follows that $\mathrm{End}((Z; R_+)) = \{e_a \mid a \in \mathbb{Z}\}$. Note that e_a has an inverse if and only if $a \in \{-1, 1\}$. Thus, $\mathrm{Aut}((Z; R_+)) = \{e_a \mid a \in \{-1, 1\}\}$.

A useful observation is that if (V, C) is an instance of CSP(Γ) with a solution $s : V \to D$, then $s' : V \to D$ defined by $s'(x) = \alpha(s(x))$ is a solution to (V, C) for every α in $\mathrm{Aut}(\Gamma)$ or $\mathrm{End}(\Gamma)$. If a function $s : V \to D$ is *not* a solution to (V, C), then $s'(x) = \alpha(s(x))$ is not a solution for any $\alpha \in \mathrm{Aut}(\Gamma)$ while s' may or may not be a solution if $\alpha \in \mathrm{End}(\Gamma) \setminus \mathrm{Aut}(\Gamma)$.

In the following, let G be a set of permutations of a set X. We say that G is a *permutation group* if the identity permutation is in G and for arbitrary $g, f \in G$, the functions $x \mapsto g(f(x))$ and $x \mapsto g^{-1}(x)$ are also in G. In other words, G is closed under function composition and inversion. If Γ is a τ-structure, then $\mathrm{Aut}(\Gamma)$ is a permutation group on the set D_Γ. For $n \geq 1$, the *orbit* of $(t_1, \ldots, t_n) \in X^n$ under G is the set $\{(\alpha(t_1), \ldots, \alpha(t_n)) \mid \alpha \in G\}$. Clearly, the orbits of n-tuples under G partition the set X^n, that is, every $(t_1, \ldots, t_n) \in X^n$ lies in precisely one orbit under G.

Example 5 Consider once again the structure $(\mathbb{Z}, R_+)$ from Example 4. It is obvious that $\{e_1, e_{-1}\}$ forms a (trivial) group under function composition. If $a \in \mathbb{Z}$, then the orbit of (a) equals $\{a, -a\}$ so $(\mathbb{Z}; R_+)$ admits an infinite number of different orbits under its automorphism group.

A (first-order) *theory* is a set of first-order sentences. When the first-order sentences are over the signature τ, we say that T is a *τ-theory*. The *(full) theory* of a τ-structure Δ (denoted $\mathrm{Th}(\Delta)$) is the set of τ-sentences ϕ such that $\Delta \models \phi$. A *model* of a τ-theory T in a τ-structure Δ such that Δ satisfies all sentences in T. Theories that have a model are called *satisfiable*. We now define a central concept: a satisfiable first-order theory T is *ω-categorical* if all countable models of T are isomorphic, and a structure is ω-categorical if its first-order theory is ω-categorical. All ω-categorical structures that appear in this article will be countably infinite, we make the convention that ω-categorical structures are countably infinite. Note that the first-order theory of a finite structure does not have infinite models so finite structures are ω-categorical. One of the first infinite structures that were found to be ω-categorical (by Cantor [15]) is the linear order of the rational numbers $(\mathbb{Q}; <)$. There are many characterisations of ω-categoricity and the most important one is in terms of the automorphism group.

Definition 6 A permutation group G over a countably infinite set X is *oligomorphic* if G has only finitely many orbits of n-tuples for each $n \geq 1$.

An accessible proof of the following theorem can be found in Hodges' book [25].

Theorem 1 (Engeler, Ryll-Nardzewski, Svenonius) *Let Γ be a countably infinite structure Γ with countable signature. The following are equivalent.*

1. *Γ is ω-categorical,*
2. *$\mathrm{Aut}(\Gamma)$ is oligomorphic, and*
3. *a relation is first-order definable in Γ if and only if it is preserved by the automorphisms of Γ.*

Example 5 immediately implies that $(\mathbb{Z}; R_+)$ is not an ω-categorical structure. Consider the structure $(\mathbb{Z}; <)$. One can verify that $\mathrm{Aut}((\mathbb{Z}; <)) = \{x \mapsto x + a \mid a \in \mathbb{Z}\}$. Hence, $(\mathbb{Z}; <)$ is not ω-categorical (despite the fact that $(\mathbb{Q}; <)$ is indeed ω-categorical): the orbits of $(0, 0)$, $(0, 1)$, $(0, 2), \ldots$ are distinct.

We conclude this section by presenting a result that connects first-order definability with ω-categoricity.

Theorem 2 (Thm. 7.3.8 in Hodges [25]) *If Γ is an ω-categorical structure and Δ is first-order definable in Γ, then Δ is ω-categorical, too.*

3 METHOD I: MODEL-COMPLETE CORES

Our first method is based on analysing a given constraint language Γ with respect to its endo- and automorphisms. We first need to introduce the concept of *homomorphically equivalent CSPs*. Let Γ and Δ denote two relational τ-structures. A bijective homomorphism from Γ to Δ is called an *isomorphism*. If Γ and Δ are isomorphic, then it is clear that CSP(Γ) and CSP(Δ) are the same computational problem. However, Γ and Δ may be non-isomorphic and still have the same CSP. This is, for instance, the case when there simultaneously exists a homomorphism from Γ to Δ and a homomorphism from Δ to Γ. In this case, we say that Γ and Δ are *homomorphically equivalent* and this defines an equivalence relation on structures. We note that there are structures that have the same CSP even when they are not homomorphically equivalent. Consider for example the structures $(\mathbb{Z}; <)$ and $(\mathbb{Q}; <)$. They have the same CSP and there is a homomorphism from $(\mathbb{Z}; <)$ to $(\mathbb{Q}; <)$ but there is no homomorphism from $(\mathbb{Q}; <)$ to $(\mathbb{Z}; <)$.

For ω-categorical structures Γ, the equivalence classes have interesting properties: the homomorphic equivalence class of Γ contains a distinguished member Δ which is up to isomorphism uniquely given by two properties: Δ is a *core* and Δ is *model-complete*. A relational structure Γ is a core if all endomorphisms (i.e. homomorphisms from Γ to Γ) are embeddings. Cores are important when studying the complexity of finite-domain CSPs: we refer to the textbook by Hell and Nešetřil [23] that extensively covers cores in the context of graph homomorphisms and to Bulatov et al. [13] that covers cores in general finite-domain CSPs. Model completeness is a central concept in model theory: a structure Γ is model-complete if every formula in $\mathrm{Th}(\Gamma)$ is equivalent to an existential formula modulo T. This may be viewed as a limited notion of quantifier elimination.

Consider the relation $<$ over the rationals $\mathbb{Q}$. The structure $(\mathbb{Q}; <)$ admits quantifier elimination [33] so every formula in $\mathrm{Th}(\{<\})$ is equivalent to a quantifier-free formula (and, naturally, an existential formula). It follows that $(\mathbb{Q}; <)$ is model-complete, and that every Γ that is first-order definable in $(\mathbb{Q}; <)$ is model-complete, too. The structure $(\mathbb{Q}; <)$ is also a core. Let $e : \mathbb{Q} \to \mathbb{Q}$ be an endomorphism of $(\mathbb{Q}; <)$, i.e. if $a < b$, then $e(a) < e(b)$. Clearly, e is injective and it preserves the relation $\geq$ (that is, the negation of $<$) since if $a > b$, then $e(a) > e(b)$ and if $a = b$, then $e(a) = e(b)$. However, there are relations R that are first-order definable in $(\mathbb{Q}; <)$ and $(\mathbb{Q}; R)$ is not a core. One trivial example is the equality relation =. The function $x \mapsto 1$ is obviously an endomorphism of = but it is not injective and thus not an embedding. We have the following important result.

Theorem 3 (Bodirsky [5]) *Every ω-categorical structure Δ is homomorphically equivalent to a model-complete core structure Γ which is unique up to isomorphism. Moreover, Γ is ω-categorical and the orbits of n-tuples are pp-definable in Γ for all $n \geq 1$.*

Since homomorphically equivalent structures have the same CSP, one can focus on ω-categorical structures that have these properties. The fact that we can pp-define the orbits of n-tuples will now become highly important.

Theorem 4 *Let Γ be a constraint language over the domain D. Assume the following:*

1. *Γ is a model-complete ω-categorical core and*
2. *the domain elements are represented in a way such that given a vector $\bar{d} = (d_1, \ldots, d_n) \in D^n$, a pp-definition in Γ of the orbit of $\bar{d}$ can be generated in polynomial time (in the size of the representation of $d_1, \ldots, d_n$).*

Then, CSP(Γ) and CSP($\Gamma \cup D_c$) are polynomial-time equivalent.

Proof. Let $\Gamma' = \Gamma \cup D_c$. The reduction from CSP(Γ) to CSP(Γ') is trivial so we concentrate on the other direction. Let $I' = (V', C')$ be an instance of CSP(Γ'). Assume without loss of generality that if $\{d_i\}(x)$ is in C', then there is no variable $y \neq x$ such that $\{d_i\}(y) \in C'$; if so, the constraint $\{d_i\}(y)$ can be removed and the variable y be replaced by x. Normalising an instance in this way can easily be done in polynomial-time. We assume (without loss of generality) that the only constraints in C' with relations from D_c are $\{d_1\}(x_1), \ldots, \{d_m\}(x_m)$. This can be achieved in polynomial time by renaming of variables.

Compute (in polynomial time) the formula $F(x_1, \ldots, x_m)$ for the orbit of $(d_1, \ldots, d_m)$. Define $I = (V, C)$ such that C equals C' extended with $F(x_1, \ldots, x_m)$ and the constant relations removed. Let V denote V' expanded with the existentially quantified variables in $F(x_1, \ldots, x_m)$. Note that I can be constructed in polynomial time and it is an instance of CSP(Γ).

If the instance I has no solution, then it follows immediately that I' does not have a solution—one can view I as being a relaxation of I' since the formula $F(x_1, \ldots, x_m)$ is, in particular, satisfiable when $x_1 = d_1, \ldots, x_m = d_m$. If the instance I has a solution $s : V \to D$, then we claim that there is a solution $s' : V' \to D$ to I', too. Since F describes the orbit of $(d_1, \ldots, d_m)$, there is an automorphism α of Γ such that $\alpha(s'(x_i)) = d_i$, $1 \leq i \leq m$. This implies that that $\alpha(s'(x))$ restricted to the set V is a solution to I. □

By Theorem 3, we know that orbit-defining formulas always can be pp-defined in Γ under the given assumptions. Whether these can be generated or not in polynomial time is a completely different question, though. We give an example based on constraint languages that are first-order definable in $(\mathbb{Q}; <)$. Such constraint languages are sometimes called *temporal constraint languages*. They are well-studied in the literature and, in fact, the computational complexity of CSP(Γ) is known for every finite Γ [4]. A concrete example of a temporal constraint language is the point algebra PA: we see that $x \leq y \Leftrightarrow (x < y) \vee (x = y)$ and $x \neq y \Leftrightarrow \neg(x = y)$. Furthermore, temporal constraint languages are ω-categorical due to Theorem 2 and it is known (by Junker and Ziegler [31], also see Cameron [14]) that there are five possible choices of $\mathrm{Aut}(\Gamma)$. We concentrate on the (for our purposes) most interesting case when $< \in \langle\Gamma\rangle$ and $\mathrm{Aut}(\Gamma) = \mathrm{Aut}(\mathbb{Q}; <)$. Arbitrarily choose such a language Γ and assume (without loss of generality) that $< \in \Gamma$. We know that Γ is model-complete so we assume that Γ is a core (for instance, Γ may be the point algebra PA). We represent all members of $\mathbb{Q}$ in the natural way, i.e. as (a/b) where $a, b \neq 0$ are integers written in binary.

The automorphisms of $(\mathbb{Q}; <)$ are the bijective functions $f : \mathbb{Q} \to \mathbb{Q}$ that are monotonously increasing. The orbit of 1-tuples equals $\mathbb{Q}$ while the orbit of a 2-tuple (a, b) with $a < b$ equals $\{(x, y) \in \mathbb{Q}^2 \mid x < y\}$. More generally, the orbit of a k-tuple $(a_1, \ldots, a_k)$ with $a_1 < a_2 < \cdots < a_k$ equals

$$\{(x_1, \ldots, x_k) \in \mathbb{Q}^k \mid x_1 < x_2 < \cdots < x_k\}$$

so the orbit-defining formulas can be generated in polynomial time. Theorem 4 is thus applicable and CSP($\Gamma \cup \mathbb{Q}_c$) is polynomial-time equivalent to CSP(Γ). In particular, CSP($\Gamma \cup \mathbb{Q}_c$) is in P if CSP(Γ) is in P, and CSP($\Gamma \cup \mathbb{Q}_c$) is in NP if CSP(Γ) is in NP.

This example shows that ω-categoricity is indispensable. Theorem 4 combined with the tractability of CSP($(\mathbb{Q}; <, \neq)$) implies that CSP($\Gamma_{\mathbb{Q}}$) is in P when $\Gamma_{\mathbb{Q}}$ denotes $(\mathbb{Q}; <, \neq)$ extended with the unary relations in $\mathbb{Q}_c$. Recall that $(\mathbb{Z}; <)$ and $(\mathbb{Z}; <, \neq)$ are not ω-categorical and define $\Gamma_{\mathbb{Z}}$ by expanding $(\mathbb{Z}; <, \neq)$ with $\mathbb{Z}_c$. The problem CSP($\Gamma_{\mathbb{Z}}$) is NP-hard since the relation $\{0, 1, 2\}$ can be pp-defined via $x \in \{0, 1, 2\} \Leftrightarrow -1 < x \wedge x < 3$, and the problem CSP($(\mathbb{Z}; \{0, 1, 2\}, \neq)$) is NP-hard since there is an obvious polynomial-time reduction from 3-COLOURABILITY.

4 METHOD II: HOMOGENEITY

Homogeneous structures have been intensively studied in mathematics and logics (for instance, in connection with combinatorics, model theory, and group theory) and they are becoming more and more relevant in the study of CSPs. Homogeneous structures have useful properties such as admitting quantifier elimination and they are ω-categorical whenever the structure contains a finite number of relations and the domain is countably infinite. Examples include $(\mathbb{Q}; <)$, the *random* (or Rado) graph, and certain structures with connections to phylogenetic reconstruction problems. There are also many structures that are well-studied in AI that can be represented by homogeneous structures: examples include Allen's algebra [24] and RCC-8 [9]. We need some machinery before providing the formal definition. Let D be the domain of a relational τ-structure Γ and arbitrarily choose $S \subseteq D$. Then the *substructure induced by* S *in* Γ is the τ-structure Δ with domain S such that $R^{\Delta} = R^{\Gamma} \cap S^n$ for each n-ary $R \in \tau$; we also write $\Gamma[S]$ for Δ. The structure Γ is called *homogeneous* if every isomorphism $f : D_1 \to D_2$ between finite induced substructures of Γ can be extended to an automorphism of Γ, i.e. there exists an automorphism α such that $f(x) = \alpha(x)$ when $x \in D_1$. One should note that homogeneity is a more "fragile" concept than ω-categoricity. For instance, Γ being homogeneous and Δ being first-order definable in Γ does not necessarily imply that Δ is homogeneous.

To simplify the presentation, we will turn our attention to binary constraints and *partition schemes*; this concept was introduced by Ligozat & Renz [37] and it has been highly influential in CSP research. Let D be a non-empty domain. Given a finite family $\mathcal{B} = \{R_1, \ldots, R_k\}$ of binary relations over D, we say that $\mathcal{B}$ is *jointly exhaustive* (JE) if $\bigcup \mathcal{B} = D^2$ and that $\mathcal{B}$ is *pairwise disjoint* (PD) if $R_i \cap R_j = \varnothing$ whenever $1 \leq i \neq j \leq k$. If $\mathcal{B}$ is simultaneously JE and PD, then $\mathcal{B}$ forms a partition of the set D^2.

Definition 7 Let D be a non-empty domain and let $\mathcal{B} = \{R_1, \ldots, R_k\}$ be a set of binary relations over D. We say that $\mathcal{B}$ is a *partition scheme* if the following holds:

1. $\mathcal{B}$ is JEPD,
2. the equality relation $\mathtt{EQ}_D = \{(x, x) \in D^2\}$ is in $\mathcal{B}$, and
3. for every $R_i \in \mathcal{B}$, the converse relation $R_i^{\smile}$ is in $\mathcal{B}$.

It is important to note that if $\mathcal{B}$ is a partition scheme over a domain D, then for arbitrary $d, d' \in D$ there exists exactly one $B \in \mathcal{B}$ such that $(d, d') \in B$. Given a finite set of binary relations $\mathcal{B} = \{R_1, \ldots, R_k\}$, we follow notational conventions from [16, 30] and define $\mathcal{B}^{\vee=}$ as the set of all unions of relations from $\mathcal{B}$. The set $\mathcal{B}^{\vee=}$ and the problem CSP(Γ) where $\Gamma \subseteq \mathcal{B}^{\vee=}$ are the natural objects that are studied in connection with partition schemes.

Theorem 5 *Let $\mathcal{B} = \{B_1, \ldots, B_k\}$ be a partition scheme over the domain D. Assume the following:*

1. *$\mathcal{B}^{\vee=}$ is homogeneous, and*
2. *the domain elements are represented in a way such that given two elements $a, b \in D$, it is possible to find (by using an algorithm A) the unique B_i, $1 \leq i \leq m$, such that $(a, b) \in B_i$ in polynomial time (measured in the size of the representations of a and b).*

If $\mathcal{B} \subseteq \Gamma \subseteq \mathcal{B}^{\vee=}$, then CSP($\Gamma$) and CSP($\Gamma \cup D_c$) are polynomial-time equivalent.

Proof. Let $\Gamma' = \Gamma \cup D_c$. The reduction from CSP(Γ) to CSP(Γ') is trivial so we concentrate on the other direction. Let $I' = (V', C')$ be an instance of CSP(Γ'). We assume without loss of generality (just as in the proof of Theorem 4) that the only constraints in C' with relations from D_c are $\{d_1\}(x_1), \ldots, \{d_m\}(x_m)$.

Construct an instance $I = (V, C)$ of CSP($\mathcal{B}^{\vee=}$) as follows: let

- $V = V'$,
- $\widehat{C} = \{A(d_i, d_j)(x_i, x_j) \mid 1 \leq i \neq j \leq m\}$, and
- $C = (C' \cup \widehat{C}) \setminus \{\{d_1\}(x_1), \ldots, \{d_m\}(x_m)\}$.

The instance $I = (V, C)$ can obviously be generated in polynomial time.

If the instance I' has a solution, then it follows immediately that I has a solution—the constraints in $\widehat{C}$ are satisfiable by the assignment $x_1 = d_1, \ldots, x_m = d_m$.

If the instance I has a solution $s : V \to D$, then we claim that there is a solution $s' : V \to D$ to I', too. Let $S = \{s(x_1), \ldots, s(x_m)\}$ and $T = \{d_1, \ldots, d_m\}$. The set T contains m elements by our initial assumptions and the set S contains m elements due to the constraints in $\widehat{C}$; all variables in $\{x_1, \ldots, x_m\}$ are assigned distinct values since none of the constraints in $\widehat{C}$ allows equality (due to the fact that $\mathcal{B}$ is a partition scheme and $d_1, \ldots, d_m$ are distinct values). Thus, $f : S \to T$ is a well-defined bijective function if we let $f(s(x_i)) = d_i$, $1 \leq i \leq m$. We continue by proving the following claim.

Claim: f is an homomorphism from $B[S]$ to $B[T]$ when $B \in \mathcal{B}$. Arbitrarily choose a tuple $(a, b) \in B[S]$. By the choice of S, we know that $a = s(x_i)$ and $b = s(x_j)$ for some distinct $1 \leq i, j \leq m$. We see that

$$(f(a), f(b)) = (f(s(x_i)), f(s(x_j))) = (d_i, d_j).$$

We know that $d_i, d_j \in T$ so it remains to show that $(d_i, d_j) \in B$. If $A(d_i, d_j) = B$, then we are done. If $A(d_i, d_j) = B' \neq B$, then

$B'(x_i, x_j) \in \widehat{C} \subseteq C$ so $(s(x_i), s(x_j)) \in B'$. This contradicts that $a = s(x_i)$, $b = s(x_j)$, and $(a, b) \in B$ since $B \cap B' = \varnothing$.

We show that f is a homomorphism from $\mathcal{B}^{\vee=}[S]$ to $\mathcal{B}^{\vee=}[T]$. Since f is bijective, it follows that f is an isomorphism between $\mathcal{B}^{\vee=}[S]$ and $\mathcal{B}^{\vee=}[T]$, too. Arbitrarily choose a relation $R \in \mathcal{B}^{\vee=}$ where $R = B_1 \cup \cdots \cup B_p$ and $B_i \in \mathcal{B}$, $1 \leq i \leq p$. Arbitrarily choose $(a, b) \in R[S]$. The tuple (a, b) is a member of some relation B_i in $\{B_1, \ldots, B_p\}$. By the Claim, $(f(a), f(b)) \in B_i[T]$ so $(f(a), f(b)) \in R[T]$ since $B_i \subseteq R$. It follows that f is a homomorphism from $R[S]$ to $R[T]$ since (a, b) was arbitrarily chosen in $R[S]$. This, in turn, implies that f is a homomorphism $\mathcal{B}^{\vee=}[S]$ to $\mathcal{B}^{\vee=}[T]$ since R was arbitrarily chosen in $\mathcal{B}^{\vee=}$.

Since $\mathcal{B}^{\vee=}$ is a homogeneous structure, the function f can be extended to an automorphism α of $\mathcal{B}^{\vee=}$. It follows that the function $s' : V \to D$ defined such that $s'(x) = \alpha(s(x))$ is a solution to I'; merely note that $s'(x_i) = d_i$, $1 \leq i \leq m$. □

Consider Allen's algebra $\mathcal{A}$ with domain $\mathbb{I}$ where intervals are represented as (I^-, I^+) where $I^- < I^+$, $I^-, I^+ \in \mathbb{Q}$, and the members of $\mathbb{Q}$ are represented as in Section 3. Hirsch [24] has shown that $\mathcal{A}$ is a homogeneous structure and the second precondition of Theorem 5 is clearly satisfied with the given representation. We conclude that CSP($\mathcal{A} \cup \mathbb{I}_c$) and CSP($\mathcal{A} \cup \mathbb{I}_f$) are NP-complete problems since CSP($\mathcal{A}$) is NP-complete. We can also conclude that CSP($\mathcal{H} \cup \mathbb{I}_c$) is in P when $\mathcal{H}$ is the ORD-Horn subclass [40] since $\mathcal{H}$ contains all 13 basic relations. More examples of homogeneous structures that are relevant for computer science are described in, for example, Bodirsky [5], Bodirsky and Chen [6], and Bodirsky and Wölfl [9].

5 METHOD III: SMALL SOLUTIONS

The methods in Section 3 and 4 provide polynomial-time equivalences between CSP(Γ) and CSP($\Gamma \cup D_c$) under certain conditions. In this section, we will instead analyse the constraint language $\Gamma \cup D_c$ directly. The main result will be weaker than in the previous two sections since we will only be able to prove membership in NP. On the other hand, the approach is applicable also without ω-categoricity.

Let Γ be an arbitrary constraint language with domain D, and assume that the relations in Γ and the elements in D are represented is some fixed way. We say that Γ has the *small solution property* if there exists a polynomial p (that only depends on the choice of Γ) such that for every satisfiable instance $I = (V, C)$ of CSP(Γ), there exists a solution $s : V \to D$ such that $||s(v)|| \leq p(||I||)$ for every $v \in V$.

Lemma 8 *Let Γ denote a constraint language over the domain D. Assume that*

1. *Γ has the small solution property and*
2. *there exists an algorithm A and a polynomial q such that for arbitrary k-ary $R \in \Gamma$ and $d_1, \ldots, d_k \in D$, algorithm A can verify whether $(d_1, \ldots, d_k) \in R$ or not in time $O(q(||R|| + \sum_{i=1}^{k} ||d_i||))$.*

Then CSP(Γ) is in NP.

Proof. Let (V, C) denote an arbitrary instance of CSP(Γ). To show that $I = (V, C)$ is satisfiable, non-determinstically guess a solution $s : V \to D$ such that $||s(v)|| \leq p(||I||)$ for every $v \in V$ (where p denotes a fixed polynomial). Such a solution exists since Γ has the small solution property, and the size of s is consequently polynomially bounded in $||I||$. The solution s can thus be verified in polynomial time with the aid of algorithm A. □

The small solution property is particularly useful in connection with partition schemes.

Lemma 9 *Let $\mathcal{B}$ be a partition scheme with domain D such that precondition (2) of Lemma 8 is satisfied. If $\mathcal{B} \cup D_c$ has the small solution property, then $\mathcal{B}^{\vee=} \cup D_c$ has the small solution property and both CSP($\mathcal{B}^{\vee=} \cup D_c$) and CSP($\mathcal{B}^{\vee=} \cup D_f$) are in NP.*

Proof. Let $I = (V, C)$ denote an instance of CSP($\mathcal{B}^{\vee=} \cup D_c$) with solution $s : V \to D$. Replace each binary constraint $x(b_1 \cup \cdots \cup b_m)y \in C$ (where $\{b_1, \ldots, b_m\} \subseteq \mathcal{B}$) with the constraint $x\{b_i\}y$, $1 \leq i \leq m$, and $(s(x), s(y)) \in b_i$. The resulting instance $I' = (V, C')$ is solvable, $||I'|| \leq ||I||$, and it is an instance of CSP($\mathcal{B} \cup D_c$). We know that $\mathcal{B} \cup D_c$ has the small solution property so there is a solution $s' : V \to D$ such that $||s'(v)|| \leq p(||I'||)$ for every $v \in V$ and some polynomial p that only depends on $\mathcal{B}$. Since s' is a solution to I, too, it follows that $||s'(v)|| \leq p(||I'||) \leq p(||I||)$. Thus, $\mathcal{B}^{\vee=} \cup D_c$ has the small solution property. Lemma 8 implies that CSP($\mathcal{B}^{\vee=} \cup D_c$) is in NP and consequently Lemma 3 implies that CSP($\mathcal{B}^{\vee=} \cup D_f$) is in NP. □

Many well-known structures possess the small solution property. A prime example is relations R defined by linear expressions, that is, R is defined by

$$(x_1, \ldots, x_k) \in R \Leftrightarrow \sum_{i=1}^{k} c_i \cdot x_i \leq c_0$$

or

$$(x_1, \ldots, x_k) \in R \Leftrightarrow \sum_{i=1}^{k} c_i \cdot x_i = c_0$$

where the coeffecients are in $\mathbb{Z}$ and the variables ranges over, for instance, $\mathbb{Q}$ or $\mathbb{Z}$. Given a constraint language Γ containing such relations, the small solution property for $\mathbb{Q}$ follows from the fact that linear programming can be solved (and a concrete solution written down) in polynomial time while the property for $\mathbb{Z}$ has been proven by Papadimitriou [41]. This example is interesting in several respects. First of all, the constants in $\mathbb{Q}_c$ are, of course, linear. Furthermore, we know (from Example 4) that not even the language $\Gamma = \{\{(x, y, z) \in \mathbb{Z}^3 \mid x + y = z\}$ is ω-categorical; the same can be proved for the domain $\mathbb{Q}$. Thus, the methods in Section 3 and 4 are not applicable in this case.

We illustrate the small solution property with a different example: the RCC-5 formalism. The RCC formalisms [42] are designed for reasoning about spatial regions and they are the basis for a large part of the work in *qualitative spatial reasoning* (QSR). There are several variants such as RCC-23, RCC-8, and RCC-5. We concentrate on the simplest formalism RCC-5. The interpretation of the five basic relations in RCC-5 is given in Figure 1 and it is easy to see that they constitute a partition scheme. The choice of objects is important in RCC-5 and different choices may give rise to different computational problems. A (slightly degenerated) example is if the set of regions only contains one member. In this case, all basic relations except `EQ` are empty and this makes the CSP problem for the power set of the partition scheme tractable. If we instead assume that the regions are non-empty regular subsets of an infinite topological space, then the very same problem is NP-hard [43]. In the sequel, we consider the variant of RCC-5 where the objects are non-empty subsets of an infinite set, e.g., of $\mathbb{N}$. We denote this variant by RCC-5_{Set} and we let $\mathcal{R}$ be the corresponding set of basic relations. This particular interpretation is interesting since it can be viewed as the least restricted variant of RCC-5: if there is a solution to an RCC-5 instance when the regions are taken from some set X that does not contain the empty set, then there is a solution where the regions are taken from $2^{\mathbb{N}} \setminus \varnothing$. This is well-known and it can quite easily be proved by methods similar to those used in Drakengren and Jonsson [20] (which, incidentally, also have inspired the proof of Proposition 10 below). Further discussions concerning different interpretations of RCC-5 and other spatial formalisms can be found in [6, 21, 36].

We first establish that the methods in Sections 3 and 4 are not applicable. One could do this by analysing the automorphism group and conclude that RCC-5_{Set} is not ω-categorical. A simpler way is the following (but it is tacitly based on the assumption P $\neq$ NP). Let $\Gamma = \mathcal{R} \cup \{\neq\}$ where $\neq$ equals $\bigcup \mathcal{R} \setminus \{\mathtt{EQ}\}$. It is known that CSP($\Gamma$) is in P [29, 43]. We extend Γ with one constant: $\Gamma' = \Gamma \cup \{\{0,1,2\}\}$. Consider the constraints $\{y\{\mathtt{PP}\}z, \{0,1,2\}(z)\}$. It is clear that if s is a solution, then $s(y) \in 2^{\{0,1,2\}} \setminus \{\varnothing, \{0,1,2\}\}$, i.e. there are 6 distinct possible choices for the variable y. This implies that there is a straightforward polynomial-time reduction from 6-COLOURABILITY to CSP(Γ') (since the relation $\neq$ is in Γ') and, consequently, that CSP(Γ') is NP-complete. If Theorem 4 or Theorem 5 were applicable, then CSP(Γ') would be polynomial-time solvable.

Proposition 10 *Let $D = 2^{\mathbb{N}} \setminus \varnothing$. The constraint language $\mathcal{R}^{\vee=} \cup D_c$ has the small solution property and CSP($\mathcal{R}^{\vee=} \cup D_c$) is in NP.*

Proof. By Lemma 9, it is sufficent to show that $\mathcal{R} \cup D_c$ has the small solution property. Let $I = (V, C)$ be a satisfiable instance of CSP($\mathcal{R} \cup D_c$) with solution $s : V \to 2^{\mathbb{N}} \setminus \{\varnothing\}$. Construct a new instance $I' = (V', C')$ as follows.

Step 1. Remove every $x\{\mathtt{EQ}\}y$ constraint: this can be done by collapsing the variables x and y (we leave the obvious details of this step to the reader).

Step 2. Replace every $x\{\mathtt{PP}^{\smile}\}y$ constraint with $y\{\mathtt{PP}\}x$.

Step 3. Remove every $x\{\mathtt{PO}\}y$ constraint by replacing it with

$$\begin{array}{lll} z_1\{\mathtt{DR}\}z_2 & z_2\{\mathtt{DR}\}z_3 & z_3\{\mathtt{DR}\}z_1 \\ z_1\{\mathtt{PP}\}x & z_1\{\mathtt{DR}\}y & \\ z_2\{\mathtt{PP}\}x & z_2\{\mathtt{PP}\}y & \\ z_3\{\mathtt{PP}\}y & z_3\{\mathtt{DR}\}x & \end{array}$$

where z_1, z_2, z_3 are fresh variables.

Note that I' is still a satisfiable instance of CSP($\mathcal{R} \cup D_c$) and that the only non-unary relations that appear in I are DR and PP. Additionally note that if there is a solution to I with codomain S, then there is a solution to I' with codomain S.

We say that two variables u, v in I' are PP-*connected* if there exists a sequence of variables $w_1, \ldots, w_p$ such that

1. $w_1 = u$,
2. $w_p = v$, and
3. $w_i\{\mathtt{PP}\}w_{i+1} \in C'$ for all $1 \leq i < p$.

Note that if u and v are PP-connected, then in any solution s' of I' we have that $(s'(u), s'(v)) \in \mathtt{PP}$.

Let T denote the number of elements in the largest unary relation appearing in I. If u is PP-connected with some variable v and $U(v) \in C$, then we know that $|s'(u)| < T$ for any solution s' to I'. We prove that at most $|V| \cdot T$ different elements are needed for representing a solution by induction over the number of variables in V'. This implies the result by reasoning as follows: we can without loss of generality assume that the set of possible values is $2^{\{1,\ldots,|V|\cdot T\}} \setminus \{\varnothing\}$. To represent such a value, we need at most $|V| \cdot T$ bits if we view each value as a 0/1-vector where the i:th component equals 1 if and only i is a member of the value. Hence, CSP($\mathcal{R} \cup D_c$) has the small solution property since $|V| \leq ||I'|| \leq ||I||$ and $T \leq ||I'|| \leq ||I||$.

Basis step. If $|V'| = 1$ and $V = \{v\}$, then either one value is sufficient (if v is not constrained by a unary relation) or T values are sufficient (otherwise).

Induction hypothesis. Assume the claim holds when $|V'| = p$.

Induction step. We show the claim when $|V'| = p+1$. Choose a variable $v \in V$ such that v is maximal with respect to PP-connectedness, i.e. v is not PP-connected to any other variable. By the induction hypothesis, we need at most pT values for the instance $I' \setminus \{v\}$. If there exists $U(v) \in C$, then we need at most T values for v which gives us at most $pT + T = (p+1)T$ values in total. If there is no $U(v) \in C$, then we need at most one additional value for v so we need at most $pT + 1 \leq (p+1)T$ values in total. To see this, v may (in the worst case) be PP-connected to every other variable and v must (by the induction hypothesis) contain at least pT different values. However, it must also be a strict superset of the other variables and this is accomplished by adding one fresh element. □

The basic proof idea of Proposition 10 is to analyse the growth of the variables that are not constrained by any unary relation. Clearly, if a variable v is constrained by a constant relation $\{c\}$, then any solution must satisfy $s(v) = c$ and $||c|| \leq ||I||$. Otherwise, PP-connectedness gives a way of estimating the size of the contents of the other variables. This idea can readily be extended to other classes of relations that are related to RCC-5 such as (certain variants of) *set relations* (cf. Bodirsky and Hils [7] and the references in their paper), and it can also be generalised in other directions. An interesting observation is that the NP membership results for RCC5 and RCC8 with polygonal regions in the plane by Li et al. [35] is implicitly based on the small solution property. Here, the representational size of the regions are analysed and bounded by exploiting a particular parameter that is related to embeddings of planar graphs in the plane. Another interesting observation is that Li [34] uses concepts that are similar to PP-connectedness when constructing different realisations of the RCC8 formalism. This may indicate that the approach taken in the proof of Proposition 10 may quite easily be adapted to other spatial formalisms.

We conclude this section by a few observations concerning the small solution property. First of all, it is important to realise that the converse of Lemma 8 does not necessarily hold. To see this, define the rapidly increasing function

$$\text{Tower}(n) = \underbrace{2^{2^{\cdot^{\cdot^{2}}}}}_{n \text{ times}}.$$

Clearly, $\log(\text{Tower}(n))$ grows faster than any polynomial in n. Now consider the constraint language $\Gamma = \{U_1, U_2, \ldots\}$ where $U_i = \{x \in \mathbb{N} \mid x = \text{Tower}(i)\}$. Checking if there an instance of CSP(Γ) is satisfiable or not can trivially be solved in polynomial time if $U_1, U_2, \ldots$ are represented in a reasonable way—for instance, if U_i is represented by the number i written in binary. Thus, CSP(Γ) is in NP, too. It is obvious, though, that Γ does not have the small solution property if we represent the natural numbers in binary.

Another important observation is that it is *not* sufficient to verify that Γ itself has the small solution property—one need to verify that $\Gamma \cup D_c$ has the small solution property. We exemplify by using the

$$
\begin{array}{lll}
X\{\texttt{PP}\}Y & \Leftrightarrow & X \subset Y \\
X\{\texttt{PP}^{\smile}\}Y & \Leftrightarrow & X \supset Y \\
X\{\texttt{DR}\}Y & \Leftrightarrow & X \cap Y = \varnothing \\
X\{\texttt{PO}\}Y & \Leftrightarrow & \exists a,b,c : a \in X, a \notin Y, b \in X, b \in Y, c \notin X, c \in Y \\
X\{\texttt{EQ}\}Y & \Leftrightarrow & X = Y
\end{array}
$$

Figure 1. The five basic relations of RCC-5.

relation $R = \{(x, y) \in \mathbb{N}^2 \mid x = 2^{y-1}\}$. The constraint language $\{R\}$ has the small solution property since every instance has the solution that assigns 1 to every variable. However, CSP($\{R, \{2\}\}$) does not have small solutions. Consider the instance (V, C) where $V = \{x_0, \ldots, x_n\}$ and

$$C = \{\{2\}(x_0), R(x_1, x_0), R(x_2, x_1), \ldots, R(x_n, x_{n-1})\}.$$

It is easy to verify that (V, C) is solvable and every solution $s : V \to \mathbb{N}$ must satisfy $s(x_n) = \text{Tower}(n)$.

Finally, we want to emphasise once again that the choice of exact interpretation and representation of relations and domain elements is extremely important. Bodirsky and Chen [6] have presented an interpretation of RCC-5 that is homogeneous. In this case, adding constants preserves computational complexity (up to polynomial-time reductions) by Theorem 5 (given that relations and domain elements are represented in a suitable way). We know from earlier examples that this does not hold for RCC-5_{Set}.

6 DISCUSSION

We have presented three different methods for analysing the complexity of qualitative CSPs extended with finite unary relations, and identifying additional general methods for studying the complexity of such CSPs is an obvious research direction. One should observe that restricting oneself to *finite* unary relations may be reasonable in certain cases but not in others. For instance, a substantial part of the literature on temporal reasoning is concerned with TCSPs and the simple temporal problem (STP) [19]: the basic binary relations here are expressions $a \leq x - y \leq b$ (where $a, b \in \mathbb{Q}$ and x, y are variables) and unary relations $a \leq x \leq b$ (which are either constants or infinite unary relations depending on the choice of a and b). It is easy to see that extending a formalism with (non-trivial) infinite unary relations may yield an easier computational problem than adding finite unary relations. For instance, PA extended with the finite unary relation $\{0, 1, 2\}$ is NP-hard since the disequality relation $\neq$ is in PA, while PA extended with the infinite unary relation $\{x \in \mathbb{Q} \mid 0 \leq x \leq 2\}$ is tractable [28]. Thus, it would be interesting to study the computational complexity of CSPs extended with non-finite unary relations.

Our methods I and II are based on certain model-theoretical properties of the underlying constraint languages. While methods based on model theory and universal algebra have been very common when studying CSPs from the viewpoint of theoretical computer science [2, 4, 12], such methods have been less popular within the AI community (with some notable exceptions such as Huang [26]). Thus, we take the opportunity to discuss these methods in slightly more detail.

Method I. (model-complete cores) The main obstacle for applying method I is the need for computing orbit-defining formulas efficiently. In fact, it is not even known if this problem is decidable or not in the general case. Studying this problem is a very important future research direction. In cases where we do not know how to effeciently generate orbit-defining formulas, there are (at least) two possible workarounds. The first one is proposed by Bodirsky [5, Sec. 7]: if the set of possible constants is finite, then an orbit-defining formula for these constants can be computed off-line and subsequently be used without additional cost. Another workaround is to sacrifice polynomial-time equivalence and allow more time for computing the orbit-defining formula. If the problem at hand is NP-hard, then a (preferably mildly) exponential algorithm can be acceptable. In both cases, algorithmic methods for generating orbit-defining formulas would be helpful. We note, on the positive side, that related definability problems have recently been successfully addressed, cf. Bodirsky et al. [8]. Their methods are interesting since they combine methods taken from universal algebra, Ramsey theory, and topological dynamics.

Method II. (homogeneity) We have chosen to present the results when the constraint language is restricted to partition schemes. This is convenient but not inherently necessary—generalisations to (for instance) higher-arity relations are possible. One should consequently not view our results as the only possible way of exploiting homogeneity: how to exploit homogeneity must be decided on a case-by-case basis.

Given a structure Γ, it may be difficult to verify that it is indeed homogeneous. Here, one should note that if Γ contains a finite number of relations, the domain of Γ is countably infinite, and Γ is homogeneous, then Γ is ω-categorical. This is a consequence of Theorem 1 and the details are to be found in Macpherson [39]. A first step is thus to verify the ω-categoricity of Γ, and this can quite often be accomplished by using Theorem 1. If Γ is ω-categorical, then Γ is homogeneous *if and only if* every formula in $\text{Th}(\Gamma)$ is equivalent to a quantifier-free formula (see, for instance, Macpherson [39]). This gives an alternative way of proving homogeneity than using the automorphism-based definition directly. This also clarifies the connections between method I and method II: recall that Γ is model-complete if and only if every formula in $\text{Th}(\Gamma)$ is equivalent to an existential formula.

Another approach for using homogeneity is to construct suitable homogeneous structures "from scratch". The main tool for this is *Fraïssé amalgamation*. The details are outside the scope of this article: Macpherson [39] outlines the approach and concrete constructions for RCC5 and RCC8 can be found in Bodirsky and Chen [6] and Bodirsky and Wölfl [9], respectively. One should note that amalgamation is quite common in the literature on CSPs and related problems; however, it is often referred to as the *patchwork property* [26, 38, 44]

ACKNOWLEDGEMENTS

We thank Manuel Bodirsky, Fredrik Heintz, and Johan Thapper for providing valuable input to this paper.

REFERENCES

[1] Libor Barto. The dichotomy for conservative constraint satisfaction problems revisited. In *Proc. 26th Annual IEEE Symposium on Logic in Computer Science (LICS-2011)*, pages 301–310, 2011.

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

[3] Manuel Bodirsky and M. Grohe. Non-dichotomies in constraint satisfaction complexity. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP-2008)*, pages 184–196, 2008.

[4] Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *J. ACM*, 57(2), 2010.

[5] Manuel Bodirsky. Cores of countably categorical structures. *Logical Methods in Computer Science*, 3(1):1–16, 2007.

[6] Manuel Bodirsky and Hubie Chen. Qualitative temporal and spatial reasoning revisited. *J. Log. Comput.*, 19(6):1359–1383, 2009.

[7] Manuel Bodirsky and Martin Hils. Tractable set constraints. *J. Artif. Intell. Res. (JAIR)*, 45:731–759, 2012.

[8] Manuel Bodirsky, Michael Pinsker, and Todor Tsankov. Decidability of definability. *J. Symb. Log.*, 78(4):1036–1054, 2013.

[9] Manuel Bodirsky and Stefan Wölfl. RCC8 is polynomial on networks of bounded treewidth. In *Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*, pages 756–761, 2011.

[10] Andrei Bulatov. Tractable conservative constraint satisfaction problems. In *Proceedings of the 18th annual IEEE symposium on logic in computer science*, pages 321–???, 2003.

[11] Andrei Bulatov. Conservative constraint satisfaction re-revisited. *Journal of Computer and System Sciences*, 82(2):347–356, 2016.

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

[13] Andrei A. Bulatov, Andrei A. Krokhin, and Peter G. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.

[14] Peter J. Cameron. *Oligomorphic permutation groups*. Cambridge University Press, Cambridge, 1990.

[15] Georg Cantor. Über unendliche, lineare Punktmannigfaltigkeiten. *Mathematische Annalen*, 23:453–488, 1884.

[16] David Cohen, Peter Jeavons, Peter Jonsson, and Manolis Koubarakis. Building tractable disjunctive constraints. *Journal of the ACM*, 47(5):826–853, 2000.

[17] Anthony G. Cohn and Jochen Renz. Qualitative spatial representation and reasoning. In *Handbook of Knowledge Representation*, pages 551–596. Elsevier, 2008.

[18] Daniel de Leng and Fredrik Heintz. Qualitative spatio-temporal stream reasoning with unobservable intertemporal spatial relations using landmarks. In *Proc. 30th AAAI Conference on Artificial Intelligence (AAAI-2016)*, pages ...–..., 2016.

[19] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artif. Intell.*, 49:61–95, 1991.

[20] Thomas Drakengren and Peter Jonsson. Reasoning about set constraints applied to tractable inference in intuitionistic logic. *Journal of Logic and Computation*, 8(6):855–875, 1998.

[21] Ivo Düntsch, Hui Wang, and Stephen McCloskey. A relation-algebraic approach to the region connection calculus. *Theor. Comput. Sci.*, 255(1-2):63–83, 2001.

[22] Stella Giannakopoulou, Charalampos Nikolaou, and Manolis Koubarakis. A reasoner for the RCC-5 and RCC-8 calculi extended with constants. In *Proc. 28th AAAI Conference on Artificial Intelligence (AAAI-2014)*, pages 2659–2665, 2014.

[23] Pavol Hell and Jaroslav Nešetřil. *Graphs and Homomorphisms*. Oxford University Press, Oxford, 2004.

[24] Robin Hirsch. Expressive power and complexity in algebraic logic. *Journal of Logic and Computation*, 7(3):309–351, 1997.

[25] Wilfrid Hodges. *Model theory*. Cambridge University Press, 1993.

[26] Jinbo Huang. Compactness and its implications for qualitative spatial and temporal reasoning. In *Proc. 13th International Conference on Knowledge Representation and Reasoning (KR-2012)*, 2012.

[27] Peter Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.

[28] Peter Jonsson and Christer Bäckström. A unifying approach to temporal constraint reasoning. *Artif. Intell.*, 102(1):143–155, 1998.

[29] Peter Jonsson and Thomas Drakengren. A complete classification of tractability in RCC-5. *J. Artif. Intell. Res.*, 6:211–221, 1997.

[30] Peter Jonsson and Victor Lagerkvist. Upper and lower bounds on the time complexity of infinite-domain CSPs. In *Proc. 21st International Conference on Principles and Practice of Constraint Programming (CP-2015)*, pages 183–199, 2015.

[31] Markus Junker and Martin Ziegler. The 116 reducts of $(\mathbb{Q}, <, a)$. *J. Symb. Log.*, 73(3):861–884, 2008.

[32] Arne Kreutzmann and Diedrich Wolter. Qualitative spatial and temporal reasoning with AND/OR linear programming. In *Proc. 21st European Conference on Artificial Intelligence (ECAI-2014*, pages 495–500, 2014.

[33] C. Langford. Some theorems on deducibility. *Annals of Mathematics*, 28:16–40, 1927.

[34] Sanjiang Li. On topological consistency and realization. *Constraints*, 11(1):31–51, 2006.

[35] Sanjiang Li, Weiming Liu, and Sheng-sheng Wang. Qualitative constraint satisfaction problems: An extended framework with landmarks. *Artif. Intell.*, 201:32–58, 2013.

[36] Sanjiang Li and Mingsheng Ying. Extensionality of the RCC8 composition table. *Fundam. Inform.*, 55(3-4):363–385, 2003.

[37] Gérard Ligozat and Jochen Renz. What is a qualitative calculus? A general framework. In *Proceedings of PRICAI*, pages 53–64, 2004.

[38] Carsten Lutz and Maja Milicic. A tableau algorithm for description logics with concrete domains and general tboxes. *J. Autom. Reasoning*, 38(1-3):227–259, 2007.

[39] Dugald Macpherson. A survey of homogeneous structures. *Discrete Mathematics*, 311(15):1599–1634, 2011.

[40] Bernhard Nebel and Hans-Jürgen Bürckert. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *J. ACM*, 42(1):43–66, 1995.

[41] Christos Papadimitriou. On the complexity of integer programming. *J. ACM*, 28(4):765–768, 1981.

[42] David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *KR*, pages 165–176, 1992.

[43] Jochen Renz and Bernhard Nebel. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. *Artif. Intell.*, 108(1–2):69–123, 1999.

[44] Michael Sioutis and Manolis Koubarakis. Consistency of chordal RCC-8 networks. In *Proc. 24th International Conference on Tools with Artificial Intelligence (ICTAI-2012)*, pages 436–443, 2012.

[45] Frank Wolter and Michael Zakharyaschev. Spatio-temporal representation and reasoning based on RCC-8. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning*, pages 3–14. Morgan Kaufmann, 2000.