Datalog and Constraint Satisfaction with Infinite Templates

Manuel Bodirsky[1] and Víctor Dalmau[2]

[1] CNRS/LIX, École Polytechnique, bodirsky@lix.polytechnique.fr
[2] Universitat Pompeu Fabra, victor.dalmau@upf.edu

Abstract. On finite structures, there is a well-known connection between the expressive power of Datalog, finite variable logics, the existential pebble game, and bounded hypertree duality. We study this connection for infinite structures. This has applications for constraint satisfaction with infinite templates. If the template Γ is ω-categorical, we present various equivalent characterizations for whether the constraint satisfaction problem (CSP) for Γ can be solved by a Datalog program. We also show that CSP(Γ) can be solved in polynomial time for arbitrary ω-categorical structures Γ if the input is restricted to instances of bounded tree-width. Finally, we prove universal-algebraic characterizations of those ω-categorical templates whose CSP has Datalog width 1, and for those whose CSP has strict Datalog width k.
An extended abstract of this paper appeared in the proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science (STACS'06) [6].
keywords: Logic in computer science, computational complexity, constraint satisfaction, Datalog, countably categorical structures

1 Introduction

In a constraint satisfaction problem we are given a set of variables and a set of constraints on these variables, and want to find an assignment of values from some domain D to the variables such that all the constraints are satisfied. The computational complexity of the constraint satisfaction problem depends on the constraint language that is used in the instances of the problem. For *finite* domains D, the complexity of the constraint satisfaction problem attracted a lot of attention in recent years [2,9,15,17,18,21,26,29,30,32,37]; this list covers only a very small fraction of relevant publications, and we refer to a recent survey paper for a more complete account [8].

Constraint satisfaction problems where the domain D is infinite have been studied in Artificial Intelligence and the theory of binary relation algebras [19, 34], with applications for instance in temporal and spatial reasoning. Well-known examples of such binary relation algebras are the *point algebra*, the *containment algebra*, *Allen's interval algebra*, and the *left linear point algebra*. The corresponding constraint satisfaction problems received a lot of attention; see [14, 19, 27, 34] and the references therein.

Constraint satisfaction problems can be modeled as homomorphism problems [21]. For detailed formal definitions of relational structures and homomorphisms, see Section 2. Let Γ be a (finite or infinite) structure with a finite relational signature τ. Then the *constraint satisfaction problem (CSP)* for Γ is the following computational problem.

CSP(Γ)
INSTANCE: A finite τ-structure S.
QUESTION: Is there a homomorphism from S to Γ?

The structure Γ is called the *template* of the constraint satisfaction problem CSP(Γ). For example, if the template is the dense linear order of the rational numbers $(\mathbb{Q}, <)$, then it is easy to see that CSP(Γ) is the well-known problem of digraph-acyclicity.

A class $\mathcal{C}$ is said to be *closed under inverse homomorphisms* (sometimes also *anti-monotone*) if $B \in \mathcal{C}$ implies that $A \in \mathcal{C}$, whenever there is a homomorphism from A to B [22]. Clearly, the set of all positive instances of a CSP is closed under inverse homomorphisms. Conversely, if the set of positive instances of a computational problem is closed under inverse homomorphisms and disjoint unions, then it can be formulated as a CSP with a countably infinite template. This is an observation due to Feder and Vardi, and easy to prove, and it shows in particular that every CSP can be formulated with an at most countable structure.

Datalog. Datalog is the language of logic programs without function symbols, see e.g. [20, 32]. Datalog is an important algorithmic tool to study the complexity of CSPs. For constraint satisfaction with finite domains, this was first investigated systematically by Feder and Vardi [21]. For CSPs with infinite domains, one of the most studied algorithms are the *arc-consistency* and the *path-consistency* algorithm, which can also be formulated by Datalog programs.

Let τ be a relational signature; the relation symbols in τ will also be called the *input relation symbols*. A Datalog program consists of a finite set of Horn clauses $C_1, \ldots, C_k$ (they are also called the *rules* of the Datalog program) containing atomic formulas with relation symbols from the signature τ, together with atomic formulas with some new relation symbols. These new relation symbols are called *IDBs* (short for *intensional database*). Each clause is a set of literals where at most one of these literals is positive. The positive literals do not contain an input relation. Before we give formal definitions of the semantics of a Datalog program in Section 2, we show an instructive example.

$$\begin{aligned} \mathtt{tc}(x,y) &:- \mathsf{edge}(x,y) \\ \mathtt{tc}(x,y) &:- \mathtt{tc}(x,u), \mathtt{tc}(u,y) \\ \mathsf{false} &:- \mathtt{tc}(x,x) \end{aligned}$$

Here, the binary relation edge is the only input relation, tc is a binary relation computed by the program, and false is a 0-ary relation computed by the program. The Datalog program computes with the help of the relation tc the transitive closure of the edges in the input relation, and derives false if and only if the input (which can be seen as a digraph defined on the variables) contains a directed cycle. In general, we say that a problem is *solved* by a Datalog program, if the distinguished 0-ary predicate false is derived on an instance of the problem if and only if the instance has no solution. This will be made precise in Section 2.

We say that a Datalog program Π has *width* (l, k), $0 \leq l \leq k$, if it has at most l variables in rule heads and at most k variables per rule (we also say that Π is an (l, k)-*Datalog program*). A problem is *of width* (l, k), if it can be solved by an (l, k)-Datalog program. The problem of acyclicity has for instance width $(2, 3)$, as demonstrated above. A problem has *width* l if it is of width (l, k) for some $k \geq l$, and it is of *bounded width*, if it has width l for some $l \geq 0$. It is easy to see that all bounded width problems are tractable, since the rules can derive only a polynomial number of facts. It is an open question whether there is an algorithm that decides for a given finite template T whether CSP(T) can be solved by a Datalog program of width (l, k). Similarly, we do not know how to decide width l, or bounded width, with the notable exception of width one [21] (see also [18]).

The vast majority of the known results regarding Datalog programs only takes the number of variables per rule into account. In this work, we are interested in capturing a finer distinction, in which the number of variables in the head of the rule plays an important role. The double parameterization where we consider both the number l of variables per head and the number k of variables per rule, is less common, but more general, and has already been considered in the literature on constraint satisfaction and Datalog [21].

Results. We study the connection between the expressive power of Datalog, finite variable logics, the existential pebble game, and bounded hypertree duality. We show that this connection fails for general infinite structures (Section 3), but holds if the template is ω-*categorical* (Section 4.3); the concept of ω-categoricity is of central importance in model theory and will be introduced in Section 4.1. It is well-known that all the constraint satisfaction problems for the binary relation algebras (and their fragments) mentioned above and many problems in temporal and spatial reasoning can all be formulated with ω-categorical structures.

Our results on the connection between bounded hypertree duality and Datalog have applications for constraint satisfaction with ω-categorical templates. We show that $\mathrm{CSP}(\Gamma)$ can be solved in polynomial time if Γ is ω-categorical and the input is restricted to instances of bounded tree-width (in fact, it suffices that the cores of the instances have bounded tree-width).

We also investigate which constraint satisfaction problems can be solved with a Datalog program in polynomial time, if no restriction is imposed on the input instances (Section 6). In particular, we prove a characterization of constraint satisfaction problems with ω-categorical templates Γ having width $(1, k)$, generalizing a result from [18].

To obtain this result, we show that every problem that is closed under disjoint unions and has Datalog width one can be formulated as a constraint satisfaction problem with an ω-categorical template (Section 5); here we apply a recent model-theoretic result of Cherlin, Shelah, and Shi [12]. Another important tool to characterize the expressive power of Datalog for constraint satisfaction is the notion of *canonical Datalog programs*. This concept was introduced by Feder and Vardi for finite templates; we present a generalization to ω-categorical templates. We prove that a CSP with an ω-categorical template can be solved by an (l, k)-Datalog program if and only if the canonical (l, k)-Datalog program for Γ solves the problem (Section 4.2).

A special case of width 1 problems are problems that can be decided by establishing *arc-consistency* (sometimes also called *hyperarc-consistency*), which is a well-known and intensively studied technique in artificial intelligence. We show that if a constraint satisfaction problem with an ω-categorical template can be decided by establishing arc-consistency, then it can also be formulated as a constraint satisfaction problem with a finite template (Section 6).

Finally, we characterize *strict width* l, a notion that was again introduced for finite templates and for $l \geq 2$ in [21]; for a formal definition see Section 6. Jeavons et al. [29] say that in this case *establishing strong k-consistency ensures global consistency.* For finite templates, strict width l can be characterized by an algebraic closure condition [21, 29]. In Section 7 we generalize this result to ω-categorical templates Γ with a finite signature, and show that $\mathrm{CSP}(\Gamma)$ has strict width l if and only if for every finite subset A of Γ there is an $l+1$-ary polymorphism of Γ that is a *near-unanimity operation on* A, i.e., it satisfies the identity $f(x, \dots, x, y, x, \dots, x) = x$ for all $x, y \in A$ from the domain.

2 Definitions and Basic Facts

A *relational signature* τ is a (here always at most countable) set of *relation symbols* R_i, each associated with an *arity* k_i. A *(relational) structure* Γ *over relational signature* τ (also called τ-*structure*) is a set D_Γ (the *domain*) together with a relation $R_i \subseteq D_\Gamma^{k_i}$ for each relation symbol of arity k_i. If necessary, we write R^Γ to indicate that we are talking about the relation R belonging to the structure Γ. For simplicity, we denote both a relation symbol and its corresponding relation with the same symbol. For a τ-structure Γ and $R \in \tau$ it will also be convenient to say that $R(u_1, \dots, u_k)$ *holds in* Γ iff $(u_1, \dots, u_k) \in R$. We sometimes use the shortened notation $\overline{x}$ for a vector $x_1, \dots, x_n$ of any length.

The Gaiffman graph (sometimes also called the *shadow*) of a relational structure A is a graph on the vertex set $v_1, \dots, v_n$ where two distinct vertices v_k and v_l are adjacent if there is a relation in A that is imposed on both v_k and v_l, i.e., there is a relation R such that A satisfies $R(v_{i_1}, \dots, v_{i_j})$ and $k, l \in \{i_1, \dots, i_j\}$.

2.1 Homomorphisms

Let Γ and Γ' be τ-structures. A *homomorphism* from Γ to Γ' is a function f from D_Γ to $D_{\Gamma'}$ such that for each n-ary relation symbol R in τ and each n-tuple $\overline{a}$, if $\overline{a} \in R^\Gamma$, then $(f(a_1), \dots, f(a_n)) \in R^{\Gamma'}$. In this case we say that the map f *preserves* the relation R. Two structures Γ_1 and Γ_2 are called *homomorphically equivalent*, if there is a homomorphism from Γ_1 to Γ_2 and a homomorphism from Γ_2 to Γ_1. A *strong homomorphism* f satisfies the stronger condition that for each n-ary relation symbol in τ and each n-tuple $\overline{a}$, $\overline{a} \in R^{\Gamma}$ if and only if $(f(a_1), \ldots, f(a_n)) \in R^{\Gamma'}$. An *embedding* of a Γ in Γ' is an injective strong homomorphism, an *isomorphism* is a surjective embedding. Isomorphisms from Γ to Γ are called *automorphisms*.

A partial mapping h from a relational structure A to a relational structure B is called a *partial homomorphism* (from A to B) if h is a homomorphism from the the restriction of A to the range of h to B.

The *disjoint union* of two τ-structures Γ and Γ' is a τ-structure that is defined on the disjoint union of the domains of Γ and Γ'. A relation holds on vertices of the disjoint union if and only if it either holds in Γ or in Γ'. A τ-structure is called *connected* iff it is not the disjoint union of two τ-structures with a non-empty domain.

2.2 First-order Logic

First-order formulae φ over the signature τ (or, short, τ-formulae) are 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 usual Tarskian style. A τ-formula without free variables is called a τ-sentence. We write $\Gamma \models \varphi$ iff the τ-structure Γ is a model for the τ-sentence φ; this notation is lifted to sets of sentences in the usual way. A good introduction to logic and model theory is [28].

As usual, we 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. If we add relations to a given τ-structure Γ, then the resulting structure Γ' with a larger signature $\tau' \supset \tau$ is called a *τ'-expansion* of Γ, and Γ is called a *τ-reduct* of Γ'. This should not be confused with the notions of *extension* and *restriction*. Recall from [28]: If Γ and Γ' are structures of the same signature, with $D_\Gamma \subseteq D_{\Gamma'}$, and the inclusion map is an embedding, then we say that Γ' is an *extension* of Γ, and that Γ a *restriction* of Γ'.

A first-order formula φ is said to be *primitive positive* (we say φ is a *pp-formula*, for short) iff it is of the form $\exists \overline{x}(\varphi_1(\overline{x}) \wedge \cdots \wedge \varphi_k(\overline{x}))$ where $\varphi_1, \ldots, \varphi_k$ are atomic formulas (which might be equality relations of the form $x = y$).

2.3 Canonical queries

A basic concept to link structure homomorphisms and logic is the *canonical conjunctive query* ϕ^A of a relational structure A, which is a first-order formula of the form $\exists v_1, \ldots, v_n. \psi_1 \wedge \cdots \wedge \psi_m$, where $v_1, \ldots, v_n$ are the vertices of A, and $\{\psi_1, \ldots, \psi_m\}$ is the set of atomic formulas of the form $R(v_{i_1}, \ldots, v_{i_j})$ that holds in A.

It is a fundamental property of the canonical query ϕ^A that ϕ^A holds in a structure Γ if and only if there is a homomorphism from A to Γ [11].

2.4 Finite variable logics

Let $0 \leq l < k$ be positive integers. The class $\exists\mathcal{L}^k_{\infty\,\omega}$ is defined as the class of formulas that have at most k variables and are obtained from atomic formulas using infinitary conjunction, infinitary disjunction, and existential quantification only. The class $\bigcup_{k\geq 0} \exists\mathcal{L}^k_{\infty\,\omega}$ is denoted by $\exists\mathcal{L}^\omega_{\infty\,\omega}$.

We want to bring the parameter l into the picture. To this end we define the following refinement of $\exists\mathcal{L}^k_{\infty\,\omega}$. A conjunction $\bigwedge \Psi$ is called *l-bounded* if Ψ is a collecion of $\exists\mathcal{L}^\omega_{\infty\,\omega}$ formulas ψ that are quantifier-free or have at most l free variables. Similarly, an disjunction $\bigvee \Psi$ is called *l-bounded* if Ψ is a collecion of $\exists\mathcal{L}^\omega_{\infty\,\omega}$ formulas ψ that are quantifier-free or have at most l free variables. The set of $\exists\mathcal{L}^{l,k}_{\infty\,\omega}$ formulas is defined as the restriction of $\exists\mathcal{L}^k_{\infty\,\omega}$ obtained by only allowing infinitary l-bounded conjunction and l-bounded disjunction instead of full infinitary conjunction and disjunction. Note that $\bigcup_{0\leq l<k} \exists\mathcal{L}^{l,k}_{\infty\,\omega}$ equals $\exists\mathcal{L}^{k}_{\infty\,\omega}$. The logic $\exists\mathcal{L}^{k}_{\infty\,\omega}$ was introduced (under a different name) by Kolaitis and Vardi as an existential negation-free variant of well-studied infinitary logics to study the exprssive power of Datalog [31]; in subsequent work, they used the name $\exists\mathcal{L}^{k}_{\infty\,\omega}$ to denote this logic [32] and we follow this convention.

We denote by $L^{l,k}$ the logic $\exists\mathcal{L}^{l,k}_{\infty\,\omega}$ without disjunctions and with just *finite* l-bounded conjunctions. In other words, a formula in $L^{l,k}$ is composed out of existential quantification and finitary l-bounded conjunction, and uses only k distinct variables. The language L^k, where only the parameter k, but not the parameter $l \leq k$ is specified, has been studied for example by Kolaitis and Vardi [32] and later by Dalmau, Kolaitis, and Vardi [16].

2.5 Datalog

We now formally define Datalog. Our definition will be purely operational; for the standard semantical approach to the evaluation of Datalog programs see [20,32]. A Datalog program is a finite set of Horn clauses, i.e., clauses of the form $\psi : -\phi_1, \ldots, \phi_l$, where $l \geq 0$ and where $\psi, \phi_1, \ldots, \phi_l$ are atomic formulas. The formula ψ is called the *head* of the rule, and $\phi_1, \ldots, \phi_l$ is called the *body*. For technical reasons, we assume that all variables in the head also occur in the body. The relation symbols occurring in the head of some clause are called *intentional database predicates* (or *IDBs*), and all other relation symbols in the clauses are called *extensional database predicates* (or *EDBs*). A Datalog program has *width (l,k)* if all IDBs are at most l-ary, and if all rules have at most k distinct variables.

We might use Datalog programs to solve constraint satisfaction problems CSP(Γ) as follows. Let Π be a Datalog program whose extensional symbols are from Γ, and let L be the set of intentional relation symbols of Π. We assume that there is one distinguished 0-ary intentional relation symbol false. Now, suppose we are given an instance S of CSP(Γ). An *evaluation* of Π on S proceeds in steps $i = 0, 1, \ldots$. At each step i we maintain a set of literals S^i with relation symbols from L; it always holds that $S^i \subset S^{i+1}$. Each clause of Π is understood as a rule that may derive a new literal (with a relation symbol from L) from the literals in S^i. Initially, we have $S^0 := S$. Now suppose that $R_1(x^1_1, \ldots, x^1_{k_1}), \ldots, R_l(x^l_1, \ldots, x^l_{k_l})$ are literals in S^i, and $R_0(y^0_1, \ldots, y^0_{k_0}) :- R_1(y^1_1, \ldots, y^1_{k_1}), \ldots, R_l(y^l_1, \ldots, y^l_{k_l})$ is a rule from Π, where $y^i_j = y^{i'}_{j'}$ if and only if $x^i_j = x^{i'}_{j'}$. Then $R_0(x^0_1, \ldots, x^0_l)$ is the newly derived literal in S^{i+1}, where $x^0_j = x^i_{j'}$ if and only if $y^0_j = y^i_{j'}$. The procedure stops if no new literal can be derived. We say that Π is *sound* for CSP(Γ) if S does not homomorphically map to Γ whenever Π derives false on S. We say that Π *solves* CSP(Γ) if Π derives false on S if and only if S does not homomorphically map to Γ.

2.6 The existential pebble game

The existential k-pebble game was studied in the context of constraint satisfaction for instance in [16,21,32]. As in [21], we study this game with a second parameter, and first define the *existential* (l, k)*-pebble game*. The usual existential k-pebble game is exactly the existential $(k-1, k)$-pebble game in our sense. The second parameter is necessary to obtain the strongest formulations of our results.

The game is played by the players Spoiler and Duplicator on (possibly infinite) structures A and B of the same relational signature. Each player has k pebbles, $p_1, \ldots, p_k$ for Spoiler and $q_1, \ldots, q_k$ for Duplicator. Spoiler places his pebbles on elements of A, Duplicator her pebbles on elements of B. Initially, no pebbles are placed. In each round of the game Spoiler picks $k-l$ pebbles. If some of these pebbles are already placed on A, then Spoiler removes them from A, and Duplicator responds by removing the corresponding pebbles from B. Spoiler places the $k - l$ pebbles on elements of A, and Duplicator responds by placing the corresponding pebbles on elements of B. Let $i_1, \ldots, i_m$ be the indices of the pebbles that are placed on A (and B) after the i-th round. Let $a_{i_1}, \ldots, a_{i_m}$ ($b_{i_1}, \ldots, b_{i_m}$) be the elements of A (B) pebbled with the pebbles $p_{i_1}, \ldots, p_{i_m}$ ($q_{i_1}, \ldots, q_{i_m}$) after the i-th round. If the partial mapping h from A to B defined by $h(a_{i_j}) = b_{i_j}$, for $1 \leq j \leq m$, is not a partial homomorphism from A to B, then the game is over, and Spoiler wins. Duplicator wins if the game continues forever.

We would like to characterize the situations where *Duplicator* can win the game, i.e., where Spoiler does not have a winning strategy. It turns out that in this case Duplicator can always play *memoryless* in the sense that the decisions of Duplicator are only based on the current position of the pebbles, and not the previous decisions of the game. This holds for a much larger class of pebble games, see [23].

Definition 1. *A* (positional) winnning strategy for Duplicator *for the existential* (l, k)*-pebble game on* A, B *is a non-empty set* $\mathcal{H}$ *of partial homomorphisms from* A *to* B *such that*

- $\mathcal{H}$ *is closed under restrictions of its members, and*
- *for all functions* h *in* $\mathcal{H}$ *with* $|\mathrm{dom}(h)| = d \leq l$ *and for all* $a_1, \ldots, a_{k-d} \in A$ *there is an extension* $h' \in \mathcal{H}$ *of* h *such that* h' *is also defined on* $a_1, \ldots, a_{k-d}$.

2.7 Treewidth

In what remains of the section we define the treewidth of a relational structure. As in [21], we need to extend the ordinary notion of treewidth for relational structures in such a way that we can introduce the additional parameter l.

Let $0 \leq l < k$ be positive integers. An (l, k)*-tree* is defined inductively as follows:

- A k-clique is a (l, k)-tree
- For every (l, k)-tree G and for every l-clique induced be nodes $v_1, \ldots, v_l$ in G, the graph G' obtained by adding $k - l$ new nodes $v_{l+1}, \ldots, v_k$ to G and adding edges (v_i, v_j) for all $i \neq j$ with $i \in \{1, \ldots, k\}$, $j \in \{l + 1, \ldots, k\}$ (so that $v_1, \ldots, v_k$ forms a k-clique) is also a (l, k)-tree.

A *partial* (l, k)*-tree* is a (not necessarily induced) subgraph of an (l, k)-tree.

Definition 2. *Let* $0 \leq l < k$ *and let* τ *be a relational signature. We say that a* τ*-structure* S *has* treewidth (l, k) *if the Gaiffman graph of* S *is a partial* (l, k)*-tree.*

If a structure has treewidth $(k, k + 1)$ we also say that it has *treewidth* k, and it is not difficult to see that these structures are precisely the structures of treewidth k in the sense of [32]. It is also possible to define partial (l, k)-trees by using tree-decompositions.

Definition 3. *A* tree-decomposition *of a graph* G *is a tree* T *such that*

1. *The nodes of* T *are sets of nodes of* G*;*
2. *Every edge of* G *is entirely contained in some node of* T*;*
3. *If a node* v *belongs to two nodes* x*,* y *of* T *it must also be in every node in the unique path from* x *to* y.

A tree-decomposition T is said to be of *width* (l, k) if every node of T contains at most k nodes of G and the intersection of two different nodes of T has size at most l. The following is a straightforward generalization of a well-known fact for single parameter k, and the proof can be obtained by adapting for instance the proof given in [3].

Proposition 1. *A graph is a partial* (l, k)*-tree if and only if it has a tree-decomposition of width* (l, k).

It was shown in [32] that the canonical query for a structure S of tree-width k can be expressed in the logic L^{k+1}. We show an analogous statement for both parameters l and k.

Lemma 1. *Let* A *be a finite structure of treewidth* (l, k)*. Then the canonical query* ϕ^A *for* A *is expressible in* $L^{l,k}$.

Proof. Let A be a finite structure of treewidth (l,k), let G be its Gaiffman graph, and let T be a tree-decomposition of G. Let us view T as a rooted tree with root $t = \{a_1, \dots, a_{k'}\}$, $k' \leq k$. We shall show by structural induction on T that there exists a formula $\phi^A(y_1, \dots, y_{k'})$ in $L^{l,k}$ with free variables $y_1, \dots, y_{k'}$ such that for every structure B and elements $b_1, \dots, b_{k'}$ in B the two following sentences are equivalent:

1. The partial mapping from A to B that maps a_i to b_i for $1 \leq i \leq k'$ can be extended to a homomorphism from A to B;
2. $Q(b_1, \dots, b_{k'})$ holds in B.

The base case is when the tree contains only one node t. In this case ϕ^A is obtained by removing the existential quantifier in the canonical conjunctive query of A. For the inductive step, let $t_1, \dots, t_m$ be the children of the root t in T. Consider the m subtrees $T_1, \dots, T_m$ of T obtained by removing t. For every $i = 1, \dots, m$ we root T_i at t_i and consider the substructure A_i of A induced by the set of all nodes of A contained in some node of T_i. Then, T_i is a tree-decomposition of A_i and the induction hypothesis provides a formula ϕ^{A_i} for which (1) and (2) are equivalent. Let $\phi^A(y_1, \dots, y_{k'})$ be the formula $\bigwedge \mathcal{S}$ where $\mathcal{S}$ is the following set of formulas:

(a) For each $i = 1, \dots, m$ the set $\mathcal{S}$ contains the formula obtained by existentially quantifying all free variables y_j in ϕ^{A_i} where $a_j \notin t$. Note that the resulting formula has at most l-free variables.
(b) The set S contains the canonical database of the substructure of A induced by the nodes in t in which all existential quantifiers have been removed (as in the base case).

To show that (1) and (2) are equivalent, let B be an arbitrary structure. By the properties of the tree-decomposition we know that h is a homomorphism from A to B that maps a_i to b_i if and only if for all $i = 1, \dots, m$ the restriction of h to A_i is a homomorphism from A_i to B and the restriction of h to the elements of t is a partial homomorphism from A to B as well. The former condition is equivalent to the fact that the assignment $y_{a_i} \to b_i$, satisfyes every formula of $\mathcal{S}$ included in stage (a). The latter condition is equivalent to the fact that the very same assignment satisfies the formula introduced in stage (b). □

3 General Infinite Structures

In this section, we recall facts known shown for finite structures [16, 31], and show that they fail if Γ has an infinite domain. The following theorem was proved by [31] for *finite* structures.

Theorem 1. *Let A, B be finite relational structures over the same signature τ. The the following are equivalent.*

1. *Duplicator has a winning strategy for the existential k-pebble game on A and B;*
2. *All τ-sentences in $\exists\mathcal{L}^k_{\infty\omega}$ that hold in A also hold in B;*
3. *Every finite τ-structure whose core has treewidth $k-1$ that homomorphically maps to A also homomorphically maps to B.*

We show that for infinite structures B it is in general not true that 3 implies 1.

Proposition 2. *There are infinite τ-structures A and B such that*

- *Duplicator does not have a winning strategy for the existential 2-pebble game on A and B;*
- *Every finite τ-structure C of tree-width 1 that homomorphically maps to A also maps to B.*

Proof. Let B be a disjoint union of directed paths of length 1,2,... Consider $A = \vec{C_3}$. Every finite τ-structure C of tree-width 1 is a finite oriented tree, and therefore homomorphically maps to A and to B. However, Spoiler clearly has a winning strategy. After Spoiler places his first pebble, Duplicator has to place his first pebble on a path of length l in B. By walking with his two pebbles in one direction on the directed cycle A, Spoiler can trap Duplicator after l rounds of the game. □

From now on, we are interested in computational problems of the form $\mathrm{CSP}(\Gamma)$ for a countably infinite structure Γ. The following states results obtained in [16, 21, 31].

Theorem 2. *Let Γ be a τ-structure over a finite domain. Then the following statements are equivalent.*

1. *For all finite τ-structures A, if Duplicator has a winning strategy for the existential k-pebble game on A and Γ, then A is in $CSP(\Gamma)$.*
2. *The complement of $CSP(\Gamma)$ can be formulated in $\exists\mathcal{L}^k_{\infty\,\omega}$.*
3. *For all finite τ-structures A, if every finite τ-structure C of tree-width k that homomorphically maps to A also maps to Γ, then A homomorphically maps to Γ.*
4. *There is an* obstructions set *$\mathcal{N}$, i.e., a set of finite structures of tree-width $k-1$ such that a finite τ-structure A is homomorphic to Γ if and only if no structure in $\mathcal{N}$ is homomorphic to A.*

Also Theorem 2 fails for structures Γ over an infinite domain. Intuitively, the reason is that the expressive power of infinitary disjunction is relatively larger for $\mathrm{CSP}(\Gamma)$ if Γ has an infinite domain.

Proposition 3. *There is a τ-structure Γ over an infinite domain such that*

- *the complement of $CSP(\Gamma)$* can *be formulated in $\exists\mathcal{L}^2_{\infty\,\omega}$;*
- *there is a finite τ-structure A such that Duplicator wins the existential* 2*-pebble game on A and Γ, but A is not in $CSP(\Gamma)$.*

Proof. We choose Γ to be $(\mathbb{Q}, <)$, and let $\vec{C_3}$ be the directed cycle on three vertices. Duplicator wins the existential 2-pebble game on $\vec{C_3}$ and Γ, but there is no homomorphism from $\vec{C_3}$ to Γ.

The complement of $\mathrm{CSP}(\mathbb{Q}, <)$ can be formulated in $\exists\mathcal{L}^2_{\infty\,\omega}$. Let Φ be an $\exists\mathcal{L}^2_{\infty\,\omega}$-sentence that expresses that a structure contains copies of $(\{1, \dots, n\}, <)$ for arbitrarily large n. The structures that are not in $\mathrm{CSP}(\mathbb{Q}, <)$ are precisely the directed graphs containing a cycle; clearly, Φ holds precisely on those finite directed graphs having a cycle. □

4 Datalog for ω-categorical structures

The concept of ω-categoricity is of central interest in model theory [10,28]. We show that many facts that are known about Datalog programs for finite structures extend to ω-categorical structures.

4.1 Countably Categorical Structures

A countable structure Γ is called *ω-categorical*, if all countable models of the first-order theory of Γ are isomorphic to Γ. The following is a well-known and deep connection that shows that ω-categoricity of Γ is a property of the automorphism group of Γ, without reference to concepts from logic (see [28]). The *orbit* of an n-tuple $\overline{a}$ from Γ is the set $\{\alpha(\overline{a}) \mid \alpha \text{ is an automorphism of } \Gamma\}$.

Theorem 3 (Engeler, Ryll-Nardzewski, Svenonius; see e.g. [28]). *The following properties of a countable structure Γ are equivalent:*

1. *the structure Γ is ω-categorical;*
2. *for each $n \geq 1$, there are finitely many orbits of n-tuples in the automorphism group $Aut(\Gamma)$ of Γ;*
3. *for each $n \geq 1$, there are finitely many inequivalent first-order formulas with n free variables over Γ;*
4. *A relation R is first-order definable in Γ if and only if it is preserved by all automorphisms of Γ.*

Examples. An example of an ω-categorical directed graph is the set of rational numbers with the dense linear order $(\mathbb{Q}, <)$ [28]. The (tractable) constraint satisfaction problem for this structure is digraph acyclicity. Another important example it the *universal triangle free graph* $\not\triangle$. This structure is the up to isomorphism unique countable K_3-free graph with the following *extension property*: whenever S is a subset and T is a disjoint independent subset of the vertices in $\not\triangle$, then $\not\triangle$ contains a vertex $v \notin S \cup T$ that is linked to no vertex in S and to all vertices in T. Since the extension property can be formulated by an (infinite) set of first-order sentences, it follows that $\not\triangle$ is ω-categorical [28]. The structure $\not\triangle$ is called the *universal* triangle free graph, because every other countable triangle free graph embeds into $\not\triangle$. Hence, $\mathrm{CSP}(\not\triangle)$ is clearly tractable. However, this simple problem can not be formulated as a constraint satisfaction problem with a finite template [21, 37].

For ω-categorical templates we can apply the so-called *algebraic approach to constraint satisfaction* [8, 9, 30]. This approach was originally developed for constraint satisfaction with finite templates, but several fundamental theorems also hold for ω-categorical templates [7, 25].

The following lemma states an important property of ω-categorical structures needed several times later. The proof contains a typical proof technique for ω-categorical structures.

Lemma 2. *Let Γ be a finite or infinite ω-categorical structure with relational signature τ, and let Δ be a countable relational structure with the same signature τ. If there is no homomorphism from Δ to Γ, then there is a finite substructure of Δ that does not homomorphically map to Γ.*

Proof. Suppose every finite substructure of Δ homomorphically maps to Γ. We show the contraposition of the lemma, and prove the existence of a homomorphism from Δ to Γ. Let $a_1, a_2, \ldots$ be an enumeration of Δ. We construct a directed acyclic graph with finite out-degree, where each node lies on some level $n \geq 0$. The nodes on level n are equivalence classes of homomorphisms from the substructure of Δ induced by $a_1, \ldots, a_n$ to Γ. Two such homomorphisms f and g are equivalent, if there is an automorphism α of Γ such that $f\alpha = g$. Two equivalence classes of homomorphisms on level n and $n+1$ are adjacent, if there are representatives of the classes such that one is a restriction of the other. Theorem 3 asserts that Γ has only finitely many orbits of k-tuples, for all $k \geq 0$ (clearly, this also holds if Γ is finite). Hence, the constructed directed graph has finite out-degree. By assumption, there is a homomorphism from the structure induced by $a_1, a_2, \ldots, a_n$ to Γ for all $n \geq 0$, and hence the directed graph has vertices on all levels. König's Lemma asserts the existence of an infinite path in the graph, which defines a homomorphism from Δ to Γ. $\square$

4.2 Canonical Datalog Programs

In this section we define the canonical Datalog program of an ω-categorical structure Γ. We then prove that $\mathrm{CSP}(\Gamma)$ can be solved by an (l, k)-Datalog program if and only if the canonical (l, k)-Datalog program solves the problem.

For *finite* templates T with a relational signature τ the canonical Datalog program for T was defined in [21]. This motivates the following definition of canonical Datalog programs for ω-categorical structures Γ. The *canonical (l, k)-Datalog program* for Γ contains an IDB for every at most l-ary primitive positive definable relation in Γ. The empty 0-ary relation serves as false. The input relation symbols are precisely the relation symbols from τ.

By ω-categoricity, Theorem 3 asserts that there is a finite number of inequivalent logical implications $\Psi(\overline{x})$ of the form $(\exists \overline{y}(\psi_1(\overline{x}, \overline{y}) \wedge \cdots \wedge \psi_j(\overline{x}, \overline{y})) \rightarrow R(\overline{x})$ in Γ having at most k variables, where $\psi_1, \ldots, \psi_i$ are atomic formulas of the form $R_1(\overline{z}_1), \ldots, R_j(\overline{z}_j)$ for IDBs or EDBs $R_1, \ldots, R_j$ and an IDB R. For each of these inequivalent implications, we introduce a rule

$$R(\overline{x}){:-}R_1(\overline{z}_1), \ldots, R_j(\overline{z}_j)$$

into the canonical Datalog program, if $\forall \overline{x}. \Psi(\overline{x})$ is valid in Γ, in other words, if the pp-definable relation for which the IDB R was introduced is implied by $\exists \overline{y}. \psi_1(\overline{x}, \overline{y}) \wedge \cdots \wedge \psi_j(\overline{x}, \overline{y})$ in Γ. Since there are finitely many inequivalent implications Ψ, the canonical (l, k)-Datalog program is finite.

On a given instance a Datalog program can only derive a finite number of facts, which is polynomial in the size of the instance. Thus, Datalog programs can be evaluated in polynomial time. Observe that all stages during the evaluation of a canonical Datalog program on a given instance give rise to another instance S' of CSP(Γ'), where Γ' is the expansion of Γ by all at most l-ary primitive positive definable relations, and where S' contains all the derived tuples from these relations [20].

The following is easy to see.

Proposition 4. *Let Γ be ω-categorical. Then the canonical (l,k)-Datalog program for Γ is sound for CSP(Γ).*

Proof. We have to show that if the canonical (l,k)-Datalog program derives false on a given instance S, then S is unsatisfiable. Assume for contradiction that there is a homomorphism $f : S \to \Gamma$ although the canonical (l,k)-program for Γ derives false on the instance S. The derivation tree of false corresponds via f to a set of valid implications in the template. Finally, the implication of false corresponds to an implication of the 0-ary empty relation in the template, a contradiction. ◻

4.3 Datalog for Countably Categorical Structures

The following theorem is the promised link between Datalog, the existential pebble game, finite variable logics, and hypertree duality for ω-categorical structures. We present it in its most general form with both parameters l and k. The assumption of ω-categoricity will be used for the transition from 2 to 3 (note that the canonical Datalog program is only defined for ω-categorical structures).

Theorem 4. *Let Γ be a ω-categorical structure with a finite relational signature τ, and let S be a finite τ-structure. Then the following statements are equivalent.*

1. *Every sound (l,k)-Datalog program for CSP(Γ) does not derive* false *on S.*
2. *The canonical (l,k)-Datalog program for Γ does not derive* false *on S.*
3. *Duplicator has a winning strategy for the existential (l,k)-pebble game on S and Γ.*
4. *All sentences in $L^{l,k}$ that hold in S also hold in Γ.*
5. *Every finite τ-structure with a core of treewidth (l,k) that homomorphically maps to A also homomorphically maps to Γ.*

Proof. The implication from 1 to 2 follows from Theorem 4.

To show that 2 implies 3, let S be an instance such that the canonical Datalog program does not derive false. We define a winning strategy for Duplicator as follows. It contains all those partial mappings $f : S \to \Gamma$ with domain D of size at most k such that for every derived IDB $R(x_1, \ldots, x_d)$, where $x_1, \ldots, x_d \in D$, the tuple $(f(x_1), \ldots, f(x_d))$ belongs to R.

By construction, $\mathcal{H}$ is closed under subfunctions, contains only partial homomorphisms, and is non-empty (since false is not derived, $\mathcal{H}$ contains the partial mapping with the empty domain). We have to prove that $\mathcal{H}$ has the (l,k)-extension property. Let h be a function with domain $x_1, \ldots, x_d$ of size at most l and let D be any superset of $\{x_1, \ldots, x_d\}$ of size k. Let $R_1, \ldots, R_j$ be the IDBs that have been derived over the domain D. Then there is a rule of the form $R(\overline{x}){:-}\psi_1(\overline{z}_1), \ldots, \psi_j(\overline{z}_j)$ in the canonical Datalog program, where R is some d-ary IDB, which shows that $R(x_1, \ldots, x_d)$ can be derived. By construction of $\mathcal{H}$ we have that $(h(x_1), \ldots, h(x_d))$ belongs to $\mathcal{H}$. This implies that h can be extended to a partial homomorphism with domain D.

Next, we show the implication from 3 to 4. The proof closely follows the corresponding proof for finite structures given in [32], with the important difference that we have both parameters l and k in our proof, whereas previously the results has only been stated with the parameter k.

Suppose Duplicator has a winning strategy $\mathcal{H}$ for the existential (l,k)-pebble game on A and B. Let ϕ be a τ-sentence from $L^{l,k}$ that holds in A. We have to show that ϕ also holds in B. For that, we prove by induction on the syntactic structure of $L^{l,k}$ formulas that if $\psi(v_1, \ldots, v_m)$ is an $L^{l,k}$ formula that is an l-bounded conjunction or has at most l free variables (i.e., $m \leq l$), then for all $h \in \mathcal{H}$ and all elements $a_1, \ldots, a_m$ from the domain of h, if A satisfies $\psi(a_1, \ldots, a_m)$, then B satisfies $\psi(h(a_1), \ldots, h(a_m))$.

Clearly, choosing $m = 0$, this implies that ϕ holds in B.

The base case of the induction is obvious, since atomic formulas are preserved under homomorphisms. Next, suppose that $\psi(v_1, \ldots, v_m)$ is an l-bounded conjunction of a set of formulas Ψ. Then each formula in Ψ either has at most l free variables, or is quantifier-free. In both cases we can use the inductive hypothesis, and the inductive step follows directly.

Assume that the formula $\psi(v_1, \ldots, v_m)$ is of the form $\exists u_1, \ldots, u_n. \chi(v_1, \ldots, v_m, u_1, \ldots, u_n)$. If $m > l$, there is nothing to show. Otherwise, we choose χ and n such that n is largest possible. Therefore, χ is either an l-bounded conjunction or an atomic formula. Since ψ is from $L^{l,k}$, we know that $n + m \leq k$. We will use the inductive hypothesis for the formula $\chi(v_1, \ldots, v_m, u_1, \ldots, u_n)$. Let h be a homomorphism in $\mathcal{H}$. We have to show that if $a_1, \ldots, a_m$ are arbitrary elements from the domain of h such that A satisfies $\psi(a_1, \ldots, a_m)$, then B satisfies $\phi(h(a_1), \ldots, h(a_m))$.

Let $a'_1, \ldots, a'_m, a'_{m+1}, \ldots, a'_{m+n}$ be a sequence of elements from A such that

- A satisfies $\chi(a'_1, \ldots, a'_m, a'_{m+1}, \ldots, a'_{m+n})$
- If i is an index from $1, \ldots, m$ such that $u_i = v_j$ for some j from $1, \ldots, n$, then $a'_i = a'_{m+j}$.
- If i is an index from $1, \ldots, m$ such that u_i does not appear in $v_1, \ldots, v_n$, then $a'_i = a_i$.

Consider the restriction h^* of h to the subset $\{a_1, \ldots, a_m\}$ of the domain of h. Because of the first property of winning strategies $\mathcal{H}$, the homomorphism h^* is in $\mathcal{H}$. Since $m \leq l$, we can apply the forth property of $\mathcal{H}$ to h^* and $a'_{m+1}, \ldots, a'_{m+n}$, and there are $b_1, \ldots, b_n$ such that $h' = h^* \cup \{(a'_{m+i}, b_i) \mid 1 \leq i \leq n\}$ is in $\mathcal{H}$. By applying the induction hypothesis to $\chi(v_1, \ldots, v_m, u_1, \ldots, u_n)$ and to h', we infer that B satisfies $\chi(h'(a_1), \ldots, h'(a_m), h'(a_{m+1}), \ldots, h'(a'_{m+n}))$ and, hence, B satisfies $\psi(h(a_1), \ldots, h(a_m))$.

4 implies 5. Let T be a finite τ-structure T whose core T' has treewidth (l,k) such that T homomorphically maps to S. By Lemma 1 there exists an $L^{l,k}$-sentence ϕ such that ϕ holds in a structure B if and only if T' homomorphically maps to B. In particular, ϕ must hold in S. Then 5 implies that ϕ holds in Γ, and therefore T' homomorphically maps to Γ. But then we can compose the homomorphism from T to T' and the homomorphism from T' to Γ, which shows the claim.

We finally show that 5 implies 1. Assume 5, and suppose for contradiction that there is a sound (l,k)-Datalog program Π for Γ that derives false on A. The idea is to use the 'derivation tree of false' to construct a τ-structure S of tree-width (l,k) that homomorphically maps to A, but not to Γ. Formally, we start our construction of S with the empty structure without vertices. In order to prove that S has treewidth (l,k) we also describe an (l,k)-tree G such that the Gaiffman graph of S is a subgraph of G.

We consider the inference steps of Π on A in reverse order, starting from the last inference of the Datalog program, in which false was derived. If $R_0(y^0_1, \ldots, y^0_{k_0})$ is the constraint that is derived from the previously derived literals $R_1(y^1_1, \ldots, y^1_{k_1}), \ldots, R_s(y^s_1, \ldots, y^s_{k_s})$ in a step of the evaluation of Π on A, we inductively assume that we have already created vertices $v^0_1, \ldots, v^0_{k_0}$ for $y^0_1, \ldots, y^0_{k_0}$ in S (this is clearly true for the first inference step we consider, where $k_0 = 0$). We also inductively assume that we have already constructed an (l,k)-tree G such that the Gaiffman graph of S is a subgraph of G. Recall that $k_0 \leq l$.

We then create new vertices $v^1_1, \ldots, v^1_{k_1}, \ldots, v^s_1, \ldots, v^s_{k_s}$ in S and in G for all variables in $y^1_1, \ldots, y^1_{k_1}, \ldots, y^s_1, \ldots, y^s_{k_s}$ that do not appear in $y^0_1, \ldots, y^0_{k_0}$; note that there are at most $k - k_0$ many new variables. Otherwise, if y^j_i equals y^0_p, $1 \leq p \leq k_0$, we set v^j_i to v^0_p. If for some i from $1, \ldots, s$ the relation R_i is a relation from τ, then we also add the tuple $(v^i_{k_1}, \ldots, v^i_{k_i})$ to the relation R_i from S. In G, we add $k' = k - |\{v^1_1, \ldots, v^1_{k_1}, \ldots, v^s_1, \ldots, v^s_{k_s}\}|$ additional new vertices $v_1, \ldots, v_{k'}$. Moveover, we add in G edges (u,v) between distinct vertices u, v from $v^1_1, \ldots, v^1_{k_1}, \ldots, v^s_1, \ldots, v^s_{k_s}, v_1, \ldots, v_{k'}$. Note that then these vertices induce a k-clique in G, and clearly G is a (l,k)-tree such that the Gaiffman graph of S is a subgraph of G. Also observe that if we would then apply the program Π on S, and if Π would derive $R_1(v^1_1, \ldots, v^1_{k_1}), \ldots, R_s(v^s_1, \ldots, v^s_{k_s})$, then it would also derive $R_0(v^0_1, \ldots, v^0_{k_0})$. In this fashion we proceed for all inference steps of the Datalog program.

The resulting structure S has treewidth at most (l, k), because its Gaiffman graph is contained in the (l, k)-tree G. Moreover, there is a homomorphism from S to A: each vertex v in S was introduced for a variable x in A, and if we map v to x the resulting mapping clearly is a homomorphism. Now, suppose for contradiction that there is a homomorphism h from S to Γ. Then the observation we made in the last paragraph implies that the Datalog program derives false on S as well. Since the Datalog program is sound for CSP(Γ), there cannot be a homomorphism from S to Γ, a contradiction. ☐

4.4 Application to Constraint Satisfaction

We discuss an important consequence of Theorem 4 with many concrete applications: we prove that CSP(Γ) for ω-categorical Γ is tractable if the input is restricted to instances of tree-width (l, k). In fact, for the tractability result we only have to require that the *cores* of the input structures have bounded tree-width. A finite relational structure A is a *core* if every endomorphism of A is an automorphism of A. It is easy to see that every finite relational structure B is homomorphically equivalent to a core. The statement where we only require that the *core* of input structures has tree-width (l, k) is considerably stronger (also see [24]); the corresponding statement for finite structures and single parameter k has been observed in [16].

Corollary 1. *Let Γ be an ω-categorical structure with finite relational signature τ. Then every instance S of CSP(Γ) whose core has tree-width (l, k) can be solved in polynomial time by the canonical (l, k)-Datalog program.*

Proof. It is clear that an (l, k)-Datalog program can be evaluated on a (finite!) instance S of CSP(Γ) in polynomial time. If the canonical (l, k)-Datalog program derives false on S, then, because the canonical Datalog program is always sound, the instance S is not homomorphic to Γ. Now, suppose that the canonical Datalog program does not derive false on a finite structure S whose core has tree-width (l, k). Then, by Theorem 4, every τ-structure whose core has tree-width (l, k) that homomorphically maps to S also homomorphically maps to Γ. This holds in particular for S itself, and hence S is homomorphic to Γ. ☐

The following direct consequence of Theorem 4 yields other characterizations of bounded Datalog width.

Theorem 5. *Let Γ be a ω-categorical structure with a finite relational signature τ. Then the following statements are equivalent.*

1. *There is an (l, k)-Datalog program that solves CSP(Γ).*
2. *The canonical (l, k)-Datalog program solves CSP(Γ).*
3. *For all finite τ-structures A, if Duplicator has a winning strategy for the existential (l, k)-pebble game on A and Γ, then A is in CSP(Γ).*
4. *For all finite τ-structures A, if all sentences in $L^{l,k}$ that hold in A also hold in Γ, then A homomorphically maps to Γ.*
5. *For all finite τ-structures A, if every finite τ-structure S of tree-width (l, k) that homomorphically maps to A also homomorphically maps to Γ, then A homomorphically maps to Γ.*
6. *There is a set $\mathcal{N}$ of finite structures of tree-width (l, k) such that every finite τ-structure A is homomorphic to Γ if and only if no structure in $\mathcal{N}$ is homomorphic to A.*

Proof. Suppose that an (l, k)-Datalog program Π solves CSP(Γ), and let S be an instance of CSP(Γ). If the canonical (l, k)-Datalog program derives false on S, then by Proposition 4 the structure S is not homomorphic to Γ. Otherwise, since Π is sound, the implication from 2 to 1 in Theorem 4 shows that the canonical (l, k)-Datalog program does not derive false on S as well. Hence, the canonical Datalog program solves CSP(Γ), and this is the proof of the implication from 1 to 2. The implications from 2 to 3, 3 to 4, 4 to 5, and 5 to 1 are straightfoward consequences of Theorem 4.

To show that 5 implies 6, let $\mathcal{N}$ be the set of all those structures of tree-width (l, k) that does not homomorphically map to Γ. Let A is a finite τ-structure. If A homomorphically maps to Γ, then clearly there is no structure C in $\mathcal{N}$ that maps to A, because then C would also map to Γ, a contradiction to the definition of $\mathcal{N}$. Conversely, suppose that no structure in $\mathcal{N}$ homomorphically maps to A. In other words, every structure that homomorphically maps to A also maps to Γ. Using 5, this implies that A homomorphically maps to Γ.

Finally, 6 implies 1. Let $\mathcal{N}$ be as in 6, let A be a finite τ-structure such that every finite τ-structure S of tree-width (l, k) that homomorphically maps to A also homomorphically maps to Γ. In particular, no structure in $\mathcal{N}$ homomorphically maps to A. Therefore, A homomorphically maps to Γ. □

5 1-Datalog, MMSNP, and Constraint Satisfaction

In this section we show that every class of structures with Datalog width one can be formulated as a constraint satisfaction problem with an ω-categorical template. A Datalog program of width one accepts a class of structures that can be described by a sentence of a fragment of existential second order logic called *monotone monadic SNP without inequalities (MMSNP)*. We show that every problem in MMSNP can be formulated as the constraint satisfaction problem for an ω-categorical template.

An *SNP sentence* is an existential second-order sentence with a universal first-order part. The first order part might contain the existentially quantified relation symbols and additional relation symbols from a given signature τ (the *input* relations). We shall assume that SNP formulas are written in *negation normal form*, i.e., the first-order part is written in conjunctive normal form, and each disjunction is written as a negated conjunction of positive and negative literals. The class SNP consists of all problems on τ-structures that can be described by an SNP sentence.

The class *MMSNP*, defined by Feder and Vardi, is the class of problems that can be described by an SNP sentence with the additional requirements that the existentially quantified relations are monadic, that every input relation symbol occurs negatively in the SNP sentence, and that it does not contain inequalities. Every problem in MMSNP is under randomized Turing reductions equivalent to a constraint satisfaction problem with a finite template [21]; a deterministic reduction was recently announced by Kun [33]. It is easy to see that MMSNP contains all constraint satisfaction problems with finite templates. Thus, MMSNP has a dichotomy if and only if CSP has a dichotomy.

It is easy to see that $(1, k)$-Datalog is contained in MMSNP: We introduce an existentially quantified unary predicate for each of the unary IDBs in the Datalog program. It is then straightforward to translate the rules of the Datalog program into first-order formulas with at most k first-order variables. We now want to prove that every problem in MMSNP can be formulated as a constraint satisfaction problem with a countably categorical template. In full generality, this cannot be true constraint satisfaction problems are always closed under disjoint union (a simple example of a MMSNP problem not closed under disjoint union is the one defined by the formula $\forall x, y\, \neg(P(x) \wedge Q(x)))$. Hence we shall assume that we are dealing with a problem in MMSNP that is *closed under disjoint union*.

To prove the claim under this assumption, we need a recent model-theoretic result of Cherlin, Shelah and Shi [12]. Let $\mathcal{N}$ be a finite set of finite structures with a relational signature τ. In this paper, a τ-structure Δ is called $\mathcal{N}$-*free* if there is no homomorphism from any structure in $\mathcal{N}$ to Δ. A structure Γ in a class of countable structures $\mathcal{C}$ is called *universal* for $\mathcal{C}$, if it contains all structures in $\mathcal{C}$ as an induced substructure.

Theorem 6 (of [12]). *Let $\mathcal{N}$ be a finite set of finite connected τ-structures. Then there is an ω-categorical universal structure Δ that is universal for the class of all countable $\mathcal{N}$-free structures.*

Cherlin, Shelah and Shi proved this statement for (undirected) graphs, but the proof does not rely on this assumption on the signature, and works for arbitrary relational signatures. The statement in its general form also follows from a result in [13]. We use the ω-categorical structure Δ to prove the following.

Theorem 7. *Every problem in MMSNP that is closed under disjoint unions can be formulated as CSP(Γ) with an ω-categorical template Γ.*

Proof. Let Φ be a MMSNP sentence with input signature τ whose set $\mathcal{M}$ of finite models is closed under disjoint unions. We have to find an ω-categorical τ-structure Γ, such that $\mathcal{M}$ equals $\mathrm{CSP}(\Gamma)$. Recall the assumption that Φ is written in negation normal form. Let $P_1, \ldots, P_k$ be the existential monadic predicates in Φ. By monotonicity, all such literals with input relations are positive. For each existential monadic relation P_i we introduce a relation symbol P_i', and replace negative literals of the form $\neg P_i(x)$ in Φ by $P_i'(x)$. We shall denote the formula obtained after this transformation by Φ'. Let τ' be the signature containing the input relations from τ, the existential monadic relations P_i, and the symbols P_i' for the negative occurrences of the existential relations. We define $\mathcal{N}$ to be the set of τ'-structures containing for each clause $\neg(L_1 \wedge \cdots \wedge L_m)$ in Φ' the canonical database [11] of $(L_1 \wedge \cdots \wedge L_m)$. We shall use the fact that a τ'-structure S satisfies a clause $\neg(L_1 \wedge \cdots \wedge L_m)$ if and only if the the canonical database of $(L_1 \wedge \cdots \wedge L_m)$ is not homomorphic to S.

We can assume without loss of generality that Φ is minimal in the sense that if we remove a literal from some of the clauses the formula obtained is inequivalent. We shall show that then all structures in $\mathcal{N}$ are connected. Let us suppose that this is not the case. Then there is a clause C in Φ that corresponds to a non connected structure in $\mathcal{N}$. The clause C can be written as $\neg(E \wedge F)$ where the set X of variables in E and the set Y of variables in F do not intersect. Consider the formulas Φ_E and Φ_F obtained from Φ by replacing C by $\neg E$ and C by $\neg F$, respectively. By minimality of Φ there is a structure M_E that satisfies Φ but not Φ_E, and similarly there exists a structure M_F that satisfies Φ but not Φ_F. By assumption, the disjoint union M of M_E and M_F satisfies Φ. Then there exists a τ''-expansion M'' of M where $\tau'' = \tau \cup \{P_1, \ldots, P_k\}$ that satisfies the first-order part of Φ. Consider the substructures M_E'' and M_F'' of M'' induced by the vertices of M_E and M_F. We have that M_E'' does not satisfy the first-order part of Φ_E (otherwise M_E would satisfy Φ_E). Consequently, there is an assignment s_E of the universal variables that falsifies some clause. This clause must necessarily be $\neg E$ (since otherwise M'' would not satisfy the first-order part of Φ). By similar reasoning we can infer that there is an assignment s_F of the universal variables of Φ to elements of M_F that falsifies $\neg F$. Finally, fix any assignment s that coincides with s_E over X and with s_F over Y (such an assignment exists because X and Y are disjoint). Clearly, s falsifies C and M does not satisfy Φ, a contradiction. Hence, we shall assume that every structure in $\mathcal{N}$ is connected.

Then Theorem 6 asserts the existence of a $\mathcal{N}$-free ω-categorical τ'-structure Δ that is universal for all $\mathcal{N}$-free structures. We use Δ to define the template Γ for the constraint satisfaction problem. To do this, restrict the domain of Δ to those points that have the property that either P_i or P_i' holds (but not both P_i and P_i') for all existential monadic predicates P_i. The resulting structure Δ' is non-empty, since the problem defined by Φ is non-empty. Then we take the reduct of Δ' that only contains the input relations from τ. It is well-known [28] that reducts and first-order restrictions of ω-categorical structures are again ω-categorical. Hence the resulting τ-structure Γ is ω-categorical.

We claim that an τ-structure S satisfies Φ if and only if $S \in \mathrm{CSP}(\Gamma)$. Let S be an structure that has a homomorphism h to Γ. Let S' be the τ'-expansion of S such that for each $i = 1, \ldots, k$ the relation $P_i(x)$ holds in S' if and only if $P_i(h(x))$ holds in Δ', and $P_i'(x)$ holds in S' if and only if $P_i'(h(x))$ holds in Δ'. Clearly, h defines a homomorphism from S' to Δ' and also from S to Δ. In consequence, none of the structures from $\mathcal{N}$ maps to S' (otherwise it would also map to Δ). Hence, the τ''-reduction of S' satisfies all the clauses of the first-order part of Φ and hence S satisfies Φ.

Conversely, let S be a structure satisfying Φ. Consequently, there exists a τ'-expansion S' of S that satisfies the first-order part of Φ' and where for every element x exactly one of $P_i(x)$ or $P_i'(x)$ holds. Clearly, no structure in $\mathcal{N}$ is homomorphic to the expanded structure, and by universality of Γ the τ'-structure S' is an induced substructure of Δ. Since for every point of S' exactly one of P_i an P_i' holds, S' is also an induced substructure of Δ'. Consequently, S is homomorphic to its τ-reduct Γ. This completes the proof. $\square$

In particular, we proved the following.

Theorem 8. *Every problem in $(1, k)$-Datalog that is closed under disjoint unions can be formulated as a constraint satisfaction problem with an ω-categorical template.*

For a typical example of a constraint satisfaction problem in MMSNP that cannot be described with a finite template [37] and that is not in (l, k)-Datalog for all $1 \leq l \leq k$, consider the following computational problem. Given is a finite graph S, and we want to test whether we can partition the vertices of S in two parts such that each part is triangle-free. The ω-categorical template that is used in the proof of Theorem 7 consists of two copies C_1 and C_2 of $\ntriangleleft$, where we add an undirected edge from all vertices in C_1 to all vertices in C_2. The corresponding constraint satisfaction problem is NP-hard [1, 25].

6 Bounded Width

We characterize the ω-categorical templates whose constraint satisfaction problems have bounded width. The results generalize algebraic characterizations of Datalog width that are known for constraint satisfaction with finite templates. However, not all results remain valid for infinite templates: It is well-known [21] that the constraint satisfaction of a finite template has Datalog width one if and only if the so-called *arc-consistency procedure* solves the problem. This is no longer true for infinite templates. We characterize both width one and the expressive power of the arc-consistency procedure for infinite ω-categorical templates, and present an example that shows that the two concepts are different. We also present an algebraic characterization of strict width l. Note that width one and strict width l are the only concepts of bounded Datalog width that are known to be decidable for finite templates.

6.1 Width zero

An example of a template whose constraint satisfaction problem has width 0 is the universal triangle-free graph $\ntriangleleft$. Since there is a primitive positive sentence that states the existence of a triangle in a graph, and since every graph without a triangle is homomorphic to $\ntriangleleft$, there is a Datalog program of width 0 that solves CSP($\ntriangleleft$). In general, it is easy to see that a constraint satisfaction problem has width 0 if and only if there is a finite set of *obstructions* for CSP(Γ), i.e., a finite set $\mathcal{N}$ of finite τ-structures such that every finite τ-structure A is homomorphic to Γ if and only if no substructure in $\mathcal{N}$ is homomorphic to A. Since the complement of this class is closed under homomorphisms, we can apply Rossman's theorem and obtain the result that a constraint satisfaction problem with an arbitrary infinite template has width 0 if and only if it is first-order definable [39]. For finite templates a characterization of first-order definable constraint satisfaction problems was obtained in [35] building on work in [2, 38]. Our discussion can summarized by the following theorem.

Theorem 9. *For every infinite (and not necessarily ω-categorical) template Γ the following is equivalent.*

- *CSP(Γ) is first-order definable;*
- *CSP(Γ) has a finite obstruction set;*
- *CSP(Γ) has Datalog width 0.*

Moreover, if CSP(Γ) is first-order definable we can always find an ω-categorical structure Γ' that has the same constraint satisfaction problem as Γ.

Proof. The equivalences have been discussed above and essentially follow from Rossman's theorem. The last remark is a special case of Theorem 8. □

6.2 Width one

Let Γ be an ω-categorical structure with relational signature τ, and Π be the canonical $(1,k)$-Datalog program for Γ. By Theorem 8, the class of τ-structures accepted by Π is itself a constraint satisfaction problem with an ω-categorical template. We denote this template by $\Gamma(1,k)$.

Theorem 10. *Let Γ be ω-categorical. A constraint satisfaction problem CSP(Γ) can be solved by an $(1,k)$-Datalog program if and only if there is a homomorphism from $\Gamma(1,k)$ to Γ.*

Proof. The constraint satisfaction problem for Γ has width $(1,k)$ if and only if the canonical $(1,k)$-Datalog program Π solves CSP(Γ), by Theorem 5. Since CSP(Γ) is closed under disjoint unions, we can apply Theorem 8 and know that CSP(Γ) equals CSP$(\Gamma(1,k))$. Therefore, Lemma 2 implies that CSP(Γ) has width $(1,k)$ if and only if there is a homomorphism from $\Gamma(1,k)$ to Γ. $\square$

6.3 Arc-consistency

The *arc-consistency procedure (AC)* is an algorithm for constraint satisfaction problems that is intensively studied in Artificial Intelligence. It can be described as the subset of the canonical Datalog program of width one that consists of all rules with bodies containing at most one non-IDB. An instance that is stable under inferences of this Datalog program is called *arc-consistent.* For finite templates T it is known that the arc-consistency procedure solves CSP(T) if and only if CSP(T) has width one [21]. For infinite structures, this is no longer true: consider for instance CSP$(\not\triangleleft)$, which has width 0, but cannot be solved by the arc-consistency procedure. The reason is that the width one canonical Datalog program for $\not\triangleleft$ has no non-trivial unary predicates, and we thus have to consider at least three relations in the input to infer that the input contains triangle.

Theorem 11. *Let Γ be an ω-categorical relational structure. If CSP(Γ) is solved by the arc-consistency algorithm, then Γ is homomorphically equivalent to a finite structure.*

Proof. Since Γ is ω-categorical, the automorphism group of Γ has a finite number of orbits (i.e., orbits of 1-subsets) $O_1,\dots,O_n$. We define the *orbit structure* of Γ, which is a finite relational τ-structure whose vertices $S_1,\dots,S_{2^n-1}$ are the nonempty subsets of $\{O_1,\dots,O_n\}$, and where a k-ary relation R from τ holds on $S_{i_1},\dots,S_{i_k}$ if for every vertex v_j in an orbit from S_{i_j} there are vertices $v_1,\dots,v_{j-1},v_{j+1},\dots,v_k$ from $S_{i_1},\dots,S_{i_{j-1}},S_{i_{j+1}},\dots,S_{i_k}$, respectively, such that R holds on $v_1,\dots,v_k$ in Γ. Every unary relation that can be inferred by the arc-consistency procedure corresponds to a list of orbits of Γ, because Γ is ω-categorical. Since every rule application of the procedure involves a single input relation, the definition of the orbit structure implies that the Datalog program cannot infer any new relations on the orbit structure, and therefore the orbit structure is arc-consistent. Since the arc-consistency procedure solves the constraint satisfaction problem for Γ, the orbit structure is homomorphic to Γ.

Next we show that there is also a homomorphism from Γ to the orbit structure. Every finite substructure S of Γ is a satisfiable instance of CSP(Γ), and hence the arc-consistency procedure does not derive false on it. Consider the arc-consistent instance computed by the arc-consistency procedure on S. For each variable v we have computed a set of unary predicates, that corresponds to a non-empty subset of orbits, and that tells us to which element of the orbit structure we can map v. By definition of the orbit structure, this mapping is a homomorphism, because if there is a constraint that is not supported in the orbit structure, then the arc-consistency algorithm would have removed at least one of the orbits in the orbit list for the involved variable. We have thus shown that all finite substructures S of Γ homomorphically map to the orbit structure. Since Γ is countable and the orbit structure is finite, we conclude with Lemma 2 that there is a homomorphism from Γ to the orbit structure. $\square$

7 Bounded Strict Width

The notion of strict width was introduced for finite domain constraint satisfaction problems by Feder and Vardi [21], and was defined in terms of the canonical Datalog program. In the terminology of the constraint satisfaction literature in Artificial Intelligence, strict width l is equivalent to *'strong l-consistency implies global consistency'*. Based on our generalization of the concept of canonical Datalog programs, we study the analogously defined concept of strict width l for ω-categorical structures.

The notion of strict width is defined as follows. Recall that the canonical (l,k)-Datalog Program Π for CSP(Γ) receives as input an instace S of CSP(Γ) and returns an expansion S' of S over τ' where τ' is the vocabulary that contains all predicates τ as well as a predicate for every IDB of Π. The structure S' can be seen as an instance of CSP(Γ') where Γ' is the expansion of Γ in which every IDB R is interpreted as the at most l-ary primitive positive relation associated to it. The instance S' is called *globally consistent*, if every partial homomomorphism, i.e, every homomorphism from an induced substructure of S' to Γ, can be extended to homomorphism from S to Γ. If for some $k \geq l+1 \geq 3$ all instances of CSP(Γ) that are computed by the canonical (l,k)-program are globally consistent, we say that Γ has *strict width l*. Note that strict width l implies width l, and hence CSP(Γ) can be solved in polynomial time when Γ has bounded strict width.

In this section we present an algebraic characterization of strict width l for ω-categorical templates Γ. The algebraic approach rests on the notion of *polymorphisms*. Let Γ be a relational structure with signature τ. A *polymorphism* is a homomorphism from Γ^l to Γ, for some l, where Γ^l is a relational τ-structure defined as follows. The vertices of Γ^l are l-tuples over elements from V_Γ, and k such l-tuples $(v_1^i,\dots,v_l^i)$, $1 \leq i \leq l$, are joined by a k-ary relation R from τ if $(v_j^1,\dots,v_j^k)$ is in R^Γ, for all $1 \leq j \leq l$.

We say that an operation f is a *near-unanimity operation* (short, *nu-operation*) if it satisfies the identities $f(x,\dots,x,y,x,\dots,x) = x$, i.e., in the case that the arguments have the same value x except at one argument position, the operation has the value x. We say that f is a *near-unanimity operation on A* if it satisfies the identities $f(x,\dots,x,y,x,\dots,x) = x$ for all $x,y \in A$.

Feder and Vardi [21] proved that a finite template Γ has a $l+1$-ary near-unanimity operation (in this case, they say that Γ has the $(l+1)$*-mapping property*) if and only if CSP(Γ) has strict width l. Another proof of this theorem was given in [29]. It is stated there that the proof extends to arbitrary infinite templates, if we want to characterize bounded strict width on instances of the constraint satisfaction problem that might be infinite. However, we would like to describe the complexity of constraint satisfaction problems with finite instances.

In fact, there are structures that do not have a nu-operation, but have for all finite subsets A of Γ a polymorphism that is idempotent on A. One example for such a structure is the universal triangle-free graph $\ntriangleleft$. A theorem by Larose and Tardif shows that every finite or infinite graph with a nu-operation is bipartite [36]. Since the universal triangle-free graph contains all cycles of length larger than three, it therefore cannot have a nu-operation. However, the universal triangle-free graph has strict width 2. Indeed, for any instance S accepted by the canonical $(2,3)$-Datalog program, every partial mapping from S to $\ntriangleleft$ satisfying all the facts derived by the program (and in particular not containing any triangle) can be extended to a complete homomorphisms from S to $\ntriangleleft$ – this follows from the extension properties of the template.

Theorem 12 characterizes strict width l, $l \geq 2$, for constraint satisfaction with ω-categorical templates. We first need an intermediate result.

Lemma 3. *Let Γ be a τ-structure such that CSP(Γ) has strict width l and let $\tau_\equiv$ be the superset of τ in which we add a new binary relation symbol $P_\equiv$. Let $\Gamma_\equiv$ be the $\tau_\equiv$-expansion of Γ in which $P_\equiv$ is interpreted by the usual equality relation $\{(x,x) \mid x \in D_\Gamma\}$. Then CSP($\Gamma'$) has also strict width l.*

Proof. Let $\Pi_\equiv$ be the canonical (l,k)-program for CSP($\Gamma_\equiv$), and $S_\equiv$ be an instance of CSP($\Gamma_\equiv$). Let $S'_\equiv$ be the structure computed by $\Pi_\equiv$ on $S_\equiv$. Let θ be the binary relation on the universe of $S_\equiv$ defined to be the reflexive and transitive closure of $P_\equiv^{S_\equiv}$.

Let S_θ be the τ-reduct of $S_\equiv$, in which we identify all vertices that are θ-related into a single element. More precisely, the universe of S_θ are the classes of θ, $\{\theta_a \mid a \in D_{S_\equiv}\}$, where θ_a denotes the θ-class of a, and for every $R \in \tau$, say r-ary, $R^{S_\theta} = \{(\theta_{a_1}, \ldots, \theta_{a_r}) \mid (a_1, \ldots, a_r) \in R^{S_\equiv}\}$. We now consider S_θ as a CSP(Γ) instance. Let Π be the canonical (l,k)-program of Γ. It is easy to prove by induction on the derivation order that if R is an IDB, say r-ary, and $R(\theta_{a_1}, \ldots, \theta_{a_r})$ is derived by Π on S_θ, then $R(a_1, \ldots, a_r)$ is derived by $\Pi_\equiv$ on $S_\equiv$.

We have to show that $S'_\equiv$ is globally consistent. So suppose that there is a partial homomorphism h from $S'_\equiv$ to $\Gamma_\equiv$. Since $l \geq 2$ and $k \geq 3$, $\Pi_\equiv$ will be able to derive that all elements in the same θ-class have to get the same value and hence if h is a partial homomorphism this implies that for all elements a, b in the domain of h that are θ-related, $h(a) = h(b)$. Define h_θ to be the partial mapping that maps every θ_a with a in the domain of h to $h(a)$. By the definition of $S_\equiv$ and analysis on the facts derived by Π on $S_\equiv$ carried out below we know that h_θ is a partial homomorphism from S_θ to Γ. Hence h can be extended to a full homomomorphism h' from S_θ to Γ. Finally, the mapping h' defined to be $h'(a) = h'_\theta(\theta_a)$ is a homomorphism from $S_\equiv$ to Γ and hence also from $S'_\equiv$ to Γ'. ☐

The proof of the following theorem is based on ideas from [21] and [29].

Theorem 12. *Let Γ be an ω-categorical structure with relational signature τ of bounded maximal arity. Then the following are equivalent, for $l \geq 2$:*

1. *CSP(Γ) has strict width l.*
2. *For every finite subset A of Γ there is an $l+1$-ary polymorphism of Γ that is a nuf on A.*

Proof. We first show that (1) implies (2).

We assume that CSP(Γ) has strict width l, and prove that for every finite subset A of Γ there is a polymorphism of Γ that is a $l+1$-ary nuf on A. Let τ^A be the superset of τ in which we add a new unary relation symbol R_a for each element a in A, let Γ^A be the τ^A-expansion of Γ in which R_a in interpreted by the singleton relation $\{a\}$, let Δ be $(\Gamma)^{l+1}$. Consider the set B of tuples $(a_0, \ldots, a_l)$ in A^l that have identical entries $a_i = a$ except possibly at one exceptional position. Let Δ^A be the τ^A-expansion of Δ obtained by placing in R_a all tuples $(a, \ldots, a, b, a, \ldots, a)$ in B whose majority element is a. Every homomorphism from Δ^A to Γ^A is by construction a polymorphism of Γ that is a nuf on A. Lemma 2 shows that if every finite substructure S^A of Δ^A homomorphically maps to Γ^A, then Δ^A homomorphically maps to Γ^A as well.

Let S^A be any finite substructure of Δ^A, and let S be the τ-reduct of S^A which we see as an instance of CSP(Γ). We shall show that there exists an h homomorphism from S to Γ that sends every element $(a, \ldots, a, b, a, \ldots, a)$ in $B \cap D_S$ to its majority element a. Hence h will also define a homomorphism from S^A to Γ^A.

Let $\tau_\equiv$ be the superset of τ in which we add a new binary predicate $P_\equiv$ and let $\Gamma_\equiv$ be the expansion of Γ in which $P_\equiv$ is interpreted as the equality relation. Let $S_\equiv$ be the following $\tau_\equiv$ structure: The universe of $S_\equiv$ is $D_S \times \{0,1\}$. For every predicate $P \in \tau$, say r-ary, we define $P^{S_\equiv}$ to be $P^S \times \{(0, \ldots, 0)\}$. Furthermore

$$P_\equiv^{S_\equiv} = \{((a,0),(a,1)) \mid a \in S\}.$$

By Lemma 3, $\Gamma_\equiv$ has strict width l. Hence, there exists some k such that all instances of CSP($\Gamma_\equiv$) computed by the canonical (l,k)-program Π are globaly consistent. Let $S'_\equiv$ be the instance computed by Π on $S_\equiv$ (which we recall has an expanded vocabulary $\tau'_\equiv$). Now consider the partial assigment g defined on $(B \cap D_S) \times \{1\}$ that sends every tuple of the form $((a,1), \ldots, (a,1), (b,1), (a,1), \ldots, (a,1))$ to a. We shall see that g is a partial homomorphism from $S'_\equiv$ to $\Gamma'_\equiv$: Indeed, let $(\overline{a}_1, \ldots, \overline{a}_r) \in R^{S'_\equiv}$ be any tuple entirely contained in the domain of g. For every $1 \leq j \leq r$, $\overline{a}_j$ is of the from $((a_j,1), \ldots, (a_j,1), (b_j,1), (a_j,1), \ldots, (a_j,1))$. This tuple has necessarily been placed there by the Datalog program, and hence R is an IDB and has cardinaly at most l. The pigeon-hole principle guarantees that there exists an index index i, $1 \leq i \leq l+1$ such that for every $1 \leq j \leq r$, the ith entry of $\overline{a}_j$ is precisely $(a_j,1)$. Since the ith projection is a homomorphism from S to Γ, it cannot violate any fact derived by the canonical (l,k)-Program and hence $(a_1, \ldots, a_l) \in R^{\Gamma'_{\equiv}}$. Since $S'_{\equiv}$ is globally consistent this implies that g can be extended to a full homomomorphism g' from $S'_{\equiv}$ to $\Gamma'_{\equiv}$. Finally we obtain the desired homomorphism $h : S \to D_\Gamma$ as $h(a_1, \ldots, a_l) = g'((a_1, 0), \ldots, (a_l, 0))$.

Next we show that (2) implies (1). Let S be an instance of CSP(Γ) such that the canonical (l, k)-program Π does not derive false on S, where k is larger than the maximal arity of the relations in τ, and at least $l + 1$. Let Γ' be the expansion of Γ by all at most l-ary primitive positive definable relations, and let S' be the instance of CSP(Γ') computed by Π on the instance S. We shall prove that we can extend every homomorphism h from an induced substructure K of S' to another element of S'. We prove this by induction on the number i of elements in K. Let $v_1, \ldots, v_i$ be the elements of K, and let v_{i+1} be an element of S' that is not in the domain of h.

For the case that $i \leq l$, let $R_j(\overline{v}_j), j \in J$, be the set of all predicates in S' containing only elements from $\{v_1, \ldots, v_i, v_{i+1}\}$, and let R be the IDB associated to the formula $\exists v_{i+1} \bigwedge_{j \in J} R_j(\overline{v}_j)$ with free variables $x_1, \ldots, x_i$ (here we view $v_1, \ldots, v_{i+1}$ as variables of the formula). Since $R_j(\overline{v}_j), j \in J$ are derived by Π, the predicate $R(v_1, \ldots, v_i)$ is also derived. Since h preserves this predicate, there exists some way to extend h such that it also preserves $R_j(\overline{v}_j)$ for all $j \in J$.

For the inductive case where $i \geq l+1$, select elements $w_1, \ldots, w_{l+1}$ in $\{v_1, \ldots, v_i\}$, and let h_j be the restriction of h in which w_j is undefined, for $j \in \{1, \ldots, l\}$. By induction, each of the mappings h_j can be extended to a homomorphism h'_j from the structure induced by $v_1, \ldots, v_{i+1}$ in S to Γ. Let A be a set containing the images of all functions h'_j, and let g be a $(l + 1)$-ary polymorphism of Γ that is a nuf on A. Define b to be $g(h'_1(v_{i+1}), \ldots, h'_{l+1}(v_{i+1}))$. We claim that the extension h' of h mapping v_{i+1} to b is a homomorphism from the structure induced by $\{v_1, \ldots, v_{i+1}\}$ in S'.

First we shall see that h' preserves all IDBs. Let $R(u_1, \ldots, u_r)$, $r \leq l$, be any predicate derived by Π with $u_1, \ldots, u_r \in \{v_1, \ldots, v_{i+1}\}$. Define a_j to be $h'(u_j)$ so that we have to prove that $(a_1, \ldots, a_r)$ belongs to $R^{\Gamma'}$. For each $j = 1, \ldots, l + 1$ we construct an r-tuple $b^j = (b^j_1, \ldots, b^j_r)$ in $R^{\Gamma'}$ in the following way: if v_j is not in $\{u_1, \ldots, u_r\}$, then we set b^j_t to $h_i(v_t)$ for $t \in \{1, \ldots, r\}$. Otherwise, we consider the restriction of h'_j to $\{u_1, \ldots, u_l\} \setminus \{v_j\}$. By induction hypothesis this restricted mapping can be extended to a mapping h''_j defined for all $\{u_1, \ldots, u_l\}$. We define b^j_t to be $h''_j(u_t)$ for all $t \in \{1, \ldots, r\}$. We claim that the tuple $(f(b^1_1, \ldots, b^{l+1}_1), \ldots, f(b^1_r, \ldots, b^{l+1}_r))$ is indeed $(a_1, \ldots, a_l)$. The reason is that at least l of the elements $b^1_t, \ldots, b^{l+1}_t$ are equal to a_t. Because g is a near-unanimity operation on A, we have $g(b^1_t, \ldots, b^l_t) = a_l$. Hence, by the definition of polymorphisms, this tuple belongs to $R^{\Gamma'}$.

It remains to show that h' preserves all EDBs. Let $R(u_1, \ldots, u_r)$ be any tuple initially in S. Let us denote $h'(u_j)$ by a_j so that we have to prove that $(a_1, \ldots, a_r) \in R^\Gamma$. We shall show that for every $I \subseteq \{1, \ldots, r\}$ there is a tuple $(b_1, \ldots, b_r)$ in R^Γ such that $b_i = a_i$ for all $i \in I$, by induction on $|I|$.

For the case $|I| \leq l$, let $i_1 < \cdots < i_{l'}$ be the elements of I, let R' be the IDB associated to the formula $\exists_{i \notin I} v_i R(v_1, \ldots, v_r)$ with free variables $v_{i_1}, \ldots, v_{i'_l}$. Since $R'(v_{i_1}, \ldots, v_{i_{l'}}) {:}{-} R(v_1, \ldots, v_r)$ is a rule of Π and, as we have seen that $(a_{i_1}, \ldots, a_{i_{l'}}) \in {R'}^{\Gamma'}$, the claim follows. For the inductive case $|I| > l$, select $l + 1$ elements $i_1, \ldots, i_{l+1}$ of I and consider the tuple $b^j = (b^1_1, \ldots, b^j_r)$ given inductively for $I \setminus \{i_j\}$. Let g be an $l + 1$-ary polymorphism which is a nuf on the set containing all elements in all tuples b_j for $j = 1, \ldots, r$. The tuple $(g(b^1_1, \ldots, b^1_{l+1}), \ldots, g(b^r_1, \ldots, b^r_{l+1}))$ satisfies the claim. $\square$

Note that in several papers including [4, 5] and the conference version that precedes this one, condition (2) has been stated in a different but essentially equivalent way using the notion of quasi near-unanimity operation.[3]

We say that an operation f is a *quasi near-unanimity operation* (short, *qnu-operation*), if it satisfies the identities $f(x, \ldots, x, y, x, \ldots, x) = f(x, \ldots, x)$, i.e., in the case that the arguments have the same value x except at one argument position, the operation has the value $f(x, \ldots, x)$. In

[3] In the conference version of this paper, these operations were called *weak near-unanimity operations*. However, since another similar but much weaker relaxation of near-uanimity operations was introduced recently in universal algebra as well, we decide to call our operations *quasi* near-unanimity operations.

other words, the value y of the exceptional argument does not influence the value of the operation f. Several well-known temporal and spatial constraint languages have polymorphisms that are qnu-operations [5].

For every subset A of Γ, we say that an operation is *idempotent on A* if $f(a, \ldots, a) = a$ for all $a \in A$. Hence, if a qnu-operation f is idempotent on the entire domain, then f is a near-unanimity operation. If a polymorphism f of Γ has the property that for every finite subset A of Γ there is an automorphism α of Γ such that $f(x, \ldots, x)) = \alpha(x)$ for all $x \in A$, we say that f is *oligopotent*.

Theorem 13. *Let Γ be a ω-categorical structure with relational signature τ and let $l \geq 2$. Then the following are equivalent:*

1. *CSP(Γ) has strict width l.*
2. *For every finite subset A of Γ there is an $l+1$-ary polymorphism of Γ that is a nuf on A.*
3. *Γ has an oligopotent $l+1$-ary polymorphism that is a qnu-operation.*
4. *Every primitive positive formula is in Γ equivalent to a conjunction of at most l-ary primitive positive formulas.*

Proof. The equivalence of (1) and (2) has been shown in Theorem 12, and the equivalence of (2) and (3) follows from a direct application of Lemma 2. The equivalence of (3) and (4) is shown in [4]. □

Concerning the condition of oligopotency in statement (3) of Theorem 13, we want to remark that for every ω-categorical structure Γ there is a template that has the same CSP and where all polymorphisms are oligopotent. It was shown in [25] that every ω-categorical structure is homomorphically equivalent to a *model-complete core* Δ, i.e., Δ has the property that for every finite subset A of the domain of Δ and for every *endomorphism* e of Δ (an endomorphism is a unary polymorphism) there exists an automorphism a of Δ such that $a(x) = e(x)$ for all $x \in A$. (Moreover, it is also known that Δ is unique up to isomorphism, and ω-categorical.)

Corollary 2. *Suppose that Δ is an ω-categorical model-complete core. Then Δ has strict width l if and only if Δ has an $l+1$-ary qnu-polymorphism.*

References

1. D. Achlioptas. The complexity of G-free colourability. *Discrete Mathematics*, 165:21–30, 1997.
2. A. Atserias. On digraph coloring problems and treewidth duality. In *20th IEEE Symposium on Logic in Computer Science (LICS)*, pages 106–115, 2005.
3. T. Biedl. *Graph-Theoretic Algorithms.* Lecture notes, 2005.
4. M. Bodirsky and H. Chen. Oligomorphic clones. *Algebra Universalis*, 57(1):109–125, 2007.
5. M. Bodirsky and H. Chen. Qualitative temporal and spatial reasoning revisited. In *CSL'07*, pages 194–207, 2007.
6. M. Bodirsky and V. Dalmau. Datalog and constraint satisfaction with infinite templates. In *Proceedings of STACS'06*, pages 646–659, 2006.
7. M. Bodirsky and J. Nešetřil. Constraint satisfaction with countable homogeneous templates. *Journal of Logic and Computation*, 16(3):359–373, 2006.
8. A. Bulatov, P. Jeavons, and A. Krokhin. The complexity of constraint satisfaction: An algebraic approach (a survey paper). *In: Structural Theory of Automata, Semigroups and Universal Algebra (Montreal, 2003), NATO Science Series II: Mathematics, Physics, Chemistry*, 207:181–213, 2005.
9. A. Bulatov, A. Krokhin, and P. G. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.
10. P. J. Cameron. *Oligomorphic Permutation Groups.* Cambridge Univ. Press, 1990.
11. A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceddings of STOC'77*, pages 77–90, 1977.
12. G. Cherlin, S. Shelah, and N. Shi. Universal graphs with forbidden subgraphs and algebraic closure. *Advances in Applied Mathematics*, 22:454–491, 1999.
13. J. Covington. Homogenizable relational structures. *Illinois Journal of Mathematics*, 34(4):731–743, 1990.

14. M. Cristiani and R. Hirsch. The complexity of the constraint satisfaction problem for small relation algebras. *Artificial Intelligence Journal*, 156:177–196, 2004.
15. V. Dalmau. A new tractable class of constraint satisfaction problems. Presented at the 6th International Symposium on Mathematics and Artificial Intelligence, 2000.
16. V. Dalmau, P. G. Kolaitis, and M. Y. Vardi. Constraint satisfaction, bounded treewidth, and finite-variable logics. In *Proceedings of CP'02*, pages 310–326, 2002.
17. V. Dalmau, A. A. Krokhin, and B. Larose. First-order definable retraction problems for posets and reflexive graph. In *LICS'04*, pages 232–241, 2004.
18. V. Dalmau and J. Pearson. Closure functions and width 1 problems. *Proceedings of CP'99*, pages 159–173, 1999.
19. I. Düntsch. Relation algebras and their application in temporal and spatial reasoning. *Artificial Intelligence Review*, 23:315–357, 2005.
20. H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer, 1999. 2nd edition.
21. T. Feder and M. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1999.
22. T. Feder and M. Vardi. Homomorphism closed vs. existential positive. In *Symposium on Logic in Computer Science (LICS'03)*, pages 311–320, 2003.
23. E. Grädel, W. Thomas, and T. Wilke. *Automata, Logics, and Infinite Games. A Guide to Current Research.* LNCS 2500. Springer-Verlag, 2002.
24. M. Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1), 2007.
25. P. Hell and J. Nesetril. The core of a graph. *Discrete Math.*, 109:117–126, 1992.
26. P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48:92–110, 1990.
27. R. Hirsch. Expressive power and complexity in algebraic logic. *Journal of Logic and Computation*, 7(3):309 – 351, 1997.
28. W. Hodges. *A shorter model theory.* Cambridge University Press, 1997.
29. P. Jeavons, D. Cohen, and M. Cooper. Constraints, consistency and closure. *AI*, 101(1-2):251–265, 1998.
30. P. Jeavons, D. Cohen, and M. Gyssens. Closure properties of constraints. *JACM*, 44(4):527–548, 1997.
31. P. G. Kolaitis and M. Y. Vardi. On the expressive power of Datalog: Tools and a case study. *Journal of Computer and System Sciences*, 51(1):110–134, 1995.
32. P. G. Kolaitis and M. Y. Vardi. Conjunctive-query containment and constraint satisfaction. In *Proceedings of PODS'98*, pages 205–213, 1998.
33. G. Kun. Every problem in MMSNP is polynomial time equivalent to a CSP. Personal communication, 2005.
34. P. B. Ladkin and R. D. Maddux. On binary constraint problems. *Journal of the Association for Computing Machinery*, 41(3):435–469, 1994.
35. B. Larose, C. Loten, and C. Tardif. A characterisation of first-order constraint satisfaction problems. *Logical Methods in Computer Science, DOI: 10.2168/LMCS-3(4:6)*, 2007.
36. B. Larose and C. Tardif. Strongly rigid graphs and projectivity. *Multiple-Valued Logic 7*, pages 339–361, 2001.
37. F. R. Madelaine and I. A. Stewart. Some problems not definable using structure homomorphisms. *Ars Combin.*, 67:153–159, 2003.
38. J. Nešetřil and C. Tardif. Duality theorems for finite structures. *Journal of Combininatorial Theory Series B*, 80:80–97, 2000.
39. B. Rossman. Existential positive types and preservation under homomorphisms. In *20th Annual IEEE Symposium on Logic in Computer Science (LICS' 05)*, pages 467–476, 2005.