

Datalog-Expressibility for Monadic and Guarded Second-Order Logic

Manuel Bodirsky^{1*}, Simon Knäuer^{1**}, and Sebastian Rudolph^{2***}

¹ TU Dresden, Institut für Algebra, Germany

² TU Dresden, Computational Logic Group, Germany

Abstract. We characterise the sentences in Monadic Second-order Logic (MSO) that are over finite structures equivalent to a Datalog program, in terms of an existential pebble game. We also show that for every class \mathcal{C} of finite structures that can be expressed in MSO and is closed under homomorphisms, and for all $\ell, k \in \mathbb{N}$, there exists a *canonical* Datalog program Π of width (ℓ, k) , that is, a Datalog program of width (ℓ, k) which is sound for \mathcal{C} (i.e., Π only derives the goal predicate on a finite structure \mathfrak{A} if $\mathfrak{A} \in \mathcal{C}$) and with the property that Π derives the goal predicate whenever *some* Datalog program of width (ℓ, k) which is sound for \mathcal{C} derives the goal predicate. The same characterisations also hold for Guarded Second-order Logic (GSO), which properly extends MSO. To prove our results, we show that every class \mathcal{C} in GSO whose complement is closed under homomorphisms is a finite union of constraint satisfaction problems (CSPs) of ω -categorical structures.

1 Introduction

Monadic Second-order Logic (MSO) is an important logic in theoretical computer science. By Büchi's theorem, a formal language can be defined in MSO if and only if it is regular (see, e.g., [23]). MSO sentences can be evaluated in polynomial time on classes of structures whose treewidth is bounded by a constant; this is known as Courcelle's theorem [15]. The latter result even holds for the more expressive logic of *Guarded Second-order Logic (GSO)* [17, 20], which extends First-order Logic by second-order quantifiers over *guarded relations*. Guarded Second-order Logic contains *Guarded First-order Logic* (which itself captures many description logics [19]).

Another fundamental formalism in theoretical computer science, which is heavily studied in database theory, is *Datalog* (see, e.g., [23]). Every Datalog program can be evaluated on finite structures in polynomial time. Like MSO, Datalog strikes a good balance between expressivity and good mathematical and computational properties. Two important parameters of a Datalog program Π are the maximal arity ℓ of its auxiliary predicates (IDBs), and the maximal number k of variables per rule in Π . We then say that Π has *width* (ℓ, k) , following the terminology of Feder and Vardi [18]. These parameters are important both in theory and in practice: ℓ closely corresponds to memory space and k to computation time needed when evaluating Π on a given structure.

In some scenarios we are interested in having the good computational properties of expressibility in Datalog *and* having the good computational properties of expressibility in MSO. And indeed, several important query formalisms, such as monadic Datalog, monadically defined queries, or nested monadically defined queries are contained both in MSO and in Datalog [24]. In this paper we are interested in two questions that (perhaps surprisingly) turn out to be closely related:

1. Which classes of finite structures are simultaneously expressible in MSO and in Datalog?
2. Which *constraint satisfaction problems (CSPs)* can be expressed in MSO, or, more generally, in GSO?

* The author has received funding from the European Research Council via the ERC Consolidator Grant No. 681988 (CSP-Infinity).

** The author is supported by the DFG Research Training Group 1763 (QuantLA).

*** The author has received funding from the European Research Council via the ERC Consolidator Grant No. 771779 (DeciGUT).

For a structure \mathfrak{B} with a finite relational signature τ , the *constraint satisfaction problem for \mathfrak{B}* is the class of all finite τ -structures that homomorphically map to \mathfrak{B} . Every finite-domain constraint satisfaction problem can already be expressed in monotone monadic SNP (MMSNP; [18]), which is a small fragment of MSO. On the other hand, the constraint satisfaction problem for $(\mathbb{Q}; <)$, which is the class of all finite acyclic digraphs $(V; E)$, cannot be expressed in MMSNP [5], but can be expressed in MSO by the sentence

$$\forall X \neq \emptyset \exists x \in X \forall y \in X: \neg E(x, y).$$

The class of CSPs of arbitrary infinite structures \mathfrak{B} is quite large; it is easy to see that a class \mathcal{D} of finite structures with a finite relational signature τ is a CSP of a countably infinite structure if and only if

- it is closed under disjoint unions, and
- $\mathfrak{A} \in \mathcal{D}$ for any \mathfrak{A} that maps homomorphically to some $\mathfrak{A}' \in \mathcal{D}$.

The second item can equivalently be rephrased as the *complement* of \mathcal{D} (meant within the class of all finite τ -structures; this comment applies throughout and will be omitted in the following) being *closed under homomorphisms*: a class \mathcal{C} is closed under homomorphisms if for any structure $\mathfrak{A} \in \mathcal{C}$ that maps homomorphically to some \mathfrak{C} we have $\mathfrak{C} \in \mathcal{C}$. Examples of classes of structures that are closed under homomorphisms naturally arise from Datalog. We say that a class \mathcal{C} of finite τ -structures is *in Datalog*³ if there exists a Datalog program Π with a distinguished predicate nullary goal such that Π derives goal on a finite τ -structure if and only if the structure is in \mathcal{C} ; in this case, we write $\llbracket \Pi \rrbracket$ for \mathcal{C} . Every class of τ -structures in Datalog is closed under homomorphisms. However, not every class of finite structures in Datalog describes the complement of a CSP: consider for example, for unary predicates R and B , the class $\mathcal{C}_{R,B}$ of finite $\{R, B\}$ -structures \mathfrak{A} such that $R^{\mathfrak{A}}$ is empty or $B^{\mathfrak{A}}$ is empty. Clearly, $\mathcal{C}_{R,B}$ is not closed under disjoint unions. However, a finite structure is in $\mathcal{C}_{R,B}$ if and only if the Datalog program that consists of just one rule

$$\text{goal} :- R(x), B(y)$$

does not derive goal on that structure.

An important class of CSPs is the class of CSPs for structures \mathfrak{B} that are countably infinite and ω -categorical. A structure \mathfrak{B} is ω -categorical if all countable models of the first-order theory of \mathfrak{B} are isomorphic. A well-known example of an ω -categorical structure is $(\mathbb{Q}; <)$, which is a result due to Cantor. Constraint satisfaction problems of ω -categorical structures can be evaluated in polynomial time on classes of treewidth bounded by some constant $k \in \mathbb{N}$, by a result of Bodirsky and Dalmau [6]. The polynomial-time algorithm presented by Bodirsky and Dalmau is in fact a Datalog program of width $(k - 1, k)$. A Datalog program Π is called *sound* for a class of τ -structures \mathcal{C} if $\llbracket \Pi \rrbracket \subseteq \mathcal{C}$. Bodirsky and Dalmau showed that if \mathcal{C} is the complement of the CSP of an ω -categorical τ -structure \mathfrak{B} then there exists for all $\ell, k \in \mathbb{N}$ a *canonical Datalog program of width (ℓ, k) for \mathcal{C}* , i.e., a Datalog program Π of width (ℓ, k) such that

- Π is sound for \mathcal{C} , and
- $\llbracket \Pi' \rrbracket \subseteq \llbracket \Pi \rrbracket$ for every Datalog program Π' of width (ℓ, k) which is sound for \mathcal{C} .

Moreover, whether the canonical Datalog program of width (ℓ, k) for \mathcal{C} derives goal on a given τ -structure \mathfrak{A} can be characterised in terms of the existential pebble game from finite model theory, played on $(\mathfrak{A}, \mathfrak{B})$ [6]. The *existential ℓ, k pebble game* is played by two players, called *Spoiler* and *Duplicator* (see, e.g., [16, 18, 22]). Spoiler starts by placing k pebbles on elements a_1, \dots, a_k of \mathfrak{A} , and Duplicator responds by placing k pebbles b_1, \dots, b_k on \mathfrak{B} . If the map that sends a_1, \dots, a_k to b_1, \dots, b_k is not a partial homomorphism from \mathfrak{A} to \mathfrak{B} , then the game is over and Spoiler wins. Otherwise, Spoiler removes all but at most ℓ pebbles from \mathfrak{A} , and Duplicator has to respond by removing the corresponding pebbles from \mathfrak{B} . Then Spoiler can again place all his pebbles on \mathfrak{A} ,

³ Warning: Feder and Vardi [18] say that a CSP is in Datalog if its *complement* in the class of all finite τ -structures is in Datalog.

and Duplicator must again respond by placing her pebbles on \mathfrak{B} . If the game continues forever, then Duplicator wins. If \mathfrak{B} is a finite, or more generally a countable ω -categorical structure then Spoiler has a winning strategy for the existential ℓ, k pebble game on $(\mathfrak{A}, \mathfrak{B})$ if and only if the canonical Datalog program for $\text{CSP}(\mathfrak{B})$ derives goal on \mathfrak{A} (Theorem 21). This connection played an essential role in proving Datalog inexpressibility results, for example for the class of finite-domain CSPs [2] (leading to a complete classification of those finite structures \mathfrak{B} such that the complement of $\text{CSP}(\mathfrak{B})$ can be expressed in Datalog [3]).

Results and Consequences

We present a characterisation of those GSO sentences Φ that are over finite structures equivalent to a Datalog program. Our characterisation involves a variant of the existential pebble game from finite model theory, which we call the (ℓ, k) -game. This game is defined for a homomorphism-closed class \mathcal{C} of finite τ -structures, and it is played by the two players Spoiler and Duplicator on a finite τ -structure \mathfrak{A} as follows.

- Duplicator picks a countable τ -structure \mathfrak{B} such that $\text{CSP}(\mathfrak{B}) \cap \mathcal{C} = \emptyset$.
- The game then continues as the existential (ℓ, k) pebble game played by Spoiler and Duplicator on $(\mathfrak{A}, \mathfrak{B})$.

In Section 6 we show that a GSO sentence Φ is equivalent to a Datalog program of width (ℓ, k) if and only if

- $\llbracket \Phi \rrbracket$ is closed under homomorphisms, and
- Spoiler wins the existential (ℓ, k) -game for $\llbracket \Phi \rrbracket$ on \mathfrak{A} if and only if $\mathfrak{A} \models \Phi$.

We also show that for every GSO sentence Φ whose class of finite models \mathcal{C} is closed under homomorphisms and for all $\ell, k \in \mathbb{N}$ there exists a canonical Datalog program Π of width (ℓ, k) for \mathcal{C} (Theorem 24). To prove these results, we first show that every class of finite structures in GSO whose complement is closed under homomorphisms is a finite union of CSPs that can also be expressed in GSO (Lemma 17; an analogous statement holds for MSO). Moreover, every CSP in GSO is the CSP of a countable ω -categorical structure (Corollary 10). We present an example of such a CSP which is even expressible in MSO and coNP-complete, and hence not the CSP of a reduct of a finitely bounded homogeneous structure, unless $\text{NP}=\text{coNP}$ (Proposition 20). Note that our results imply that every class of finite structures that can be expressed both in GSO and in Datalog is a finite intersection of the complements of CSPs for ω -categorical structures. In general, it is not true that a Datalog program describes a finite intersection of complements of CSPs (we present a counterexample in Example 19).

2 Preliminaries

In the entire text, τ denotes a finite signature containing relation symbols and sometimes also constant symbols. If $R \in \tau$ is a relation symbol, we write $\text{ar}(R)$ for its arity. If \mathfrak{A} is a τ -structure we use the corresponding capital roman A letter to denote the domain of \mathfrak{A} ; the domains of structures are assumed to be non-empty. If $R \in \tau$, then $R^{\mathfrak{A}} \subseteq A^{\text{ar}(R)}$ denotes the corresponding relation of \mathfrak{A} .

A *primitive positive τ -formula* (in database theory also *conjunctive query*) is a first-order τ -formula without disjunction, negation, and universal quantification. Every primitive positive formula is equivalent to a formula of the form

$$\exists x_1, \dots, x_n (\psi_1 \wedge \dots \wedge \psi_m)$$

where ψ_1, \dots, ψ_m are atomic τ -formulas. The *canonical database* of a primitive positive τ -formula ϕ which does not contain the equality symbol is the τ -structure \mathfrak{S}_ϕ whose vertices are the variables of ϕ and where $(y_1, \dots, y_n) \in R^{\mathfrak{S}_\phi}$ if and only if ϕ contains the conjunct $R(y_1, \dots, y_n)$. An *existential*

positive τ -formula is a first-order τ -formula without negation and universal quantification. It is easy to see that every existential positive τ -formula is a disjunction of primitive positive τ -formulas (and hence referred to as a *union of conjunctive queries* in database theory). We write $\psi(x_1, \dots, x_n)$ if the free variables of ψ are from x_1, \dots, x_n . If \mathfrak{A} is a τ -structure and $\psi(x_1, \dots, x_n)$ is a τ -formula, then the relation

$$\{(a_1, \dots, a_n) \mid \mathfrak{A} \models \psi(a_1, \dots, a_n)\}$$

is called the relation *defined by ψ over \mathfrak{A}* ; if ψ can be chosen to be primitive positive (or existential positive) then R is called *primitively positively definable* (or *existentially positively definable*, respectively).

For all logics over the signature τ considered in this text, we say that two formulas $\Phi(x_1, \dots, x_n)$ and $\Psi(x_1, \dots, x_n)$ are *equivalent* if for all τ -structures \mathfrak{A} and all $a_1, \dots, a_n \in A$ we have

$$\mathfrak{A} \models \Phi(a_1, \dots, a_n) \Leftrightarrow \mathfrak{A} \models \Psi(a_1, \dots, a_n).$$

Formulas without free variables are called *sentences*; in database theory, formulas are often called *queries* and sentences are often called *Boolean queries*. If Φ is a sentence, we write $\llbracket \Phi \rrbracket$ for the class of all finite models of Φ .

A *reduct* of a relational structure \mathfrak{A} is a structure \mathfrak{A}' obtained from \mathfrak{A} by dropping some of the relations, and \mathfrak{A} is called an *expansion* of \mathfrak{A}' .

2.1 Datalog

In this section we refer to the finite set of relation and constant symbols τ as *EDBs* (for *extensional database predicates*). Let ρ be a finite set of new relation symbols, called the *IDBs* (for *intensional database predicates*). A Datalog program is a set of rules of the form

$$\psi_0 :- \psi_1, \dots, \psi_n$$

where ψ_0 is an atomic ρ -formula and ψ_1, \dots, ψ_n are atomic $(\rho \cup \tau)$ -formulas. If \mathfrak{A} is a τ -structure, and Π is a Datalog program with EDBs τ and IDBs ρ , then a $(\tau \cup \rho)$ -expansion \mathfrak{A}' of \mathfrak{A} is called a *fixed point of Π on \mathfrak{A}* if \mathfrak{A}' satisfies the sentence

$$\forall \bar{x} (\psi_0 \vee \neg \psi_1 \vee \dots \vee \neg \psi_n)$$

for each rule $\psi_0 :- \psi_1, \dots, \psi_n$. If \mathfrak{A}_1 and \mathfrak{A}_2 are two $(\rho \cup \tau)$ -structures with the same domain A , then $\mathfrak{A}_1 \cap \mathfrak{A}_2$ denotes the $(\rho \cup \tau)$ -structure with domain A such that $R^{\mathfrak{A}_1 \cap \mathfrak{A}_2} := R^{\mathfrak{A}_1} \cap R^{\mathfrak{A}_2}$. Note that if \mathfrak{A}_1 and \mathfrak{A}_2 are two fixed points of Π on \mathfrak{A} , then $\mathfrak{A}_1 \cap \mathfrak{A}_2$ is a fixed point of Π on \mathfrak{A} , too. Hence, there exists a unique smallest (with respect to inclusion) fixed point of Π on \mathfrak{A} , which we denote by $\Pi(\mathfrak{A})$. It is well-known that if \mathfrak{A} is a finite structure then $\Pi(\mathfrak{A})$ can be computed in polynomial time in the size of \mathfrak{A} [23]. If $R \in \rho$, we also say that Π *computes $R^{\Pi(\mathfrak{A})}$ on \mathfrak{A}* . A Datalog program together with a distinguished predicate $R \in \rho$ may also be viewed as a formula, which we also call a *Datalog query*, and which over a given τ -structure \mathfrak{A} denotes the relation $R^{\Pi(\mathfrak{A})}$. If the distinguished predicate has arity 0, we often call it the *goal predicate*; we say that Π *derives goal on \mathfrak{A}* if $\text{goal}^{\Pi(\mathfrak{A})} = \{()\}$. The class \mathcal{C} of finite τ -structures \mathfrak{A} such that Π derives goal on \mathfrak{A} is called *the class of finite τ -structures computed by Π* , and denoted by $\llbracket \Pi \rrbracket$. Note that this class \mathcal{C} is in universal second-order logic (we have to express that in every expansion of the input by relations for the IDBs that satisfies all the rules of the Datalog program the goal predicate is non-empty).

2.2 Second-Order Logic

Second-order logic is the extension of first-order logic which additionally allows existential and universal quantification over relations; that is, if R is a relation symbol and ϕ is a second-order $\tau \cup \{R\}$ -formula, then $\exists R: \phi$ and $\forall R: \phi$ are second-order τ -formulas. If \mathfrak{A} is a τ -structure and Φ is a second-order τ -sentence, we write $\mathfrak{A} \models \Phi$ (and say that \mathfrak{A} is a model of Φ) if \mathfrak{A} satisfies Φ , which is defined in the usual Tarskian style. We write $\llbracket \Phi \rrbracket$ for the class of all finite models of Φ . A second-order formula is called *monadic* if all second-order variables are unary. We use syntactic sugar and also write $\forall x \in X: \psi$ instead of $\forall x (X(x) \Rightarrow \psi)$ and $\exists x \in X: \psi$ instead of $\exists x (X(x) \wedge \psi)$.

2.3 Guarded Second-Order Logic

Guarded Second-order Logic (GSO), introduced by Grädel, Hirsch, and Otto [20], is the extension of *guarded first-order logic* by second-order quantifiers. Guarded (first-order) τ -formulas are defined inductively by the following rules [1]:

1. all atomic τ -formulas are guarded τ -formulas;
2. if ϕ and ψ are guarded τ -formulas, then so are $\phi \wedge \psi$, $\phi \vee \psi$, and $\neg\phi$.
3. if $\psi(\bar{x}, \bar{y})$ is a guarded τ -formula and $\alpha(\bar{x}, \bar{y})$ is an atomic τ -formula such that all free variables of ψ occur in α then $\exists \bar{y}(\alpha(\bar{x}, \bar{y}) \wedge \psi(\bar{x}, \bar{y}))$ and $\forall \bar{y}(\alpha(\bar{x}, \bar{y}) \Rightarrow \psi(\bar{x}, \bar{y}))$ are guarded τ -formulas.

Guarded second-order formulas are defined similarly, but we additionally allow (unrestricted) second-order quantification; GSO generalises Courcelle’s logic MSO_2 from graphs to general relational structures.

Definition 1. *A second-order τ -formula is called guarded if it is defined inductively by the rules (1)-(3) for guarded first-order logic and additionally by second-order quantification.*

There are many semantically equivalent ways of introducing GSO [20]. Let \mathfrak{B} be a τ -structure. Then $(t_1, \dots, t_n) \in B^n$ is called *guarded in \mathfrak{B}* if there exists an atomic τ -formula ϕ and b_1, \dots, b_k such that $\mathfrak{B} \models \phi(b_1, \dots, b_k)$ and $\{t_1, \dots, t_n\} \subseteq \{b_1, \dots, b_k\}$. Note that (for $n = 1$) every element of B is guarded (because of the atomic formula $x = x$). A relation $R \subseteq B^n$ is called *guarded* if all tuples in R are guarded. Note that all unary relations are guarded. If Ψ is an arbitrary second-order sentence, we say that a finite structure \mathfrak{A} *satisfies Ψ with guarded semantics*, in symbols $\mathfrak{A} \models_g \Psi$, if all second-order quantifiers in Ψ are evaluated over guarded relations only. Note that for MSO sentences, the usual semantics and the guarded semantics coincide.

Sometimes, we will also use the term GSO (MSO, Datalog) to denote all problems (i.e., all classes of structures) that can be expressed in the formalism. In particular, this justifies to say that a certain CSP is *in GSO* (MSO, Datalog).

Proposition 2 (see [20]). *Guarded Second-order Logic and full Second-order Logic with guarded semantics are equally expressive.*

It follows that GSO is at least as expressive as MSO. There are Datalog programs that are equivalent to a GSO sentence, but not to an MSO sentence. The proof is based on a variant of an example of a Datalog query in GSO given in [11] (Example 2) and can be found in the next section.

3 A Class in Datalog and GSO, but not in MSO

Proposition 3. *There is a Datalog query that can be expressed in GSO but not in MSO.*

Proof. Let τ be the signature consisting of the binary relation symbols S, T, R, N , and let \mathcal{C} be the class of finite τ -structures such that the following Datalog program with one binary IDB U derives *goal*.

$$\begin{aligned} U(x, y) &:-S(x, y) \\ U(x', y') &:-U(x, y), N(x, x'), N(y, y'), R(x', y') \\ \text{goal} &:-U(x, y), T(x, y) \end{aligned}$$

On the left of Figure 1 one can find an example of a $\{S, T, R, N\}$ -structure \mathfrak{B} where the given Datalog program derives *goal*. To show that \mathcal{C} is not MSO definable, suppose for contradiction that there exists an MSO sentence Φ such that $\llbracket \Phi \rrbracket = \mathcal{C}$. We use Φ to construct an MSO sentence Ψ which holds on a finite word $w \in \{a, b\}^*$ (represented as a structure with signature $P_a, P_b, <$ in the usual way [23]) if and only if $w \in \{a^n b^n \mid n \geq 1\}$; this contradicts Courcelle’s theorem [15]. Let Φ' be the MSO sentence obtained from Φ by replacing all subformulas of Φ of the form

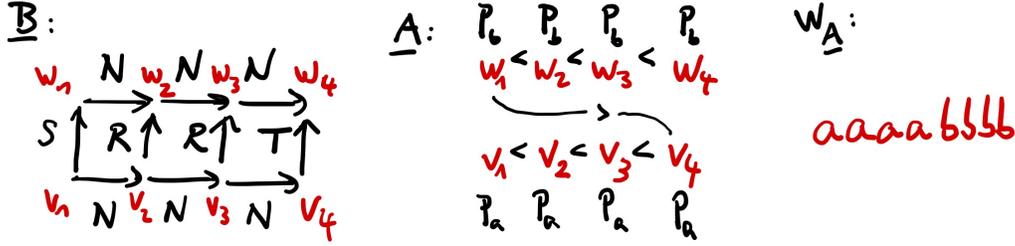


Fig. 1. An example of an $\{S, T, R, N\}$ -structure \mathfrak{B} in the class \mathcal{C} of Proposition 3.

- $S(x, y)$ by a formula $\phi_S(x, y)$ that states that x is the smallest element with respect to $<$, that $P_b(y)$, and that there is no $z < y$ in P_b ;
- $T(x, y)$ by a formula $\phi_T(x, y)$ that states that $P_a(x)$, that there is no $z > x$ in P_a , and that y is the largest element with respect to $<$;
- $R(x, y)$ by the formula $\phi_R(x, y)$ given by $x < y$;
- $N(x, y)$ by a formula $\phi_N(x, y)$ stating that y is the next element after x with respect to $<$.

The resulting MSO sentence Ψ_1 has the signature $\{P_a, P_b, <\}$; let Ψ be the conjunction of Ψ_1 with the sentence Ψ_2 which states that for all $x, y \in A$, if $x < y$ and $P_a(y)$ then $P_a(x)$. We first show that if \mathfrak{A} is a $\{<, P_a, P_b\}$ -structure that represents a word $w_{\mathfrak{A}} \in \{a, b\}^*$, then $\mathfrak{A} \models \Psi$ if and only if $w_{\mathfrak{A}}$ is of the form $a^n b^n$ for some $n \geq 1$. Let \mathfrak{B} be the $\{S, T, R, N\}$ -structure such that for $X \in \{S, T, R, N\}$ we have $X^{\mathfrak{B}} := \{(x, y) \mid \mathfrak{A} \models \phi_X(x, y)\}$. See Figure 1 for an example of a structure \mathfrak{A} such that $w_{\mathfrak{A}} = a^4 b^4$ and the corresponding $\{S, T, R, N\}$ -structure \mathfrak{B} .

If $w_{\mathfrak{A}}$ is of the form $a^n b^n$ for some $n \geq 1$, then \mathfrak{A} clearly satisfies Ψ_2 . To show that it also satisfies Ψ_1 , let $v_1, \dots, v_n, w_1, \dots, w_n \in A$ be such that $\{v_1, \dots, v_n\} = P_a^{\mathfrak{A}}$ and $\{w_1, \dots, w_n\} = P_b^{\mathfrak{A}}$ such that for all $i, j \in \{1, \dots, n\}$, if $i < j$ then $v_i <^{\mathfrak{A}} v_j$ and $w_i <^{\mathfrak{A}} w_j$. Then

$$\begin{aligned} (v_1, w_1) &\in S^{\mathfrak{B}}, & (v_n, w_n) &\in T^{\mathfrak{B}}, \\ (v_i, w_i) &\in R^{\mathfrak{B}} \text{ for all } i \in \{2, \dots, n-1\}, \\ (v_i, v_{i+1}), (w_i, w_{i+1}) &\in N^{\mathfrak{B}} \text{ for all } i \in \{1, \dots, n-1\}. \end{aligned} \quad (1)$$

It follows that \mathfrak{B} satisfies Φ and therefore $\mathfrak{A} \models \Psi$.

For the converse direction, suppose that $\mathfrak{A} \models \Psi$. Clearly, $w_{\mathfrak{A}} \in a^* b^*$ because $\mathfrak{A} \models \Psi_2$. Moreover, since $\mathfrak{A} \models \Psi_1$ we have that $\mathfrak{B} \models \Phi$, and hence there exist $n \in \mathbb{N}$ and elements $v_1, \dots, v_n, w_1, \dots, w_n \in A$ such that \mathfrak{B} satisfies (1). We first prove that $P_a^{\mathfrak{A}} = \{v_1, \dots, v_n\}$ and $|P_a^{\mathfrak{A}}| = n$. Since $(v_n, w_n) \in T^{\mathfrak{B}}$ we have $\phi_T(v_n, w_n)$ and hence $v_n \in P_a^{\mathfrak{A}}$. Since $\mathfrak{B} \models N(v_1, v_2), \dots, N(v_{n-1}, v_n)$ we have that $v_1 < v_2 < \dots < v_{n-1} < v_n$ holds in \mathfrak{A} and it also follows that $|P_a^{\mathfrak{A}}| = n$. Then for every $i \in n$ we have that $v_i \in P_a^{\mathfrak{A}}$ because $v_i \leq v_n$, $v_n \in P_a^{\mathfrak{A}}$, and $w_{\mathfrak{A}} \in a^* b^*$. Now suppose for contradiction that there exists $x \in P_a^{\mathfrak{A}} \setminus \{v_1, \dots, v_n\}$; choose x largest with respect to $<^{\mathfrak{A}}$. Since $(v_n, w_n) \in T^{\mathfrak{B}}$ and $x \in P_a^{\mathfrak{A}}$ we must have $x \leq v_n$, and hence $x < v_n$ since $x \notin \{v_1, \dots, v_n\}$. Then there exists $y \in A$ such that $\phi_N(x, y)$ holds in \mathfrak{A} . Since $y \leq v_n$, $v_n \in P_a^{\mathfrak{A}}$, and $w_{\mathfrak{A}} \in a^* b^*$, we must have $P_a^{\mathfrak{A}}$. By the maximal choice of x we get that $y = v_i$ for some $i \in \{1, \dots, n\}$. But then $\phi_N(x, v_i)$ implies that $x \in \{v_1, \dots, v_{n-1}\}$, a contradiction. Similarly, one can prove that $P_b^{\mathfrak{A}} = \{w_1, \dots, w_n\}$ and that $|P_b^{\mathfrak{A}}| = n$. This implies that $w_{\mathfrak{A}} = a^n b^n$.

We finally have to prove that \mathcal{C} is in GSO. Let Φ be the GSO $\{S, T, R, N\}$ sentence with existentially quantified unary relations V, W , and existentially quantified binary relations $R' \subseteq R$ and $N' \subseteq N$, which states that

- there are elements $v_1, v_n \in V$ and $w_1, w_n \in W$ such that $S(v_1, w_1)$ and $T(v_n, w_n)$ hold;
- for every $x \in V \setminus \{v_1\}$ there exists a unique element $y \in V \setminus \{v_n\}$ such that $N'(y, x)$ holds;
- for every $x \in V \setminus \{v_n\}$ there exists a unique element $y \in V \setminus \{v_1\}$ such that $N'(x, y)$ holds;

- for every $x \in W \setminus \{w_1\}$ there exists a unique element $y \in W \setminus \{w_n\}$ such that $N'(y, x)$ holds;
- for every $x \in W \setminus \{w_n\}$ there exists a unique element $y \in W \setminus \{w_1\}$ such that $N'(x, y)$ holds;
- for all $v \in V$ and $w \in W$ we have that $N'(v_1, v) \wedge N'(w_1, w)$ implies $R'(v, w)$.
- for all $v, v' \in V \setminus \{v_1, v_n\}$ and $w, w' \in W \setminus \{w_1, w_n\}$ we have that $R'(v, w) \wedge N'(v, v') \wedge N'(w, w')$ implies $R'(v, w)$.
- For all $v \in V$ and $w \in W$ we have that $N'(v, v_n) \wedge N'(w, w_n)$ implies $R'(v, w)$.

Then Φ holds on a finite $\{S, T, R, N\}$ -structure \mathfrak{B} if and only if B has elements $v_1, \dots, v_n, w_1, \dots, w_n$ satisfying (1), which is the case if and only if $\mathfrak{B} \in \mathcal{C}$. \square

4 Homomorphism-Closed GSO

We prove that the class of finite models of a GSO sentence is a finite union of CSPs of ω -categorical structures whenever its complement is closed under homomorphisms. In particular, every CSP in GSO (and therefore every CSP in MSO) is the CSP of an ω -categorical structure. CSPs that can be formulated as the CSP of an ω -categorical structure have been characterised [8]; this characterisation will be recalled in the next section.

4.1 CSPs for Countably Categorical Structures

By the theorem of Ryll-Nardzewski, a countable structure \mathfrak{B} is ω -categorical if and only if for every $n \in \mathbb{N}$ there are finitely many orbits of the componentwise action of the automorphism group of \mathfrak{B} on B^n (see, e.g., [21]). We now present a condition that characterises classes of structures that are CSPs of ω -categorical structures. Let \mathcal{C} be a class of finite τ -structures. Let \mathcal{A}_n be the class of primitive positive τ -formulas with free variables x_1, \dots, x_n whose canonical database is in \mathcal{C} . We define $\sim_n^{\mathcal{C}}$ to be the equivalence relation on \mathcal{A} such that $\phi_1 \sim_n^{\mathcal{C}} \phi_2$ holds if for all primitive positive τ -formulas $\psi(x_1, \dots, x_n)$ we have that the canonical database of $\phi_1(x_1, \dots, x_n) \wedge \psi(x_1, \dots, x_n)$ belongs to \mathcal{C} if and only if the canonical database of $\phi_2(x_1, \dots, x_n) \wedge \psi(x_1, \dots, x_n)$ belongs to \mathcal{C} . The *index* of an equivalence relation is the number of its equivalence classes.

Theorem 4 (Bodirsky, Hils, Martin [8]). *Let \mathcal{C} be a constraint satisfaction problem. Then there is an ω -categorical structure \mathfrak{B} such that $\mathcal{C} = \text{CSP}(\mathfrak{B})$ iff $\sim_n^{\mathcal{C}}$ has finite index for all n . Moreover, the structure \mathfrak{B} can be chosen so that for all $n \in \mathbb{N}$ the orbits of the componentwise action of the automorphism group of \mathfrak{B} on B^n are primitively positively definable in \mathfrak{B} .*

Example 5. The structure $\mathfrak{B}_1 := (\mathbb{Z}; <)$ is not ω -categorical. However, $\sim_n^{\text{CSP}(\mathfrak{B}_1)}$ has finite index for all n , and indeed $\text{CSP}(\mathbb{Z}; <) = \text{CSP}(\mathbb{Q}; <)$ and $(\mathbb{Q}; <)$ is ω -categorical. On the other hand, for $\mathfrak{B}_2 := (\mathbb{Z}; \text{Succ})$ we have that the index $\sim_2^{\text{CSP}(\mathfrak{B}_2)}$ is infinite, and it follows that there is no ω -categorical structure \mathfrak{B} such that $\text{CSP}(\mathfrak{B}_2) = \text{CSP}(\mathfrak{B})$; see [5].

A rich source of ω -categorical structures are structures with finite relational signature that are *homogeneous*, i.e., every isomorphism between finite substructures can be extended to an automorphism. There are uncountably many countable homogeneous digraphs with pairwise distinct CSP, and it follows that there are homogeneous digraphs with undecidable CSPs. A structure \mathfrak{B} is called *finitely bounded* if there exists a finite set \mathcal{F} of finite structures such that a finite structure \mathfrak{A} embeds into \mathfrak{B} if and only if no structure in \mathcal{F} embeds into \mathfrak{A} .

It is well-known that if a structure is ω -categorical, then all of its *reducts* are ω -categorical as well [21]. Moreover, it is easy to see that the CSP of reducts of finitely bounded structures is in NP. It has been conjectured that the CSP of reducts of finitely bounded homogeneous structures is in P or NP-complete [10]; this conjecture generalises the finite-domain complexity dichotomy that was conjectured by Feder and Vardi [18] and proved by Bulatov [13] and by Zhuk [25].

4.2 Quantifier Rank

In this section, it will be convenient to work with signatures that also contain constant symbols. The *quantifier rank* of a second-order τ -formula Φ is the maximal number of nested (first-order or second-order) quantifiers in Φ ; for this definition, we view Φ as a second-order sentence with guarded semantics, just as in [4]. If \mathfrak{A} and \mathfrak{B} are τ -structures and $q \in \mathbb{N}$ we write $\mathfrak{A} \equiv_q^{\text{GSO}} \mathfrak{B}$ if \mathfrak{A} and \mathfrak{B} satisfy the same GSO τ -sentences of quantifier rank at most q .

Lemma 6 (Proposition 3.3 in [4]). *Let $q \in \mathbb{N}$ and τ be a finite signature with relation and constant symbols. Then \equiv_q^{GSO} is an equivalence relation with finite index on the class of all finite τ -structures. Moreover, every class of \equiv_q^{GSO} can be defined by a single GSO sentence with quantifier rank q . The analogous statements hold for MSO as well.*

If \mathfrak{A} is a τ -structure and \bar{a} is a k -tuple of elements of A , then we write (\mathfrak{A}, \bar{a}) for a $\tau \cup \{c_1, \dots, c_k\}$ -structure expanding \mathfrak{A} where c_1, \dots, c_k denote fresh constant symbols being mapped to the corresponding entries of \bar{a} . If \mathfrak{A} and \mathfrak{B} are τ -structures and $\bar{a} \in A^k$, $\bar{b} \in B^k$, and when writing $(\mathfrak{A}, \bar{a}) \equiv_q^{\text{GSO}} (\mathfrak{B}, \bar{b})$ we implicitly assume that we have chosen the same constant symbols for \bar{a} and for \bar{b} .

Lemma 7 (Proposition 3.4 in [4]). *Let $q \in \mathbb{N}$ and let \mathfrak{A} and \mathfrak{B} be τ -structures. Then $\mathfrak{A} \equiv_{q+1}^{\text{GSO}} \mathfrak{B}$ if and only if the following properties hold:*

- (first-order forth) For every $a \in A$, there exists $b \in B$ such that $(\mathfrak{A}, a) \equiv_q^{\text{GSO}} (\mathfrak{B}, b)$.
- (first-order back) For every $b \in B$, there exists $a \in A$ such that $(\mathfrak{A}, a) \equiv_q^{\text{GSO}} (\mathfrak{B}, b)$.
- (second-order forth) For every expansion \mathfrak{A}' of \mathfrak{A} by a guarded relation, there exists an expansion \mathfrak{B}' of \mathfrak{B} by a guarded relation such that $\mathfrak{A}' \equiv_q^{\text{GSO}} \mathfrak{B}'$.
- (second-order back) For every expansion \mathfrak{B}' of \mathfrak{B} by a guarded relation, there exists an expansion \mathfrak{A}' of \mathfrak{A} by a guarded relation such that $\mathfrak{A}' \equiv_q^{\text{GSO}} \mathfrak{B}'$.

In the following, τ denotes a finite relational signature.

Definition 8. *Let $\rho := \{c_1, \dots, c_n\}$ be a finite set of constant symbols. Then \mathcal{D}_n is defined to be the set of all pairs $(\mathfrak{A}, \mathfrak{B})$ of finite $(\tau \cup \rho)$ -structures such that*

- $c^{\mathfrak{A}} = c^{\mathfrak{B}}$ for all constant symbols $c \in \rho$;
- $\{c_1^{\mathfrak{A}}, \dots, c_n^{\mathfrak{A}}\} = A \cap B = \{c_1^{\mathfrak{B}}, \dots, c_n^{\mathfrak{B}}\}$.

We write $\mathfrak{A} \uplus \mathfrak{B}$ for the structure with domain $A \cup B$ such that $R^{\mathfrak{A} \uplus \mathfrak{B}} := R^{\mathfrak{A}} \cup R^{\mathfrak{B}}$ for each relation symbol $R \in \tau$ and $c^{\mathfrak{A} \uplus \mathfrak{B}} = c^{\mathfrak{A}} = c^{\mathfrak{B}}$ for each constant symbol $c \in \rho$.

The following theorem in the special case of $n = 0$ is Proposition 4.1 in [4].

Theorem 9. *Let $q, n, r, s \in \mathbb{N}$, let $(\mathfrak{A}_1, \mathfrak{B}_1), (\mathfrak{A}_2, \mathfrak{B}_2) \in \mathcal{D}_n$, and let $\bar{a}_1 \in (A_1)^r$, $\bar{a}_2 \in (A_2)^r$, $\bar{b}_1 \in (B_1)^s$, $\bar{b}_2 \in (B_2)^s$ be such that $(\mathfrak{A}_1, \bar{a}_1) \equiv_q^{\text{GSO}} (\mathfrak{A}_2, \bar{a}_2)$ and $(\mathfrak{B}_1, \bar{b}_1) \equiv_q^{\text{GSO}} (\mathfrak{B}_2, \bar{b}_2)$. Then*

$$(\mathfrak{A}_1 \uplus \mathfrak{B}_1, \bar{a}_1, \bar{b}_1) \equiv_q^{\text{GSO}} (\mathfrak{A}_2 \uplus \mathfrak{B}_2, \bar{a}_2, \bar{b}_2).$$

Proof. Our proof is by induction on q . Every quantifier-free formula is a Boolean combination of atomic formulas, so for $q = 0$ it suffices to consider atomic formulas ϕ . By symmetry, it suffices to show that if $(\mathfrak{A}_1 \uplus \mathfrak{B}_1, \bar{a}_1, \bar{b}_1) \models \phi$ then $(\mathfrak{A}_2 \uplus \mathfrak{B}_2, \bar{a}_2, \bar{b}_2) \models \phi$. Then ϕ is built using a relation symbol $R \in \tau$, and the tuple that witnesses the truth of ϕ in $\mathfrak{A}_1 \uplus \mathfrak{B}_1$ must be from $R^{\mathfrak{A}_1}$ or from $R^{\mathfrak{B}_1}$, by the definition of $\mathfrak{A}_1 \uplus \mathfrak{B}_1$. We first consider the former case; the latter case can be treated similarly. If a constant that appears in ϕ is from $A_1 \cap B_1$, then by the definition of \mathcal{D}_n this element is denoted by a constant symbol $c \in \rho$, and therefore we may assume without loss of generality that ϕ is a formula over the signature of $(\mathfrak{A}_1, \bar{a}_1)$. Hence, $(\mathfrak{A}_1, \bar{a}_1) \models \phi$ and by assumption $(\mathfrak{A}_2, \bar{a}_2) \models \phi$. This in turn implies that $(\mathfrak{A}_2 \uplus \mathfrak{B}_2, \bar{a}_2, \bar{b}_2) \models \phi$.

For the inductive step, suppose that the claim holds for q , and that $(\mathfrak{A}_1, \bar{a}_1) \equiv_{q+1}^{\text{GSO}} (\mathfrak{A}_2, \bar{a}_2)$ and $(\mathfrak{B}_1, \bar{b}_1) \equiv_{q+1}^{\text{GSO}} (\mathfrak{B}_2, \bar{b}_2)$. By symmetry and Lemma 7 it suffices to verify the properties (first-order

forth) and (second-order forth). Let $c_1 \in A_1 \cup B_1$. We may assume that $c_1 \in A_1$; the case that $c_1 \in B_1$ can be shown similarly. By Lemma 7, there exists $c_2 \in A_2$ such that $(\mathfrak{A}_1, \bar{a}_1, c_1) \equiv_q^{\text{GSO}} (\mathfrak{A}_2, \bar{a}_2, c_2)$. By the inductive assumption, this implies that

$$(\mathfrak{A}_1 \uplus \mathfrak{B}_1, \bar{a}_1, c_1, \bar{b}_1) \equiv_q^{\text{GSO}} (\mathfrak{A}_2 \uplus \mathfrak{B}_2, \bar{a}_2, c_2, \bar{b}_2)$$

and concludes the proof of (first-order forth).

Now let R be a guarded relation of $\mathfrak{A}_1 \uplus \mathfrak{B}_1$ of arity k . Let \mathfrak{A}'_1 be the expansion of \mathfrak{A}_1 by the guarded relation $R \cap A_1^k$, and \mathfrak{B}'_1 be the expansion of \mathfrak{B}_1 by the guarded relation $R \cap B_1^k$. By Lemma 7 there are expansions \mathfrak{A}'_2 of \mathfrak{A} and \mathfrak{B}'_2 of \mathfrak{B}_2 by guarded relations such that $(\mathfrak{A}'_1, \bar{a}_1) \equiv_q^{\text{GSO}} (\mathfrak{A}'_2, \bar{a}_2)$ and $(\mathfrak{B}'_1, \bar{b}_1) \equiv_q^{\text{GSO}} (\mathfrak{B}'_2, \bar{b}_2)$. By the inductive assumption, this implies that $(\mathfrak{A}'_1 \uplus \mathfrak{B}'_1, \bar{a}_1, \bar{b}_1) \equiv_q^{\text{GSO}} (\mathfrak{A}'_2 \uplus \mathfrak{B}'_2, \bar{a}_2, \bar{b}_2)$, which completes the proof of (second-order forth). \square

Corollary 10. *Let \mathcal{C} be a CSP that can be expressed in GSO. Then there exists a countable ω -categorical structure \mathfrak{B} such that $\mathcal{C} = \text{CSP}(\mathfrak{B})$.*

Proof. Let τ be the signature of \mathcal{C} , and let Φ be a GSO τ -formula with quantifier-rank q such that $\mathcal{C} = \llbracket \Phi \rrbracket$. By Theorem 4 it suffices to show that the equivalence relation $\sim_n^{\mathcal{C}}$ has finite index for every $n \in \mathbb{N}$. Let $\rho := \{c_1, \dots, c_n\}$ be a set of new constant symbols. By Lemma 6, there exists an $m \in \mathbb{N}$ such that \equiv_q^{GSO} has m equivalence classes on $(\tau \cup \rho)$ -structures. If $\phi(x_1, \dots, x_n)$ is a primitive positive τ -formula, then define \mathfrak{S}_ϕ to be the $(\tau \cup \rho)$ -structure obtained from the canonical database of ϕ by setting $c_i^{\mathfrak{S}_\phi} := x_i$ for all $i \in \{1, \dots, n\}$.

We claim that if $\mathfrak{S}_\phi \equiv_q^{\text{GSO}} \mathfrak{S}_\psi$, then $\phi \sim_n^{\mathcal{C}} \psi$. Let $\theta(x_1, \dots, x_n)$ be a primitive positive τ -formula; we may assume that the existentially quantified variables of θ are disjoint from the existentially quantified variables of ϕ and of ψ , so that $(\mathfrak{S}_\phi, \mathfrak{S}_\theta), (\mathfrak{S}_\psi, \mathfrak{S}_\theta) \in \mathcal{D}_n$. Since $\mathfrak{S}_\phi \equiv_q^{\text{GSO}} \mathfrak{S}_\psi$ and $\mathfrak{S}_\theta \equiv_q^{\text{GSO}} \mathfrak{S}_\theta$, we have $\mathfrak{S}_\phi \uplus \mathfrak{S}_\theta \equiv_q^{\text{GSO}} \mathfrak{S}_\psi \uplus \mathfrak{S}_\theta$ by Theorem 9. This implies that the canonical database of $\phi \wedge \theta$ satisfies Φ if and only if the canonical database of $\psi \wedge \theta$ does, which proves the claim.

The claim implies that $\sim_n^{\mathcal{C}}$ has at most m equivalence classes, concluding the proof. \square

Example 11. Let Φ be the following MSO sentence.

$$\forall X (\exists x: X(x) \wedge \exists x, y \in X \forall z \in X (\neg E(x, z) \vee \neg E(y, z)))$$

It is easy to see that $\llbracket \Phi \rrbracket$ is closed under disjoint unions and that its complement is closed under homomorphisms. Corollary 10 implies that there exists a countable ω -categorical structure with $\text{CSP}(\mathfrak{B}) = \llbracket \Phi \rrbracket$.

Example 12. For $k \geq 2$, let Ψ_k be the following MSO sentence.

$$\begin{aligned} \forall X \neg (&\forall x, y (E(x, y) \Rightarrow E(y, x)) \wedge \forall x: \neg E(x, x) \\ &\wedge \exists x: x \in X \wedge \forall x, y, z \in X: \neg (E(x, y) \wedge E(y, z) \wedge E(z, x)) \\ &\wedge \forall \bar{x}, \bar{y} \in X (\bigwedge_{i, j \leq k} \neg E(x_i, x_j) \Rightarrow \exists z \in X \bigwedge_{i \leq k} E(x_i, z) \wedge \bigwedge_{i \leq \ell} \neg E(y_i, z)) \end{aligned}$$

The formula states that a given graph is symmetric and without loops, and does not contain non-empty triangle-free graphs that satisfy the triangle-free k -point extension property. Starting from $k \geq 5$, it is not known whether Ψ_k describes the class of all graphs; see [14]. It is easy to check that $\llbracket \Psi_k \rrbracket$ is closed under disjoint unions and that its complement is closed under homomorphisms, and again the assumptions of Corollary 10 hold.

4.3 Finite Unions of CSPs

In this section we prove that every class in GSO whose complement is closed under homomorphisms is a finite union of CSPs (Lemma 17); the statement announced at the beginning of Section 4 then

follows (Corollary 18). Throughout this section, let \mathcal{C} be a non-empty class of finite τ -structures whose complement is closed under homomorphisms. In particular, \mathcal{C} contains the structure \mathcal{J} with only one element where all relations are empty.

Let \sim be the equivalence relation defined on \mathcal{C} by letting $\mathfrak{A} \sim \mathfrak{B}$ if for every $\mathfrak{C} \in \mathcal{C}$ we have $\mathfrak{A} \uplus \mathfrak{C} \in \mathcal{C}$ if and only if $\mathfrak{B} \uplus \mathfrak{C} \in \mathcal{C}$; here \uplus denotes the usual disjoint union of structures, which is a special case of Definition 8 for $n = 0$. Note that the equivalence classes of \sim are in one-to-one correspondence to the equivalence classes of $\sim_0^{\mathcal{C}}$. Also note that \mathcal{C} is closed under disjoint unions if and only if \sim has only one equivalence class.

Remark 13. Since the complement of \mathcal{C} is closed under homomorphisms, \mathcal{C} is closed under disjoint unions if and only if it has the *joint homomorphism property*: if $\mathfrak{A} \in \mathcal{C}$ and $\mathfrak{B} \in \mathcal{C}$, then there exists a $\mathfrak{C} \in \mathcal{C}$ such that \mathfrak{A} and \mathfrak{B} map homomorphically to \mathfrak{C} . The joint homomorphism property is undecidable even if \mathcal{C} is given by a universal first-order sentence ϕ (i.e., $\mathcal{C} = \llbracket \phi \rrbracket$) [12].

If $\mathfrak{A} \in \mathcal{C}$, then we write $[\mathfrak{A}]$ for the equivalence class of \mathfrak{A} with respect to \sim . The following observations are immediate consequences from the definitions:

1. $[\mathfrak{A}]$ is closed under disjoint unions.
2. \sim -equivalence classes are closed under homomorphic equivalence.
3. $\mathfrak{A} \in [\mathcal{J}]$ if and only if $\mathfrak{A} \uplus \mathfrak{B} \in \mathcal{C}$ for all $\mathfrak{B} \in \mathcal{C}$.

Lemma 14. *Let $\mathfrak{A} \in \mathcal{C}$ and let \mathcal{D} be the smallest subclass of \mathcal{C} that contains $[\mathfrak{A}]$ and whose complement is closed under homomorphisms. Then*

1. \mathcal{D} is a union of equivalence classes of \sim , and
2. if \sim has more than one equivalence class, then $\mathcal{C} \setminus \mathcal{D}$ is non-empty.

Proof. Let $\mathfrak{C} \in \mathcal{D}$, let \mathfrak{B} be a finite structure with a homomorphism to \mathfrak{C} , and let $\mathfrak{B}' \in [\mathfrak{B}]$. Since $\mathfrak{B} \uplus \mathfrak{C}$ and \mathfrak{C} are homomorphically equivalent, we have that $\mathfrak{B} \uplus \mathfrak{C} \sim \mathfrak{C}$. We claim that $\mathfrak{B}' \uplus \mathfrak{C} \sim \mathfrak{C}$. To see this, let $\mathfrak{D} \in \mathcal{C}$. Then

$$\begin{aligned} \mathfrak{C} \uplus \mathfrak{D} \in \mathcal{C} &\Leftrightarrow (\mathfrak{B} \uplus \mathfrak{C}) \uplus \mathfrak{D} \in \mathcal{C} && \text{(since } \mathfrak{B} \uplus \mathfrak{C} \sim \mathfrak{C} \text{)} \\ &\Leftrightarrow \mathfrak{B} \uplus (\mathfrak{C} \uplus \mathfrak{D}) \in \mathcal{C} \\ &\Leftrightarrow \mathfrak{B}' \uplus (\mathfrak{C} \uplus \mathfrak{D}) \in \mathcal{C} && \text{(since } \mathfrak{B} \sim \mathfrak{B}' \text{)} \\ &\Leftrightarrow (\mathfrak{B}' \uplus \mathfrak{C}) \uplus \mathfrak{D} \in \mathcal{C} \end{aligned}$$

which shows the claim. So $\mathfrak{B}' \uplus \mathfrak{C} \in [\mathfrak{C}] = [\mathfrak{A}]$. Since \mathfrak{B}' has a homomorphism to $\mathfrak{B}' \uplus \mathfrak{C}$ we obtain that $\mathfrak{B}' \in \mathcal{D}$; this proves the first statement.

To prove the second statement, first observe that the statement is clear if $\mathfrak{A} \in [\mathcal{J}]$, since the complement of $[\mathcal{J}]$ is closed under homomorphisms. The statement therefore follows from the assumption that \sim has more than one equivalence class. Otherwise, if $\mathfrak{A} \notin [\mathcal{J}]$, then there exists a structure $\mathfrak{B} \in \mathcal{C}$ such that $\mathfrak{A} \uplus \mathfrak{B} \notin \mathcal{C}$. Then $\mathfrak{B} \in \mathcal{C} \setminus \mathcal{D}$ can be shown indirectly as follows: otherwise \mathfrak{B} would have a homomorphism to a structure $\mathfrak{A}' \in [\mathfrak{A}]$. Since $\mathfrak{B} \uplus \mathfrak{A}'$ is homomorphically equivalent to \mathfrak{A}' , we have $\mathfrak{B} \uplus \mathfrak{A}' \sim \mathfrak{A}' \sim \mathfrak{A}$ and in particular $\mathfrak{B} \uplus \mathfrak{A}' \in \mathcal{C}$. But $\mathfrak{B} \uplus \mathfrak{A}' \in \mathcal{C}$ if and only if $\mathfrak{B} \uplus \mathfrak{A} \in \mathcal{C}$ since $\mathfrak{A} \sim \mathfrak{A}'$. This is in contradiction to our assumption on \mathfrak{B} . \square

Example 15. We consider a signature $\tau := \{R_1, R_2, R_3\}$ of unary relation symbols. Define for every $i \in \{1, 2, 3\}$ the τ -structure \mathfrak{S}_i to be a one-element structure where R_i is non-empty and R_j , for $j \neq i$, is empty. Let

$$\mathcal{C} := \text{CSP}(\mathfrak{S}_1 \uplus \mathfrak{S}_2) \cup \text{CSP}(\mathfrak{S}_2 \uplus \mathfrak{S}_3) \cup \text{CSP}(\mathfrak{S}_3 \uplus \mathfrak{S}_1).$$

Clearly, the complement of \mathcal{C} is closed under homomorphisms. The equivalence classes of \sim can be described as follows. For distinct $i, j \in \{1, 2, 3\}$,

$$\begin{aligned} [\mathfrak{S}_i \uplus \mathfrak{S}_j] &= \text{CSP}(\mathfrak{S}_i \uplus \mathfrak{S}_j) \setminus (\text{CSP}(\mathfrak{S}_i) \cup \text{CSP}(\mathfrak{S}_j)) \\ [\mathfrak{S}_i] &= \text{CSP}(\mathfrak{S}_i) \setminus [\mathcal{J}] \\ [\mathcal{J}] &= \text{CSP}(\mathcal{J}). \end{aligned}$$

Let $q \in \mathbb{N}$ and let τ be a finite relational signature. Recall that Lemma 6 asserts that the equivalence relation \equiv_q^{GSO} on the class of finite τ -structures has finitely many equivalence classes $\mathcal{C}_1, \dots, \mathcal{C}_m$, and that each of the equivalence classes \mathcal{C}_i can be defined by a single GSO τ -sentence Ψ_i with quantifier rank q ; we write $T_q^\tau := \{\Psi_1, \dots, \Psi_m\}$ for this set of GSO sentences. If Φ is a GSO τ -sentence of quantifier rank q , let Φ_1, \dots, Φ_n , for some $n \leq m$, be the set of all sentences in T_q^τ that imply Φ . Then n is called the *degree* of Φ . Let \sim be the equivalence relation defined in the beginning of this section for the class $\mathcal{C} := \llbracket \Phi \rrbracket$.

Lemma 16. *Let Φ be a GSO τ -sentence of quantifier rank q . Then for every class $\mathcal{D} \subseteq \llbracket \Phi \rrbracket$ which is a union of \sim -classes there exists a GSO sentence Ψ of quantifier rank q such that $\mathcal{D} = \llbracket \Psi \rrbracket$.*

Proof. Let Φ_1, \dots, Φ_n be all the formulas in T_q^τ that imply Φ . As in the proof of Corollary 10 one can use Theorem 9 to show for all finite τ -structures $\mathfrak{A}, \mathfrak{B}$ that if $\mathfrak{A} \equiv_q^{\text{GSO}} \mathfrak{B}$, then $\mathfrak{A} \sim \mathfrak{B}$. It follows that there exists $I \subseteq \{1, \dots, n\}$ such that $\mathcal{D} = \bigcup_{i \in I} \llbracket \Phi_i \rrbracket$, and $\bigvee_{i \in I} \Phi_i$ is the desired GSO sentence. \square

Lemma 17. *Let Φ be a GSO τ -sentence such that the complement of $\llbracket \Phi \rrbracket$ is closed under homomorphisms. Then there are GSO sentences Φ_1, \dots, Φ_m each of which describes a CSP such that Φ is equivalent to $\Phi_1 \vee \dots \vee \Phi_m$. If Φ is an MSO sentence, then Φ_1, \dots, Φ_m can be chosen to be MSO sentences as well.*

Proof. Let q be the quantifier rank of Φ . We prove the statement by induction on the degree n of Φ . Note that Lemma 16 implies that \sim has at most n equivalence classes on τ -structures. Hence, if $n = 1$, then $\llbracket \Phi \rrbracket$ is closed under disjoint unions, and we are done.

Let $\mathfrak{A}_1, \dots, \mathfrak{A}_s$ be τ -structures such that $\{\llbracket \mathfrak{A}_1 \rrbracket, \dots, \llbracket \mathfrak{A}_s \rrbracket\}$ is the set of all equivalence classes of \sim that are distinct from $\llbracket \mathcal{J} \rrbracket$. Let \mathcal{C}_i be the smallest subclass of $\llbracket \Phi \rrbracket$ that contains $\llbracket \mathfrak{A}_i \rrbracket$ and whose complement is closed under homomorphisms. Note that $\llbracket \Phi \rrbracket = \bigcup_{i \leq s} \mathcal{C}_i$ since $\llbracket \mathcal{J} \rrbracket$ is contained in \mathcal{C}_i for all $i \leq s$. By Lemma 14 (1), each class \mathcal{C}_i is a union of \sim -classes, and by Lemma 16 it follows that there exists a GSO sentence Ψ_i of quantifier rank q such that $\mathcal{C}_i = \llbracket \Psi_i \rrbracket$. Hence, Φ is equivalent to $\bigvee_{i \leq s} \Psi_i$. Lemma 14 (2) asserts that $\llbracket \Phi \rrbracket \setminus \mathcal{C}_i$ is non-empty, and hence the degree of Ψ_i must be strictly smaller than n . The statement now follows from the inductive assumption. The same argument applies to MSO as well. \square

Lemma 17 together with Corollary 10 implies the following.

Corollary 18. *Every GSO sentence which is closed under homomorphisms is equivalent to a finite conjunction of GSO sentences each of which describes the complement of a CSP of an ω -categorical structure. The analogous statement holds for MSO.*

Not every class of structures that can be expressed in Second-order Logic is a finite intersection of complements of CSPs. We even have an example of a class of finite τ -structures that can be expressed in Datalog but cannot be written in this form.

Example 19. Let S and T be unary, and let R be a binary relation symbol. Let \mathcal{C} be the class of all finite $\{S, T, R\}$ -structures \mathfrak{A} such that the following Datalog program Π with the binary IDB E derives goal on \mathfrak{A} .

$$\begin{aligned} E(x, y) &:- S(x), S(y) \\ E(x, y) &:- E(x', y'), R(x', x), R(y', y) \\ \text{goal} &:- T(x), E(x, x'), R(x', y) \end{aligned}$$

For $n \in \mathbb{N}$, let \mathfrak{P}_n be the $\{S, T, R\}$ -structure on the domain $\{1, \dots, n\}$ with

$$S^{\mathfrak{P}_n} := \{1\} \quad T^{\mathfrak{P}_n} := \{n\} \quad R^{\mathfrak{P}_n} := \{(i, i+1) \mid i \in \{1, \dots, n-1\}\}.$$

It is easy to see that each of the structures in $\{\mathfrak{P}_n \mid n \geq 1\}$ is not contained in \mathcal{C} , and that the disjoint union of \mathfrak{P}_i and \mathfrak{P}_j , for $i \neq j$, is contained in \mathcal{C} . It follows that \mathcal{C} is not a finite intersection of complements of CSPs (and, by Corollary 18, cannot be expressed in GSO).

5 A coNP-complete CSP in MSO

In this section we show that the class of CSPs in MSO is (under complexity-theoretic assumptions) larger than the class of CSPs for reducts of finitely bounded structures (see Section 4.1). Let $\mathcal{T} = \{\mathfrak{T}_2, \mathfrak{T}_3, \dots\}$ be the set of *Henson tournaments*: the tournament \mathfrak{T}_n , for $n \geq 2$, has vertices $0, 1, \dots, n+1$ and the following edges:

- $(i, i+1)$ for $i \in \{0, \dots, n\}$;
- $(0, n+1)$;
- (j, i) for $i+1 < j$ and $(i, j) \neq (0, n+1)$.

The class \mathcal{C} of all finite loopless digraphs that do not embed any of the digraphs from \mathcal{T} is an amalgamation class, and hence there exists a homogenous structure \mathfrak{H} with age \mathcal{C} . It has been shown in [7] that $\text{CSP}(\mathfrak{H})$ is coNP-complete.

Proposition 20. *$\text{CSP}(\mathfrak{H})$ can be expressed in MSO.*

Proof. We have to find an MSO sentence that holds on a given digraph $(V; E)$ if and only if $(V; E)$ does not embed any of the tournaments from \mathcal{T} . We specify an MSO $\{X, E\}$ -sentence Φ , for a unary relation symbol X , that is true on a finite $\{X, E\}$ -structure \mathfrak{G} if and only if $(X^{\mathfrak{G}}; E^{\mathfrak{G}})$ is isomorphic to \mathfrak{T}_n , for some $n \geq 2$. In ϕ we existentially quantify over

- two vertices $s, t \in X$ (that stand for the vertex 0 and the vertex $n+1$ in \mathfrak{T}_n).
- a partition of $X \setminus \{s\}$ into two sets A and B (they stand for the set of even and the set of odd numbers in $\{1, \dots, n+1\}$).

The formula Φ has the following conjuncts:

1. a first-order formula that states that E defines a tournament on X ;
2. a first-order formula that expresses that E is a linear order on A with maximal element a ;
3. a first-order formula that expresses that E is a linear order on B with maximal element b ;
4. $E(s, t)$, $E(s, a)$, $E(a, b)$, and $E(x, s)$ for all $x \in X \setminus \{a, t\}$;
5. a first-order formula that states that if there is an edge from an element $x \in A$ to an element $y \in B$ then there is precisely one element $z \in A$ such that $(y, z), (z, x) \in E$, unless $y = t$;
6. a first-order formula that states that if there is an edge from an element $x \in B$ to an element $y \in A$ then there is precisely one element $z \in B$ such that $(y, z), (z, x) \in E$, unless $y = t$.

We claim that the MSO sentence $\forall x: \neg E(x, x) \wedge \forall X: \neg \Phi$ holds on a finite digraph if and only if the digraph is loopless and does not embed \mathfrak{T}_n , for all $n \geq 3$. The forwards implication easily follows from the observation that if $(X; T)$ is isomorphic to \mathfrak{T}_n , for some $n \geq 2$, then ϕ holds; this is straightforward from the construction of Φ (and the explanations above given in brackets). Conversely, suppose that Φ holds. Then $(X; T)$ is a tournament. We construct an isomorphism f from $(X; T)$ to $\mathfrak{T}_{|X|-1}$ as follows. Define $f(s) := 0$, $f(a) := 1$, and $f(b) := 2$. Since $E(a, b)$, by item 5 there exists exactly one $a' \in A$ such that $E(b, a')$ and $E(a', a)$. Define $f(a') := 3$. If $a' = t$ then we have found an isomorphism with \mathfrak{T}_2 . Otherwise, the partial map f defined so far is an embedding into \mathfrak{T}_n for some $n \geq 3$. Item 6 and $E(b, a')$ imply that there exists exactly one $b' \in B$ such that $E(a', b')$ and $E(b', b)$, and we define $f(b') := 4$. Continuing in this manner, we eventually define f on all of X and find an isomorphism with $\mathfrak{T}_{|X|-1}$. \square

This shows that $\text{CSP}(\mathfrak{H})$ cannot be expressed, unless $\text{NP} = \text{coNP}$, as $\text{CSP}(\mathfrak{B})$ for some reduct of a finitely bounded structure and such CSPs are in NP. We do not know how to show this statement without complexity-theoretic assumptions, even if we just want to rule out that $\text{CSP}(\mathfrak{H})$ can be expressed as $\text{CSP}(\mathfrak{B})$ for some reduct of a finitely bounded *homogeneous* structure.

6 Canonical Datalog Programs

A remarkable fact about the expressive power of Datalog for constraint satisfaction problems over finite domains is the existence of *canonical Datalog programs* [18]; this has been generalised to CSPs for ω -categorical structures.

Theorem 21 (Bodirsky and Dalmau [6]). *Let \mathfrak{B} be a countable ω -categorical τ -structure. Then for all $\ell, k \in \mathbb{N}$ there exists a canonical Datalog program Π of width (ℓ, k) for the complement of $\text{CSP}(\mathfrak{B})$. Moreover, for every finite τ -structure \mathfrak{A} the following are equivalent:*

- Π derives **goal** on \mathfrak{A} ;
- Spoiler has a winning strategy for the existential (ℓ, k) -pebble game on $(\mathfrak{A}, \mathfrak{B})$.

We later need the following well-known fact.

Lemma 22. *If \mathcal{C}_1 and \mathcal{C}_2 are in Datalog, then so are $\mathcal{C}_1 \cup \mathcal{C}_2$ and $\mathcal{C}_1 \cap \mathcal{C}_2$. If Π_1 and Π_2 are Datalog programs of width (ℓ, k) , then there is a Datalog program Π of width (ℓ, k) for $\llbracket \Pi_1 \rrbracket \cup \llbracket \Pi_2 \rrbracket$ and for $\llbracket \Pi_1 \rrbracket \cap \llbracket \Pi_2 \rrbracket$.*

Proof. For union, let Π be obtained by taking the union of the rules of Π_1 and of Π_2 , possibly after renaming IDB predicate names to make them disjoint except for **goal**. For intersection, we proceed similarly, but we first rename the symbol **goal** in Π_1 to **goal**₁ and the symbol **goal** in Π_2 to **goal**₂. Finally we add the new rule **goal** :– **goal**₁, **goal**₂ to the union of Π_1 and Π_2 . It is clear that these constructions preserve the width. \square

Theorem 23. *Let Φ be a GSO sentence such that $\llbracket \Phi \rrbracket$ is closed under homomorphisms. Let $\ell, k \in \mathbb{N}$. Then there exists a canonical Datalog program Π of width (ℓ, k) for $\llbracket \Phi \rrbracket$.*

Proof. By Corollary 18 there are GSO sentences Φ_1, \dots, Φ_m and ω -categorical structures $\mathfrak{B}_1, \dots, \mathfrak{B}_m$ such that Φ is equivalent to $\Phi_1 \wedge \dots \wedge \Phi_m$ and $\llbracket \neg \Phi_i \rrbracket = \text{CSP}(\mathfrak{B}_i)$. Let Π_i be the canonical Datalog program for $\text{CSP}(\mathfrak{B}_i)$ which exists by Theorem 21. Then Lemma 22 implies that there exists a Datalog program Π such that $\llbracket \Pi \rrbracket = \llbracket \Pi_1 \rrbracket \cap \dots \cap \llbracket \Pi_m \rrbracket$. It is clear that Π is sound for $\llbracket \Phi \rrbracket$. To see that Π is a canonical Datalog program for $\llbracket \Phi \rrbracket$, suppose that \mathfrak{A} is such that some Datalog program Π' of width (ℓ, k) which is sound for $\llbracket \Phi \rrbracket$ derives **goal** on \mathfrak{A} . Since, for every $i \in \{1, \dots, m\}$, the program Π' is also sound for $\llbracket \Phi_i \rrbracket$, and Π_i is a canonical Datalog program for $\llbracket \Phi_i \rrbracket$, the program Π_i derives **goal** on \mathfrak{A} . Hence, $\mathfrak{A} \in \llbracket \Pi \rrbracket = \llbracket \Pi_1 \rrbracket \cap \dots \cap \llbracket \Pi_m \rrbracket$. \square

Theorem 24. *Let Φ be a GSO sentence. Then Φ can be expressed in Datalog if and only if*

1. $\llbracket \Phi \rrbracket$ is closed under homomorphisms, and
2. there exist $\ell, k \in \mathbb{N}$ such that for all finite structures \mathfrak{A} , Spoiler wins the (ℓ, k) -game for $\llbracket \Phi \rrbracket$ on \mathfrak{A} if and only if $\mathfrak{A} \models \Phi$.

Proof. First suppose that $\llbracket \Phi \rrbracket$ is in Datalog. That is, there exists $\ell, k \in \mathbb{N}$ and a Datalog program Π of width (ℓ, k) such that $\llbracket \Phi \rrbracket = \llbracket \Pi \rrbracket$. Then clearly $\llbracket \Phi \rrbracket$ is closed under homomorphisms, and by Lemma 17, there are GSO sentences Φ_1, \dots, Φ_m such that Φ is equivalent to $\Phi_1 \wedge \dots \wedge \Phi_m$ and $\llbracket \Phi_i \rrbracket$ is the complement of a CSP, for each $i \in \{1, \dots, m\}$. Corollary 10 implies that there exists an ω -categorical structure \mathfrak{B}_i such that $\text{CSP}(\mathfrak{B}_i) = \llbracket \neg \Phi_i \rrbracket$. Now suppose that \mathfrak{A} is a finite τ -structure such that $\mathfrak{A} \models \Phi$. Then Spoiler wins the (ℓ, k) -game as follows. Suppose that Duplicator plays the countable structure \mathfrak{B} such that $\text{CSP}(\mathfrak{B}) \cap \llbracket \Phi \rrbracket = \emptyset$. Then $\text{CSP}(\mathfrak{B}) \cap \llbracket \Phi_i \rrbracket = \emptyset$ for some $i \in \{1, \dots, m\}$; otherwise, if there is a structure $\mathfrak{A}_i \in \text{CSP}(\mathfrak{B}) \cap \llbracket \Phi_i \rrbracket$ for every $i \in \{1, \dots, m\}$, then the disjoint union of $\mathfrak{A}_1, \dots, \mathfrak{A}_m$ satisfies Φ_i since Φ_i is closed under homomorphisms, and is in $\text{CSP}(\mathfrak{B})$ since $\text{CSP}(\mathfrak{B})$ is closed under disjoint unions; but this is in contradiction to our assumption that $\text{CSP}(\mathfrak{B}) \cap \llbracket \Phi \rrbracket = \emptyset$. Hence, $\text{CSP}(\mathfrak{B}) \subseteq \text{CSP}(\mathfrak{B}_i)$ and hence there is a homomorphism h from \mathfrak{B} to \mathfrak{B}_i (see [6]). Note that Π is sound for $\text{CSP}(\mathfrak{B}_i)$, and Π derives **goal** on \mathfrak{A} , and hence Theorem 21 implies that Spoiler wins the existential (ℓ, k) -pebble game on $(\mathfrak{A}, \mathfrak{B}_i)$. But since \mathfrak{B} homomorphically maps to \mathfrak{B}_i , this implies that Spoiler wins the existential (ℓ, k) -pebble game on

$(\mathfrak{A}, \mathfrak{B}_i)$. Now suppose that $\mathfrak{A} \models \neg\Phi$. Hence, there exists $i \in \{1, \dots, m\}$ such that $\mathfrak{A} \models \neg\Phi_i$. Then Duplicator wins the (ℓ, k) -game as follows. She starts by playing \mathfrak{B}_i . Then \mathfrak{A} homomorphically maps to \mathfrak{B}_i , and Duplicator can win the existential (ℓ, k) pebble game on $(\mathfrak{A}, \mathfrak{B}_i)$ by always playing along the homomorphism.

For the converse implication, suppose that 1. and 2. hold. Since $\llbracket \Phi \rrbracket$ is closed under homomorphisms, Corollary 18 implies that there are GSO sentences Φ_1, \dots, Φ_m and ω -categorical structures $\mathfrak{B}_1, \dots, \mathfrak{B}_m$ such that Φ is equivalent to $\Phi_1 \wedge \dots \wedge \Phi_m$ and $\llbracket \neg\Phi_i \rrbracket = \text{CSP}(\mathfrak{B}_i)$. By Theorem 21, for every $i \in \{1, \dots, m\}$ there exists a canonical Datalog program Π_i of width (ℓ, k) for $\llbracket \Phi_i \rrbracket$. Then Lemma 22 implies that there exists a Datalog program Π such that $\llbracket \Pi \rrbracket = \llbracket \Pi_1 \rrbracket \cap \dots \cap \llbracket \Pi_m \rrbracket$. Since each Π_i is sound for $\llbracket \Phi_i \rrbracket$, it follows that Π is sound for $\llbracket \Phi \rrbracket$. Hence, it suffices to show that if \mathfrak{A} is a finite τ -structure such that $\mathfrak{A} \models \Phi$, then Π derives goal on \mathfrak{A} . Since $\mathfrak{A} \models \Phi_i$ for all $i \in \{1, \dots, m\}$, the assumption implies that Spoiler wins the existential (ℓ, k) pebble game on $(\mathfrak{A}, \mathfrak{B}_i)$. By Theorem 21, it follows that Π_i derives goal on \mathfrak{A} . Hence, Π derives goal on \mathfrak{A} . \square

7 Conclusion and Open Problems

We provided a game-theoretic characterisation of those problems in Guarded Second-order Logic that are equivalent to a Datalog program. We also proved the existence of canonical Datalog programs for GSO sentences whose models are closed under homomorphisms. To prove these results, we showed that every class of finite τ -structures in GSO whose complement is closed under homomorphisms is a finite union of CSPs. We also showed that every CSP in GSO can be formulated as a CSP of an ω -categorical structure. These results also imply that the so-called universal-algebraic approach, which has eventually led to the classification of finite-domain CSPs in Datalog [3], can be applied to study problems that are simultaneously in Datalog and in GSO (also see [9]). Our results might also pave the way towards a syntactic characterisation of Datalog \cap GSO. We close with two open problems.

1. *Nested monadically defined queries (Nemodeq)* have been introduced by Rudolph and Krötzsch [24]; they prove that Nemodeq is contained both in MSO and in Datalog. We ask whether conversely, every problem in $\text{MSO} \cap \text{Datalog}$ is expressible as a Nemodeq.
2. Is every CSP of a reduct of a finitely bounded homogeneous structure in GSO?

References

1. Hajnal Andr eka, Istv an N emeti, and Johan van Benthem. Modal languages and bounded fragments of predicate logic. *J. Philos. Log.*, 27(3):217–274, 1998. URL: <https://doi.org/10.1023/A:1004275029985>, doi:10.1023/A:1004275029985.
2. Albert Atserias, Andrei A. Bulatov, and Anuj Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410(18):1666–1683, 2009.
3. Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *Journal of the ACM*, 61(1):3:1–3:19, 2014.
4. Achim Blumensath. Monadic second-order logic. Lecture Notes, 2020.
5. Manuel Bodirsky. Complexity of infinite-domain constraint satisfaction. Submitted for publication in the LNL Series, Cambridge University Press, 2020.
6. Manuel Bodirsky and V ictor Dalmau. Datalog and constraint satisfaction with infinite templates. *Journal on Computer and System Sciences*, 79:79–100, 2013. A preliminary version appeared in the proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS’05).
7. Manuel Bodirsky and Martin Grohe. Non-dichotomies in constraint satisfaction complexity. In Luca Aceto, Ivan Damgard, Leslie Ann Goldberg, Magn us M. Halld orsson, Anna Ing olfsd ottir, and Igor Walukiewicz, editors, *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science, pages 184–196. Springer Verlag, July 2008.
8. Manuel Bodirsky, Martin Hils, and Barnaby Martin. On the scope of the universal-algebraic approach to constraint satisfaction. In *Proceedings of the Symposium on Logic in Computer Science (LICS)*, pages 90–99. IEEE Computer Society, July 2010.

9. Manuel Bodirsky, Wied Pakusa, and Jakub Rydval. Temporal constraint satisfaction problems in fixed-point logic. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 237–251. ACM, 2020. URL: <https://doi.org/10.1145/3373718.3394750>, doi:10.1145/3373718.3394750.
10. Manuel Bodirsky, Michael Pinsker, and András Pongrácz. Projective clone homomorphisms. *Journal of Symbolic Logic*, 2019. To appear. doi:10.1017/jsl.2019.23.
11. Pierre Bourhis, Markus Krötzsch, and Sebastian Rudolph. Reasonable highly expressive query languages - IJCAI-15 distinguished paper (honorary mention). In Qiang Yang and Michael J. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2826–2832. AAAI Press, 2015.
12. Samuel Braulfeld. Towards the undecidability of atomicity for permutation classes via the undecidability of joint embedding for hereditary graph classes, 2019. Preprint available at arXiv:1903.11932.
13. Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17*, pages 319–330, 2017.
14. Gregory Cherlin. Two problems on homogeneous structures, revisited. *Model Theoretic Methods in Finite Combinatorics, Contemporary Mathematics*, 558, 2011.
15. Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Cambridge University Press, 40 W. 20 St. New York, NY, United States, 2012.
16. Victor Dalmau, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint satisfaction, bounded treewidth, and finite-variable logics. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP)*, pages 310–326, 2002.
17. Michael Elberfeld, Martin Grohe, and Till Tantau. Where first-order and monadic second-order logic coincide. *CoRR*, abs/1204.6291, 2012. URL: <http://arxiv.org/abs/1204.6291>, arXiv:1204.6291.
18. Tomás Feder and Moshe Y. 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.
19. Erich Grädel. Description logics and guarded fragments of first order logic. In Enrico Franconi, Giuseppe De Giacomo, Robert M. MacGregor, Werner Nutt, and Christopher A. Welty, editors, *Proceedings of the 1998 International Workshop on Description Logics (DL'98), IRST, Povo - Trento, Italy, June 6-8, 1998*, volume 11 of *CEUR Workshop Proceedings*. CEUR-WS.org, 1998. URL: <http://ceur-ws.org/Vol-11/graedel.ps>.
20. Erich Grädel, Colin Hirsch, and Martin Otto. Back and forth between guarded and modal logics. *ACM Trans. Comput. Log.*, 3(3):418–463, 2002.
21. Wilfrid Hodges. *A shorter model theory*. Cambridge University Press, Cambridge, 1997.
22. Phokion G. Kolaitis and Moshe Y. Vardi. On the expressive power of Datalog: Tools and a case study. *Journal of Computer and System Sciences*, 51(1):110–134, 1995.
23. Leonid Libkin. *Elements of Finite Model Theory*. Springer, 2004.
24. Sebastian Rudolph and Markus Krötzsch. Flag & check: Data access with monadically defined queries. In *Proc. 32nd Symposium on Principles of Database Systems (PODS'13)*, pages 151–162. ACM, June 2013. doi:10.1145/2463664.2465227.
25. Dmitriy N. Zhuk. A proof of CSP dichotomy conjecture. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17*, pages 331–342, 2017. <https://arxiv.org/abs/1704.01914>.