Datalog-Expressibility for Monadic and Guarded Second-Order Logic

MANUEL BODIRSKY, TU Dresden, Institut für Algebra, Germany
SIMON KNÄUER, TU Dresden, Institut für Algebra, Germany
SEBASTIAN RUDOLPH, TU Dresden, Computational Logic Group, Germany

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 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) in the sense of Feder and Vardi. The same characterisations also hold for Guarded Second-order Logic (GSO), which properly extends MSO. To prove our results, we show that every class C in GSO whose complement is closed under homomorphisms is a finite union of constraint satisfaction problems (CSPs) of ω-categorical structures. The intersection of MSO and Datalog is known to contain the class of *nested monadically defined queries (Nemodeq)*; likewise, we show that the intersection of GSO and Datalog contains all problems that can be expressed by the more expressive language of *nested guarded queries (GQ$^+$)*. Yet, by exploiting our results, we can show that neither of the two query languages can serve as a characterization, as we exhibit a CSP whose complement corresponds to a query in the intersection of MSO and Datalog that is not expressible in GQ$^+$.

CCS Concepts: • **Theory of computation → Finite Model Theory**.

Additional Key Words and Phrases: Monadic Second-order Logic, Guarded Second-order Logic, Datalog, constraint satisfaction, homomorphism-closed, conjunctive query, primitive positive formula, pebble game, ω-categoricity

CONTENTS

1 INTRODUCTION

Monadic Second-order Logic (MSO) is a logic of great importance in theoretical computer science. While it significantly exceeds the expressive capabilities of First-order Logic (allowing for expressing crucial structural properties like reachability or connectedness), it is still computationally reasonably well-behaved: By Büchi's theorem (see, e.g., [40]), the formal languages definable in MSO are precisely the regular ones; by Courcelle's theorem [27], MSO sentences can be evaluated in polynomial time on classes of structures whose treewidth is bounded by a constant. The latter result even holds for the more expressive logic of *Guarded Second-order Logic (GSO)* [31, 36], which extends First-order Logic by second-order quantifiers over *guarded relations*.

Another fundamental formalism in theoretical computer science, which is particularly heavily studied in database theory and logic-based knowledge representation, is *Datalog* (see, e.g., [29, 40]). Every fixed Datalog program can be evaluated on given finite structures in polynomial time. If a linear order is present and if the available relations are closed under complement, Datalog can even express *all* polytime-computable queries [1]. That is, like MSO and GSO, Datalog strikes a good balance between expressivity and good mathematical and computational properties.

Neither of the two formalisms subsumes the other in terms of what can be expressed. Yet, in various scenarios, we are interested in simultaneously having the good computational properties of expressibility in Datalog *and* having the good computational properties of expressibility in MSO or GSO. A wide variety of popular query formalisms (among them (unions of) conjunctive queries, (2-way conjunctive) regular path queries, monadic Datalog, the recently introduced almost monadic Datalog [44], guarded Datalog, monadically defined queries, or nested monadically defined queries) are known to be both in Datalog and GSO [43]. Query languages expressible both in Datalog and MSO have been shown to warrant decidable query entailment over logical theories exhibiting universal models of finite treewidth and cliquewidth [34, 35]. Also, all these formalisms have favourable properties when it comes to static analysis, most notably decidable query containment [21, 43]. Note that on the contrary, query containment in unrestricted Datalog is undecidable, as is query containment in unrestricted MSO / GSO[1]. The same holds for query entailment under logical theories, even if the latter are extremely simple. So it is really the interplay of the restrictions imposed by both formalisms that is required to ensure decidability of central tasks in databases and knowledge representation. This makes the semantic intersection of Datalog and MSO / GSO so interesting and worthwhile investigating.

In this article, we investigate two questions, which will turn out to be closely related:

(1) Which classes of finite structures are expressible both in MSO/GSO and in Datalog?
(2) Which *constraint satisfaction problems (CSPs)* can be expressed in MSO/GSO?

Indeed, as our investigation reveals, the versatile and vibrant discipline of constraint satisfaction problems offers many suitable notions and tools for our endeavour to understand "MSO/GSO ∩ Datalog". While this might come as a surprise at the first glance, interesting correspondences connecting CSPs with expressivity characterizations of logical formalisms from database theory and knowledge representation have been observed before [6, 33].

We recall that, 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 $\mathfrak{B}$. It is well known that, whenever $\mathfrak{B}$ is finite, its constraint satisfaction problem can already be expressed in a small fragment of MSO, called monotone monadic SNP (MMSNP, [32]). Yet, this correspondence does not hold for infinite $\mathfrak{B}$.

Example 1.1. *The constraint satisfaction problem for the structure* $(\mathbb{Q}; <)$*, which is the class of all finite acyclic digraphs* $(V; E)$*, cannot be expressed in MMSNP [10]. It can, however, be expressed in MSO over the class of all finite digraphs by the sentence*

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

To see this, note that if $(V; E)$ *is a finite digraph such that there exists a non-empty* $X \subseteq V$ *such that for every* $x \in X$ *there is a* $y \in Z$ *with* $E(x, y)$*, then* X *contains a directed cycle, and hence has no homomorphism to* $(\mathbb{Q}; <)$*. Conversely, if* $(V; E)$ *is a finite digraph and contains no directed cycle, then every non-empty subset of* V *must contain a sink, i.e., a vertex* x *with no outgoing* E*-edges, and hence satisfies the given sentence.* △

The class of CSPs of arbitrary infinite structures $\mathfrak{B}$ is quite large; the following is easy to see (see, e.g., [10, Lemma 1.1.8]).

Remark 1.2. *For every finite relational signature* τ*, a class* $\mathcal{D}$ *of finite* τ*-structures is the CSP of some countably infinite structure if and only if*

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

A class of finite relational structures that satisfies these two conditions is simply called a CSP.

---

[1]This follows from Trakhtenbrot's theorem which states that satisfiability of first-order logic in the finite is undecidable.

The second condition in Remark 1.2, which is sometimes also referred to as *closure under inverse homomorphisms*, can be equivalently rephrased by requiring that the *complement*[2] of $\mathcal{D}$ is *closed under homomorphisms*, where a class C is closed under homomorphisms if for any structure $\mathfrak{A} \in C$ that maps homomorphically to some $\mathfrak{C}$ we have $\mathfrak{C} \in C$. Examples of classes of structures that are closed under homomorphisms naturally arise from Datalog. We say that a class C of finite τ-structures *is definable in Datalog* (or shorter: *is in Datalog*)[3] if there exists a Datalog program Π with a distinguished nullary predicate goal such that Π derives goal on a finite τ-structure if and only if the structure is in C; in this case, we may denote C by $[\![\Pi]\!]$. While each class of τ-structures that is definable in Datalog is closed under homomorphisms, not every such class corresponds to the complement of a CSP.

Example 1.3. *Consider, for unary predicates R and B, the class $C_{R,B}$ of finite $\{R, B\}$-structures $\mathfrak{A}$ such that $R^{\mathfrak{A}}$ is empty or $B^{\mathfrak{A}}$ is empty. Clearly, $C_{R,B}$ is not closed under disjoint unions, so it cannot correspond to the CSP of any structure. Yet, it is easy to see that a finite structure is in $C_{R,B}$ if and only if the Datalog program consisting of just the one rule*

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

does not derive goal *on that structure.* △

A noteworthy subclass of all CSPs are the CSPs of structures $\mathfrak{B}$ that are countably infinite and *ω-categorical*. A structure $\mathfrak{B}$ is *ω-categorical* if all countable models of $\mathfrak{B}$'s first-order theory are isomorphic to $\mathfrak{B}$. A well-known example of an ω-categorical structure is $(\mathbb{Q}; <)$, a result going back to Cantor [23]. We would like to mention that constraint satisfaction problems of ω-categorical structures can be evaluated in polynomial time on any class of structures whose treewidth is bounded by some constant $k \in \mathbb{N}$, by a result of Bodirsky and Dalmau [14], using Datalog programs; however, the concept of bounded treewidth is not needed in the present article.

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 [32]. The special case of $\ell = 1$ is typically referred to as *monadic Datalog*. The parameters ℓ and k are important in theory, but they also bear some practical relevance: When evaluating Π on a given structure $\mathfrak{A}$ with domain A, the memory needed is bounded by $O(|A|^{\ell})$ and the number of computation steps by $O(|A|^k)$. The polynomial-time algorithm for instances of an ω-categorical CSP of treewidth at most k 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 C if $[\![\Pi]\!] \subseteq C$. Bodirsky and Dalmau showed that if C is the complement of the CSP of an ω-categorical τ-structure $\mathfrak{B}$ then there exists, for any $\ell, k \in \mathbb{N}$, a Datalog program Π of width (ℓ, k) such that

- Π is sound for C, and
- $[\![\Pi']\!] \subseteq [\![\Pi]\!]$ for every Datalog program Π' of width (ℓ, k) which is sound for C.

This Π (which is unique up to immaterial syntactic variations) is then referred to as the *canonical Datalog program of width (ℓ, k) for C.*

Interestingly and conveniently, there is a known game-theoretic characterisation capturing whether the canonical Datalog program of width (ℓ, k) for C derives goal on a given τ-structure $\mathfrak{A}$ [14]. This characterisation is based on the existential pebble game from finite model theory, which is played on a pair $(\mathfrak{A}, \mathfrak{B})$ of structures. In more detail, the *existential (ℓ, k)-pebble game* is played by two players, called *Spoiler* and *Duplicator* (see, e.g., [28, 32, 38]). Spoiler starts by placing k pebbles on elements $a_1, \ldots, a_k$ of $\mathfrak{A}$, and Duplicator responds by placing k pebbles $b_1, \ldots, b_k$ on $\mathfrak{B}$. If the map that sends $a_1, \ldots, a_k$ to $b_1, \ldots, b_k$ 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 $\mathfrak{A}$, 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 of width (ℓ, k) for CSP$(\mathfrak{B})$ derives goal on $\mathfrak{A}$ (Theorem 4.1). This connection played an essential role in proving Datalog inexpressibility results, for example for the class of finite-domain CSPs [3] (leading to a complete classification of those finite structures $\mathfrak{B}$ such that the complement of CSP$(\mathfrak{B})$ can be expressed in Datalog [5]).

Results and Consequences

In this article, we prove 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 (Theorem 3.15; an analogous statement holds for MSO). Moreover, every CSP in GSO is the CSP of a countable ω-categorical structure (Corollary 3.8). We highlight that this result elegantly generalises known results:

- our result generalises the fact that every CSP in FO is the CSP of a countable ω-categorical structure, which can be seen from combining Rossmann's theorem [42] with a generalisation of the theorem of Cherlin, Shelah, and Shi [26] from graphs to general relational structures.
- Our result also generalises the fact that every CSP in the logic MMSNP (for *monotone monadic strict NP* [32]) is the CSP of a countable ω-categorical structure [14].

It follows that every class of finite structures that can be expressed both in GSO and in Datalog is an intersection of *finitely many* complemented CSPs of ω-categorical structures. In contrast, it is *not* generally true that a Datalog program describes a finite intersection of complements of CSPs (we present a counterexample in Example 3.17).

Next, we use the connection between GSO and CSPs to present a characterisation of those GSO sentences Φ that are over finite structures equivalent to a Datalog program (Section 4). 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 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 CSP$(\mathfrak{B}) \cap C = \emptyset$.
- The game then continues as the existential (ℓ, k)-pebble game played by Spoiler and Duplicator on $(\mathfrak{A}, \mathfrak{B})$, as described above.

We use results from [14] to show that a GSO sentence Φ is over finite structures equivalent to a Datalog program of width (ℓ, k) if and only if

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

We also show that for every class of finite models C which is closed under homomorphisms and expressible in GSO, and for all $\ell, k \in \mathbb{N}$, there exists a canonical Datalog program Π of width (ℓ, k) for C (Theorem 4.6).

Our next series of results concern the most expressive syntactically defined formalism known from the literature that is contained in both Datalog and MSO, namely *nested monadically defined queries* [43], as well as the most expressive syntactically defined formalism known from the literature that is contained in both Datalog and GSO, namely *nested guarded queries* [21].

We prove that there are problems in the intersection of Datalog and GSO that cannot be expressed as a nested guarded query (Section 5). To prove this result, we introduce a modified version of the existential pebble game, which we call the *nested guarded pebble game*, and which captures precisely the expressiveness of nested guarded queries (Theorem 5.13). We also present an example of a problem which lies in the intersection of Datalog and

MSO that cannot be expressed by nested monadically defined queries (and not even by nested guarded queries; Corollary 5.19).

Finally, we provide evidence that the class of CSPs that can be expressed in MSO is *not* contained in the intensively studied class of CSPs for reducts of finitely bounded homogeneous structures: in Section 6 we present an example of a CSP which is expressible in MSO and coNP-complete, and hence not the CSP of a reduct of a finitely bounded homogeneous structure, unless NP=coNP (Proposition 6.3). As an illustration of the results of Section 4, we also prove that the given MSO sentence is not equivalent to a Datalog program.

Some of the results of this article until Section 4 have been announced in a conference paper with the title "Datalog-Expressibility for Monadic and Guarded Second-Order Logic" in the proceedings of ICALP'21 [18]; the results in Section 5 about nested monadic and nested guarded queries were not yet present in the conference version.

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 $ar(R)$ for its arity. If $\mathfrak{A}$ is a τ-structure we use the corresponding roman capital letter A to denote the domain of $\mathfrak{A}$; the domains of structures are assumed to be non-empty. If $R \in \tau$, then $R^{\mathfrak{A}} \subseteq A^{ar(R)}$ denotes the corresponding relation of $\mathfrak{A}$.

A *primitive positive τ-formula* (in database theory also referred to as *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

$$\exists x_1, \ldots, x_n (\psi_1 \wedge \cdots \wedge \psi_m)$$

where $\psi_1, \ldots, \psi_m$ are atomic τ-formulas, i.e., formulas built from relation symbols in τ or equality. An *existential positive τ-formula* is a first-order τ-formula without negation and universal quantification. We write $\psi(x_1 \ldots, x_n)$ if the free variables of ψ are from $x_1, \ldots, x_n$. If $\mathfrak{A}$ is a τ-structure and $\psi(x_1, \ldots, x_n)$ is a τ-formula, then the relation

$$R := \{(a_1, \ldots, a_n) \mid \mathfrak{A} \models \psi(a_1, \ldots, a_n)\}$$

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 $\phi(x_1, \ldots, x_n)$ and $\psi(x_1, \ldots, x_n)$ are *equivalent (over finite structures)* if for all (finite) τ-structures $\mathfrak{A}$ and all $a_1, \ldots, a_n \in A$ we have

$$\mathfrak{A} \models \phi(a_1, \ldots, a_n) \Leftrightarrow \mathfrak{A} \models \psi(a_1, \ldots, a_n).$$

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). If ϕ is a primitive positive τ-formula without equality for a relational signature τ, then the *canonical database* is the τ-structure $\mathfrak{A}$ whose domain consists of all the variables of ϕ, and where $R^{\mathfrak{A}}$, for $R \in \tau$ of arity k, consists of all tuples $(x_1, \ldots, x_k)$ for which ϕ contains the conjunct $R(x_1, \ldots, x_k)$.

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 let $[\![\phi]\!]$ denote 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. $\mathfrak{A}$ is then also called an *expansion* of $\mathfrak{A}'$.

2.1 Datalog

In this section we consider a finite set τ of constant and relation symbols, the latter being referred to 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

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

where ψ_0 (called the *head* of the rule) is an atomic ρ-formula and $\psi_1, \ldots, \psi_n$ (jointly referred to as the *body* of the rule) are atomic $(\rho \cup \tau)$-formulas; we also assume that all rules are *safe*, i.e., that every variable that appears in the head also appears in the body. If $\mathfrak{A}$ is a τ-structure, and Π is a Datalog program with EDBs from 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

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

for each rule $\psi_0 :- \psi_1, \ldots, \psi_n$. If $\mathfrak{A}_1$ and $\mathfrak{A}_2$ are two $(\rho \cup \tau)$-structures with the same domain A, we let $\mathfrak{A}_1 \cap \mathfrak{A}_2$ denote the $(\rho \cup \tau)$-structure with domain A such that

- $R^{\mathfrak{A}_1 \cap \mathfrak{A}_2} := R^{\mathfrak{A}_1} \cap R^{\mathfrak{A}_2}$, and
- $c^{\mathfrak{A}_1 \cap \mathfrak{A}_2} = c^{\mathfrak{A}_1} = c^{\mathfrak{A}_2}$.

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 $\Pi(\mathfrak{A})$.

Remark 2.1. *There is an equivalent 'operational' definition of the semantics of Datalog, and in particular of $\Pi(\mathfrak{A})$; roughly speaking, we compute $R^{\Pi(\mathfrak{A})}$ for $R \in \rho$ 'bottom up', starting from the empty relation, and adding tuples that must be contained in all fixed points, until we reach a (the) smallest fixed point (for details see, e.g., Libkin's book [40, Section 10.5]; an explicit treatment of the equivalence can e.g. be found in [10, Theorem 8.1.6]). From this equivalent description of $\Pi(\mathfrak{A})$ it is apparent that if $\mathfrak{A}$ is a finite structure then $\Pi(\mathfrak{A})$ can be computed in polynomial time in the size of $\mathfrak{A}$.* △

If $R \in \rho$, we also say that Π *defines* $R^{\Pi(\mathfrak{A})}$ *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 goal$^{\Pi(\mathfrak{A})} = \{()\}$. The class C of finite τ-structures $\mathfrak{A}$ such that Π derives goal on $\mathfrak{A}$ is called *the class of finite τ-structures defined by* Π, and denoted by $[\![\Pi]\!]$. Note that this class C is definable 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 and Monadic 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 $[\![\Phi]\!]$ 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 $\exists x(X(x) \wedge \psi)$.

Monadic second-order logic (MSO) is surprisingly powerful for expressing CSPs, which we illustrate with the following example.

Example 2.2. *Let $E := \{(x, x) \mid x \in \mathbb{N}\}$ be the binary equality relation on $\mathbb{N}$ and let $D := \{(x, y) \in \mathbb{N}^2 \mid x \neq y\}$ be the binary disequality relation on $\mathbb{N}$. The problem $\mathrm{CSP}(\mathbb{N}; E, D)$ is expressible in MSO. To see this, let ϕ_X, for a unary relation symbol X, be the following $\{E, D, X\}$-sentence.*

$$\exists x.X(x) \wedge \forall x, y\big(X(x) \wedge E(x, y) \Rightarrow X(y)\big) \wedge \big((X(x) \wedge E(y, x) \Rightarrow X(y)\big)$$

For unary relation symbols X and Y we write $Y \subsetneq X$ as a shortcut for

$$\forall x(Y(x) \Rightarrow X(x)) \land \exists x(X(x) \land \neg Y(x)).$$

Let ψ_X be the sentence that states that X is a smallest set which satisfies ϕ_X:

$$\phi_X \land \forall Y(Y \subsetneq X \Rightarrow \neg \phi_Y)$$

Note that if $\mathfrak{A}$ is an $\{E, D\}$-structure and $X \subseteq A$ satisfies ψ_X, then any homomorphism from $\mathfrak{A}$ to $(\mathbb{N}; E, D)$ must be constant on X. Also note that if $X, Y \subseteq A$ satisfy ψ_X and ψ_Y and $X \cap Y \neq \emptyset$, then $Z := X \cap Y$ satisfies ψ_Z. Hence, for every element $a \in A$ there exists a unique smallest set $X = X_a$ that satisfies ψ_X and contains a.

We claim that the CSP above can be expressed by

$$\forall X(\psi_X \Rightarrow \forall x, y(D(x, y) \Rightarrow (\neg X(x) \lor \neg X(y)))). \tag{1}$$

Indeed, suppose that $\mathfrak{A}$ is an $\{E, D\}$-structure and let $X \subseteq A$ be smallest with the property that it satisfies ϕ_X. If there are $x, y \in X$ such that $(x, y) \in D^{\mathfrak{A}}$ then clearly there is no homomorphism from $\mathfrak{A}$ to $(\mathbb{N}; E, D)$. On the other hand, if $\mathfrak{A}$ satisfies (1), then any map $h \colon A \to \mathbb{N}$ such that $h^{-1}(h(a)) = X_a$ for all $x \in A$ is a homomorphism from $\mathfrak{A}$ to $(\mathbb{N}; E, D)$. Clearly, there exists such a map h, which proves the claim. △

2.3 Guarded Second-Order Logic

Guarded Second-order Logic (GSO), introduced by Grädel, Hirsch, and Otto [36], is the extension of *guarded first-order logic* by second-order quantifiers. Guarded (first-order) τ-formulas are defined inductively by the following rules [2]:

(1) all atomic τ-formulas are guarded τ-formulas;
(2) if ϕ and ψ are guarded τ-formulas, then so are $\phi \land \psi$, $\phi \lor \psi$, 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 $\exists \bar{y}(\alpha(\bar{x}, \bar{y}) \land \psi(\bar{x}, \bar{y}))$ 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 [36]. Let $\mathfrak{B}$ be a τ-structure. Then a tuple $(t_1, \ldots, t_n) \in B^n$ is called *guarded in $\mathfrak{B}$* if there exists an atomic τ-formula ϕ and $b_1, \ldots, b_k$ such that $\mathfrak{B} \models \phi(b_1, \ldots, b_k)$ and $\{t_1, \ldots, t_n\} \subseteq \{b_1, \ldots, b_k\}$. A relation $R \subseteq B^n$ is called *guarded* if all tuples in R are guarded. Note that (for $n = 1$) every element of B is guarded (because of the valid atomic formula $x = x$), and consequently all unary relations are guarded. If Ψ is an arbitrary second-order sentence, we say that a structure $\mathfrak{B}$ *satisfies Ψ with guarded semantics*, in symbols $\mathfrak{B} \models_g \Psi$, 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.

Proposition 2.3 (see Proposition 3.9 in [36]). *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 many Datalog programs that are equivalent to a GSO sentence, but not to an MSO sentence. However, since MSO is surprisingly expressive (see Example 2.2) it may be quite challenging to *prove* that a specific problem cannot be expressed in MSO. For example, it is an open problem whether $\mathrm{CSP}(\mathbb{Q}; B)$ is expressible in MSO, where

$$B = \{(x, y, z) \in \mathbb{Q}^3 \mid x < y < z \lor z < y < x\}$$

Structure $\mathfrak{B}$

Structure $\mathfrak{A}$
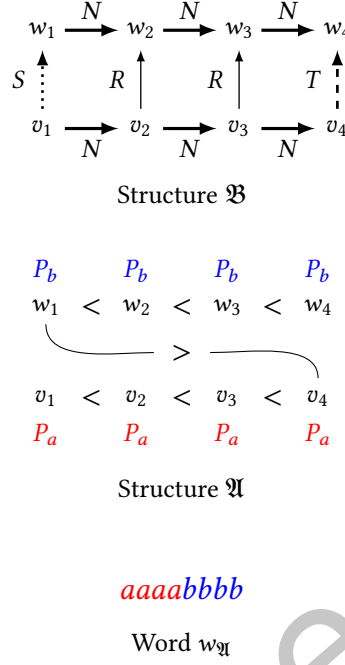
aaaabbbb

Word $w_{\mathfrak{A}}$

Fig. 1. An example of an $\{S, T, R, N\}$-structure $\mathfrak{B}$ in the class C of Proposition 2.4.

is the so-called *betweenness relation* [27]. One such MSO-inexpressibility result is part of the following proposition whose proof is based on a variant of an example of a Datalog query in GSO given in [21] (Example 2).

Proposition 2.4. *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 C be the class of finite τ-structures such that the following Datalog program with one binary IDB U derives goal.

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

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. Note that the Datalog program derives goal on a given finite structure if and only if a 'ladder' structure as shown on the left of Figure 1 homomorphically maps to the structure.

To show that C is not MSO-expressible, suppose toward a contradiction suppose for contradiction that there exists an MSO sentence Φ such that $[\![\Phi]\!] = C$. We use Φ to construct an MSO sentence Ψ which holds on a finite word $w \in \{a, b\}^*$ (represented as a structure with the signature $\{P_a, P_b, <\}$ in the usual way [40]) if and only if $w \in \{a^n b^n \mid n \geq 1\}$; this contradicts the theorem of Büchi-Elgot-Trakhtenbrot (see, e.g., [40]), which states that a language of finite words can be expressed in MSO if and only if the language is regular. Let Ψ_1 be the MSO sentence obtained from Φ by replacing all subformulas of Φ of the form

- $S(x, y)$ by a formula $\phi_S(x, y)$ that states that x is the smallest element with respect to $<$, that $P_b(y)$ holds, and that there is no $z < y$ in P_b;

- $T(x, y)$ by a formula $\phi_T(x, y)$ that states that $P_a(x)$ holds, 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 w.r.t. $<$.

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 $P_a(x)$.

Claim. 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 with the same domain as $\mathfrak{A}$ 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 $\mathfrak{B}$. Note that $\mathfrak{B} \models \Phi$ if and only if $\mathfrak{A} \models \Psi_1$. 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 $v_1, \ldots, v_n, w_1, \ldots, w_n \in A$ be such that $\{v_1, \ldots, v_n\} = P_a^{\mathfrak{A}}$ and $\{w_1, \ldots, w_n\} = P_b^{\mathfrak{A}}$ such that for all $i, j \in \{1, \ldots, n\}$, if $i < j$ then $v_i <^{\mathfrak{A}} v_j$ and $w_i <^{\mathfrak{A}} w_j$. Then

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

It follows that $\mathfrak{B}$ satisfies Φ and therefore $\mathfrak{A} \models \Psi_1$.

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 $v_1, \ldots, v_n, w_1, \ldots, w_n \in A$ such that $\mathfrak{B}$ satisfies (2). We first prove that $P_a^{\mathfrak{A}} = \{v_1, \ldots, v_n\}$ 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 $\mathfrak{B} \models N(v_1, v_2), \ldots, N(v_{n-1}, v_n)$ we have that $v_1 < v_2 < \cdots < v_{n-1} < v_n$ holds in $\mathfrak{A}$ and it also follows that $|P_a^{\mathfrak{A}}| \geq n$. 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 $x \in P_a^{\mathfrak{A}} \setminus \{v_1, \ldots, v_n\}$; 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 $x \notin \{v_1, \ldots, v_n\}$. 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 $y \in P_a^{\mathfrak{A}}$. By the maximal choice of x we get that $y = v_i$ for some $i \in \{1, \ldots, n\}$. But then $\phi_N(x, v_i)$ implies that $x \in \{v_1, \ldots, v_{n-1}\}$, a contradiction. Similarly, one can prove that $P_b^{\mathfrak{A}} = \{w_1, \ldots, w_n\}$ and that $|P_b^{\mathfrak{A}}| = n$. This implies that $w_{\mathfrak{A}} = a^n b^n$.

We finally have to prove that 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 is a unique element $y \in V \setminus \{v_n\}$ such that $N'(y, x)$ holds;
- for every $x \in V \setminus \{v_n\}$ there is a unique element $y \in V \setminus \{v_1\}$ such that $N'(x, y)$ holds;
- for every $x \in W \setminus \{w_1\}$ there is a unique element $y \in W \setminus \{w_n\}$ such that $N'(y, x)$ holds;
- for every $x \in W \setminus \{w_n\}$ there is 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 $v_1, \ldots, v_n, w_1, \ldots, w_n$ satisfying (2), which is the case if and only if $\mathfrak{B} \in C$, which can be seen from the inductive computation of the fixed point by the Datalog program, see Remark 2.1. □

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 (the complement of) a certain CSP is *in* GSO (MSO, Datalog).

3 HOMOMORPHISM-CLOSED GSO

In this section, 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 $\mathfrak{B}$. This result will be essential for our proof of the existence of canonical Datalog programs for GSO in Section 4. In our proof, we use an equivalent characterisation of the CSPs that can be formulated as the CSP of an ω-categorical structure from [17]; this characterisation will be recalled in Section 3.1.

Remark 3.1. *This remark is addressed to readers familiar with the universal-algebraic approach to CSPs (see [10] for an introduction), and can be skipped. Recall that whether the CSP of an ω-categorical structure $\mathfrak{B}$ is in Datalog only depends on the polymorphism clone of $\mathfrak{B}$ (see [10, Corollary 8.3.4] for an even stronger statement). Therefore, our result opens up the repertoire of methods from universal algebra for further studies of the intersection of GSO and Datalog.*

3.1 CSPs for Countably Categorical Structures

Given permutation group G on a set B, an *orbit* of G is a set of the form $\{g(b) \mid g \in G\}$ for some $b \in B$. 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 $\mathrm{Aut}(\mathfrak{B})$ of $\mathfrak{B}$ on B^n (see, e.g., [37]). It follows that if a structure is ω-categorical, then all of its reducts are ω-categorical as well [37].

There is a known condition that characterises classes of structures that are CSPs of ω-categorical structures. Let τ be a relational signature and let C be a class of finite τ-structures. Let Λ_n be the class of primitive positive τ-formulas with free variables $x_1, \ldots, x_n$ whose canonical database is in C. We define $\sim_n^C$ to be the equivalence relation on Λ_n such that $\phi_1 \sim_n^C \phi_2$ holds if for all primitive positive τ-formulas $\psi(x_1, \ldots, x_n)$ we have that $\phi_1(x_1, \ldots, x_n) \wedge \psi(x_1, \ldots, x_n)$ is satisfiable in a structure from C if and only if $\phi_2(x_1, \ldots, x_n) \wedge \psi(x_1, \ldots, x_n)$ is satisfiable in a structure from C.[4] The *index* of an equivalence relation is the number of its equivalence classes. If $\sim$ is an equivalence relation on a set A, and $a \in A$, then $[a]_\sim$ denotes the equivalence class of a; if the reference to $\sim$ is clear, we omit $\sim$ in this notation.

THEOREM 3.2 (BODIRSKY, HILS, MARTIN [17], THEOREM 4.27). *Let C be a constraint satisfaction problem. Then there is an ω-categorical structure $\mathfrak{B}$ such that $C = \mathrm{CSP}(\mathfrak{B})$ iff $\sim_n^C$ 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 $\mathrm{Aut}(\mathfrak{B})$ on B^n are primitively positively definable in $\mathfrak{B}$.*

Example 3.3. *The structure $\mathfrak{B}_1 := (\mathbb{Z}; <)$ is not ω-categorical. However, $\sim_n^{\mathrm{CSP}(\mathfrak{B}_1)}$ has finite index for all n, and indeed $\mathrm{CSP}(\mathbb{Z}; <) = \mathrm{CSP}(\mathbb{Q}; <)$ and $(\mathbb{Q}; <)$ is ω-categorical. On the other hand, for $\mathfrak{B}_2 := (\mathbb{Z}; Succ)$ we have that the index $\sim_2^{\mathrm{CSP}(\mathfrak{B}_2)}$ is infinite, and it follows that there is no ω-categorical structure $\mathfrak{B}$ such that $\mathrm{CSP}(\mathfrak{B}_2) = \mathrm{CSP}(\mathfrak{B})$; see [10].* △

Remark 3.4. *Since we make essential use of Theorem 3.2, it might be helpful for the reader to have the following idea about its proof. We use the equivalence relation $\sim_n^C$ to expand the structures in C by new relations; for the resulting class we can use Fraïssé's construction of homogeneous structures (Definition 15). The assumption that $\sim_n^C$ has finite*

---

[4]A closely related concept was considered for graph classes in parametrised complexity [30, Definition 6.7.5]; we thank Antoine Mottet for pointing this out to us.

index is the reason that the resulting structure is also ω-categorical. By taking a reduct to the original signature, we might lose homogeneity, but keep ω-categoricity, and obtain the desired structure $\mathfrak{B}$. △

3.2 Quantifier Rank

In order to construct ω-categorical structures for a given CSP in GSO, we need to verify the condition given in Theorem 3.2. In this context, 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 [7]. If $\mathfrak{A}$ and $\mathfrak{B}$ are τ-structures and $q \in \mathbb{N}$ we write $\mathfrak{A} \equiv_q^{\mathrm{GSO}} \mathfrak{B}$ if $\mathfrak{A}$ and $\mathfrak{B}$ satisfy the same GSO τ-sentences of quantifier rank at most q. The following lemma is a natural extension of a classical result of Fraïssé and Hintikka on first-order logic (see, e.g., [37]).

Lemma 3.5 (Proposition 3.4 in [8]). *Let $q \in \mathbb{N}$ and τ be a finite signature with relation and constant symbols. Then $\equiv_q^{\mathrm{GSO}}$ is an equivalence relation with finite index on the class of all finite τ-structures. Moreover, every class of $\equiv_q^{\mathrm{GSO}}$ 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 $\tau \cup \{c_1, \ldots, c_k\}$-structure expanding $\mathfrak{A}$ where $c_1, \ldots, c_k$ 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 $(\mathfrak{A}, \bar{a}) \equiv_q^{\mathrm{GSO}} (\mathfrak{B}, \bar{b})$, we implicitly assume that we have chosen the same constant symbols for $\bar{a}$ and for $\bar{b}$. The following lemma is a natural extension of classical results of Ehrenfeucht and Fraïssé on first-order logic (see, e.g., [37]).

Lemma 3.6 (Proposition 3.6 in [8]). *Let $q \in \mathbb{N}$ and let $\mathfrak{A}$ and $\mathfrak{B}$ be τ-structures. Then $\mathfrak{A} \equiv_{q+1}^{\mathrm{GSO}} \mathfrak{B}$ if and only if the following properties hold:*

- *(first-order forth) For every $a \in A$, there exists $b \in B$ such that $(\mathfrak{A}, a) \equiv_q^{\mathrm{GSO}} (\mathfrak{B}, b)$.*
- *(first-order back) For every $b \in B$, there exists $a \in A$ such that $(\mathfrak{A}, a) \equiv_q^{\mathrm{GSO}} (\mathfrak{B}, b)$.*
- *(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 $\mathfrak{A}' \equiv_q^{\mathrm{GSO}} \mathfrak{B}'$.*
- *(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 $\mathfrak{A}' \equiv_q^{\mathrm{GSO}} \mathfrak{B}'$.*

In the following, τ denotes a finite relational signature. Theorem 3.7 below states that the $\equiv_q^{\mathrm{GSO}}$-equivalence class of the union of two structures which might overlap on some elements that are named by constants can be recovered from the equivalence classes of its components.

Definition 2. *Let $\rho \coloneqq \{c_1, \ldots, c_n\}$ 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 $c \in \rho$;*
- *$\{c_1^{\mathfrak{A}}, \ldots, c_n^{\mathfrak{A}}\} = A \cap B = \{c_1^{\mathfrak{B}}, \ldots, c_n^{\mathfrak{B}}\}$.*

Given a $(\mathfrak{A}, \mathfrak{B}) \in \mathcal{D}_n$, we write $\mathfrak{A} \uplus \mathfrak{B}$ for the structure with domain $A \cup B$ such that $R^{\mathfrak{A} \uplus \mathfrak{B}} \coloneqq R^{\mathfrak{A}} \cup R^{\mathfrak{B}}$ 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 special case of the following theorem for $n = 0$ (i.e., for disjoint unions with no overlap) is Proposition 4.2 in [8].

Theorem 3.7. *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 $(\mathfrak{A}_1, \bar{a}_1) \equiv_q^{\mathrm{GSO}} (\mathfrak{A}_2, \bar{a}_2)$ and $(\mathfrak{B}_1, \bar{b}_1) \equiv_q^{\mathrm{GSO}} (\mathfrak{B}_2, \bar{b}_2)$. Then*

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

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 $(\mathfrak{A}_1, \bar{a}_1) \equiv^{\mathrm{GSO}}_{q+1} (\mathfrak{A}_2, \bar{a}_2)$ and $(\mathfrak{B}_1, \bar{b}_1) \equiv^{\mathrm{GSO}}_{q+1} (\mathfrak{B}_2, \bar{b}_2)$. By symmetry and Lemma 3.6 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 3.6, there exists $c_2 \in A_2$ such that $(\mathfrak{A}_1, \bar{a}_1, c_1) \equiv^{\mathrm{GSO}}_q (\mathfrak{A}_2, \bar{a}_2, c_2)$. By the inductive assumption, this implies that

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

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 3.6 there are expansions $\mathfrak{A}'_2$ of $\mathfrak{A}$ and $\mathfrak{B}'_2$ of $\mathfrak{B}_2$ by guarded relations such that $(\mathfrak{A}'_1, \bar{a}_1) \equiv^{\mathrm{GSO}}_q (\mathfrak{A}'_2, \bar{a}_2)$ and $(\mathfrak{B}'_1, \bar{b}_1) \equiv^{\mathrm{GSO}}_q (\mathfrak{B}'_2, \bar{b}_2)$. By the inductive assumption, this implies that $(\mathfrak{A}'_1 \uplus \mathfrak{B}'_1, \bar{a}_1, \bar{b}_1) \equiv^{\mathrm{GSO}}_q (\mathfrak{A}'_2 \uplus \mathfrak{B}'_2, \bar{a}_2, \bar{b}_2)$, which completes the proof of (second-order forth). □

Corollary 3.8. *Let C be a CSP that can be expressed in GSO. Then there exists a countable ω-categorical structure $\mathfrak{B}$ such that $C = \mathrm{CSP}(\mathfrak{B})$.*

Proof. Let τ be the signature of C, and let Φ be a GSO τ-formula with quantifier rank q such that $C = [\![\Phi]\!]$. By Theorem 3.2 it suffices to show that the equivalence relation $\sim^C_n$ has finite index for every $n \in \mathbb{N}$. Let $\rho \coloneqq \{c_1, \ldots, c_n\}$ be a set of new constant symbols. By Lemma 3.5, there exists an $m \in \mathbb{N}$ such that $\equiv^{\mathrm{GSO}}_q$ has m equivalence classes on $(\tau \cup \rho)$-structures. If $\phi(x_1, \ldots, x_n)$ is a primitive positive τ-formula, then define $\mathfrak{S}_\phi$ to be the following $(\tau \cup \rho)$-structure:

- the elements are the equivalence classes of the smallest equivalence relation on the variables of ϕ that contains all pairs x, y such that ϕ contains the conjunct $x = y$, and
- $(C_1, \ldots, C_n) \in R^{\mathfrak{S}}$ for $R \in \tau$ if and only if there are $y_1 \in C_1, \ldots, y_n \in C_n$ such that $R(y_1, \ldots, y_n)$ is a conjunct of ϕ;
- finally, we set $c_i^{\mathfrak{S}_\phi} \coloneqq [x_i]$ for all $i \in \{1, \ldots, n\}$.

We claim that if $\mathfrak{S}_\phi \equiv^{\mathrm{GSO}}_q \mathfrak{S}_\psi$, then $\phi \sim^C_n \psi$. Let $\theta(x_1, \ldots, x_n)$ 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 $\mathfrak{S}_\phi \equiv^{\mathrm{GSO}}_q \mathfrak{S}_\psi$ and $\mathfrak{S}_\theta \equiv^{\mathrm{GSO}}_q \mathfrak{S}_\theta$, we have $\mathfrak{S}_\phi \uplus \mathfrak{S}_\theta \equiv^{\mathrm{GSO}}_q \mathfrak{S}_\psi \uplus \mathfrak{S}_\theta$ by Theorem 3.7. Now suppose that $\phi \wedge \theta$ is satisfiable in a model of Φ. This is the case if and only if $\mathfrak{S}_\phi \uplus \mathfrak{S}_\theta$ satisfies Φ, which in turn implies that $\mathfrak{S}_\psi \uplus \mathfrak{S}_\theta$ satisfies Φ since Φ has quantifier rank q. This in turn is the case if and only if $\psi \wedge \theta$ is satisfiable in a model of Φ, which proves the claim.

The claim implies that $\sim^C_n$ has at most m equivalence classes, concluding the proof. □

Example 3.9. *Let Φ be the following MSO sentence Φ.*

$$\forall X \big((\exists x. X(x)) \Rightarrow \exists x, y \in X \, \forall z \in X (\neg E(x, z) \vee \neg E(y, z)) \big)$$
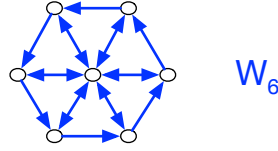
Fig. 2. An illustration of the digraph W_6 from Example 3.9.

Note that a finite digraph does not satisfy Φ if it has a non-empty subgraph with the following extension property: for any $x, y \in X$ there exists $z \in X$ such that $E(x, z)$ and $E(y, z)$.

It is easy to see that $\llbracket\Phi\rrbracket$ is closed under disjoint unions and that its complement is closed under homomorphisms. Corollary 3.8 implies that there exists a countable ω-categorical structure with $\mathrm{CSP}(\mathfrak{B}) = \llbracket\Phi\rrbracket$. △

Remark 3.10. We would like to point out that the existence of an ω-categorical structure $\mathfrak{B}$ with the properties from Example 3.9 does not simply follow from the theorem of Cherlin, Shelah, and Shi [26], which states that for every finite set of finite connected structures $\mathcal{F}$ there exists an ω-categorical structure $\mathfrak{C}$ such that a finite structure $\mathfrak{A}$ homomorphically maps to $\mathfrak{C}$ if and only if there is no homomorphism from a structure in $\mathcal{F}$ into $\mathfrak{A}$.

Formally, we claim that there is no finite set $\mathcal{F}$ of finite structures $\mathcal{F}$ such that a finite structure $\mathfrak{A}$ has a homomorphism to $\mathfrak{B}$ if and only if no structure from $\mathcal{F}$ homomorphically maps to $\mathfrak{A}$. To see this, first note that the following graphs W_i, for $i \geq 2$, satisfy the mentioned extension property: the vertex set of W_i is $\{0, 1, \ldots, i\}$, there is an edge from p to q, for $p, q \in \{0, \ldots, i-1\}$, if $p - q = 1 \mod i$, and additionally we have the edges (p, i) and (i, p) for every $p \in \{0, \ldots, i-1\}$. See Figure 2 for an illustration of W_6. Note that for every $i \geq 2$ the graph W_i is a core, and that they are pairwise homomorphically incomparable. Moreover, every subgraph of W_i does not satisfy the extension property. △

3.3 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 in GSO (Theorem 3.15); the statement announced at the beginning of Section 3 then follows (Corollary 3.16). Throughout this section, let τ be a relational signature and let C be a non-empty class of finite τ-structures whose complement is closed under homomorphisms. In particular, C contains the structure $\mathfrak{I}$ with only one element where all relations are empty.

Definition 3. Let $\sim$ be the equivalence relation defined on C by letting $\mathfrak{A} \sim \mathfrak{B}$ if for every $\mathfrak{C} \in C$ we have $\mathfrak{A} \uplus \mathfrak{C} \in C$ if and only if $\mathfrak{B} \uplus \mathfrak{C} \in C$; here $\uplus$ denotes the usual disjoint union of structures, which is a special case of Definition 2 for $n = 0$.

Note that the equivalence classes of $\sim$ are in one-to-one correspondence to the equivalence classes of $\sim_0^C$. Also note that C is closed under disjoint unions if and only if $\sim$ has only one equivalence class.

If $\mathfrak{A} \in C$, 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) each $\sim$-equivalence class is closed under homomorphic equivalence,[5]
(2) each $\sim$-equivalence class is closed under disjoint unions.
(3) $\mathfrak{A} \in [\mathfrak{I}]$ if and only if $\mathfrak{A} \uplus \mathfrak{B} \in C$ for all $\mathfrak{B} \in C$.

Lemma 3.11. Let $\mathfrak{A} \in C$ and let $\mathcal{D}$ be the smallest subclass of C that contains $[\mathfrak{A}]$ and whose complement is closed under homomorphisms. Then

---

[5]Two structures are said to be *homomorphically equivalent* if there are homomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ and from $\mathfrak{B}$ to $\mathfrak{A}$.

(1) $\mathcal{D}$ is a union of equivalence classes of $\sim$, and

(2) if $\sim$ has more than one equivalence class, then $C \setminus \mathcal{D}$ is non-empty.

Proof. Let $\mathfrak{C} \in [\mathfrak{A}]$, 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 $\mathfrak{D} \in C$. Then

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

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 $\mathfrak{A} \in [\mathfrak{J}]$, since the complement of $[\mathfrak{J}]$ is closed under homomorphisms. The statement therefore follows from the assumption that $\sim$ has more than one equivalence class. Otherwise, if $\mathfrak{A} \notin [\mathfrak{J}]$, then there exists a structure $\mathfrak{B} \in C$ such that $\mathfrak{A} \uplus \mathfrak{B} \notin C$. Then $\mathfrak{B} \in C \setminus \mathcal{D}$ 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 $\mathfrak{B} \uplus \mathfrak{A}' \in C$. But $\mathfrak{B} \uplus \mathfrak{A}' \in C$ if and only if $\mathfrak{B} \uplus \mathfrak{A} \in C$ since $\mathfrak{A} \sim \mathfrak{A}'$. This is in contradiction to our assumption on $\mathfrak{B}$. □

Example 3.12. 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

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

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

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

$\triangle$

Recall that Lemma 3.5 asserts that the equivalence relation $\equiv_q^{\mathrm{GSO}}$ on the class of finite τ-structures has finitely many equivalence classes $C_1, \ldots, C_m$, and that each of the equivalence classes C_i can be defined by a single GSO τ-sentence Ψ_i with quantifier rank q; we write $T_q^\tau := \{\Psi_1, \ldots, \Psi_m\}$ for this set of GSO sentences.

Definition 4. Let Φ be a GSO τ-sentence of quantifier rank q. Let $J \subseteq \{1, \ldots, m\}$ be such that $\{\Psi_j \in T_q^\tau \mid j \in J\}$ is exactly the set of all sentences in T_q^τ that imply Φ. Then $|J|$ is called the degree of Φ.

It is easy to see that the degree of Φ is exactly the index of $\equiv_q^{\mathrm{GSO}}$ restricted to $[\![\Phi]\!]$.

Lemma 3.13. Let Φ be a GSO τ-sentence of quantifier rank q such that the complement of $C := [\![\Phi]\!]$ is closed under homomorphisms. Let $\sim$ be the equivalence relation from Definition 3 for C. Then for every $\sim$-class $\mathcal{D}$ there exists $I \subseteq \{1, \ldots, m\}$ such that $\mathcal{D} = \bigcup_{i \in I} [\![\Psi_i]\!]$.

Proof. As in the proof of Corollary 3.8 one can use Theorem 3.7 to show for all finite τ-structures $\mathfrak{A}, \mathfrak{B}$ that if $\mathfrak{A} \equiv_q^{\mathrm{GSO}} \mathfrak{B}$, then $\mathfrak{A} \sim \mathfrak{B}$. This means that $\mathcal{D}$ is a union of $\equiv_q^{\mathrm{GSO}}$-classes and therefore there exists $I \subseteq J \subseteq \{1, \ldots, m\}$ such that $\mathcal{D} = \bigcup_{i \in I} [\![\Psi_i]\!]$. □

From this lemma we get immediately the following corollary.

Corollary 3.14. Let Φ and $\sim$ be as in Lemma 3.13. Then the index of $\sim$ is smaller than or equal to the degree of Φ.

THEOREM 3.15. *Let Φ be a GSO sentence. If the complement of $[\![\Phi]\!]$ is closed under homomorphisms, then there are finitely many GSO τ-sentences $\Phi_1, \ldots, \Phi_t$ each of which describes a CSP such that Φ is equivalent to $\Phi_1 \vee \cdots \vee \Phi_t$. If Φ is an MSO sentence, then $\Phi_1, \ldots, \Phi_t$ can be chosen to be MSO sentences as well.*

PROOF. We prove the statement by induction on the degree n of Φ. By Corollary 3.14 the equivalence relation $\sim$ has at most n equivalence classes on τ-structures. Hence, if $n = 1$, then $[\![\Phi]\!]$ is closed under disjoint unions, and we are done by Remark 1.2.

Let $\mathfrak{A}_1, \ldots, \mathfrak{A}_s$ be τ-structures such that $\{[\mathfrak{A}_1], \ldots, [\mathfrak{A}_s]\}$ is the set of all equivalence classes of $\sim$ that are distinct from $[\mathfrak{I}]$ and contained in $[\![\Phi]\!]$. Let $\mathcal{D}_i$ be the smallest subclass of $[\![\Phi]\!]$ that contains $[\mathfrak{A}_i]$ and whose complement is closed under homomorphisms. Note that $[\![\Phi]\!] = \bigcup_{i \leq s} \mathcal{D}_i$ since $[\mathfrak{I}]$ is contained in $\mathcal{D}_i$ for all $i \leq s$. By Lemma 3.11 (1), each $\mathcal{D}_i$ is a union of $\sim$-classes which are themselves a union of $\equiv_q^{\mathrm{GSO}}$-classes by Lemma 3.13. It follows that there exists $I_i \subseteq \{1, \ldots, m\}$ such that $\mathcal{D}_i = \bigcup_{j \in I_i} [\![\Psi_j]\!]$. We define $\Phi_i := \bigvee_{j \in I_i} \Psi_j$. Note that the GSO sentence Φ_i is of quantifier rank q such that $\mathcal{D}_i = [\![\Phi_i]\!]$. Hence, Φ is equivalent to $\bigvee_{i \leq s} \Phi_i$. Lemma 3.11 (2) asserts that $[\![\Phi]\!] \setminus \mathcal{D}_i$ is non-empty, and hence the degree of Φ_i must be strictly smaller than n for all $i \in \{1, \ldots, s\}$. The statement now follows from the inductive assumption. The same argument applies to MSO as well. □

By taking complements, Theorem 3.15 together with Corollary 3.8 implies the following.

Corollary 3.16. *Every GSO sentence whose finite models are closed under homomorphisms is equivalent to a finite conjunction of GSO sentences each of which describes the complement of a CSP of a countable ω-categorical structure. The analogous statement holds for MSO.*

At this point, it seems worthwhile to stress that this property of GSO does not extend to full SO: Not every homomorphism-closed class of structures that can be expressed in Second-order Logic is a finite intersection of complements of CSPs. We present an example of a class of finite τ-structures that can even be expressed in Datalog but cannot be written in this form.

Example 3.17. *Let S and T be unary, and let R be a binary relation symbol. Let 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 $\mathfrak{A}$.*

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

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

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

It is easy to see that each of the structures in $\{\mathfrak{P}_n \mid n \geq 1\}$ is not contained in C, and that the disjoint union of $\mathfrak{P}_i$ and $\mathfrak{P}_j$, for $i \neq j$, is contained in C. It follows that C is not a finite intersection of complements of CSPs (and, by Corollary 3.16, cannot be expressed in GSO). △

4 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* and the characterisation of Datalog expressivity in terms of an existential pebble game [32]; this has been generalised to CSPs for ω-categorical structures [14]. In this section, we build on these results to characterise Datalog expressibility of GSO queries.

4.1 The Canonical Datalog Program of Width (ℓ, k)

A Datalog program Π is called *sound* for a class of τ-structures C if $\llbracket \Pi \rrbracket \subseteq C$. It is called a *canonical Datalog program of width (ℓ, k) for C*, for $\ell, k \in \mathbb{N}$, if if

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

The following definition is due to Feder and Vardi [32] for finite structures and has been generalised to ω-categorical structures by Bodirsky and Dalmau [14].

Definition 5. *Let $\mathfrak{B}$ be an ω-categorical structure with a finite relational signature τ. Let $\mathfrak{B}'$ be the expansion of $\mathfrak{B}$ by all primitively positively definable relations of arity at most ℓ and let τ' be the (finite) signature of $\mathfrak{B}'$. Then the canonical (ℓ, k)-Datalog program for $\mathfrak{B}$ has IDBs $\tau' \setminus \tau$ and EDBs τ. The empty 0-ary relation serves as goal. There is a finite number of inequivalent formulas $\psi(\bar{x}, \bar{y})$ of the form*

$$(\psi_1(\bar{x}, \bar{y}) \wedge \cdots \wedge \psi_j(\bar{x}, \bar{y})) \Rightarrow R(\bar{x})$$

having at most k variables and where $\psi_1, \ldots, \psi_j$ are atomic τ'-formulas and $R \in \tau' \setminus \tau$. For each of the inequivalent formulas $\psi(\bar{x}, \bar{y})$ such that $\mathfrak{B}' \models \forall \bar{x}, \bar{y}.\psi(\bar{x}, \bar{y})$ we introduce a rule

$$R(\bar{x}) :- \psi_1(\bar{x}, \bar{y}), \ldots, \psi_j(\bar{x}, \bar{y}).$$

Theorem 4.1 (Bodirsky and Dalmau [14], Theorem 4). *Let $\mathfrak{B}$ be a countable ω-categorical τ-structure. Let $\ell, k \in \mathbb{N}$ and let Π be the canonical Datalog program of width (ℓ, k) for $\mathfrak{B}$. Then 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 $(\mathfrak{A}, \mathfrak{B})$.*

Moreover, Π is a canonical Datalog program of width (ℓ, k) for the complement of $\text{CSP}(\mathfrak{B})$.

These results extend to finite unions of complements of CSPs for ω-categorical structures (and therefore to all homomorphism-closed GSO-expressible classes of structures by Corollary 3.16) because of the following well-known fact.

Lemma 4.2. *If C_1 and C_2 are in Datalog, then so are $C_1 \cup C_2$ and $C_1 \cap C_2$. 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 :- $\text{goal}_1, \text{goal}_2$ to the union of Π_1 and Π_2. It is clear that these constructions preserve the width. □

Theorem 4.3. *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 3.16 there are GSO sentences $\Phi_1, \ldots, \Phi_m$ and ω-categorical structures $\mathfrak{B}_1, \ldots, \mathfrak{B}_m$ such that Φ is equivalent to $\Phi_1 \wedge \cdots \wedge \Phi_m$ and $\llbracket \neg \Phi_i \rrbracket = \text{CSP}(\mathfrak{B}_i)$. Let Π_i be a canonical Datalog program for $\text{CSP}(\mathfrak{B}_i)$ which exists by Theorem 4.1. Then Lemma 4.2 implies that there exists a Datalog program Π such that $\llbracket \Pi \rrbracket = \llbracket \Pi_1 \rrbracket \cap \cdots \cap \llbracket \Pi_m \rrbracket$. 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 $i \in \{1, \ldots, m\}$, 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, $\mathfrak{A} \in \llbracket \Pi \rrbracket = \llbracket \Pi_1 \rrbracket \cap \cdots \cap \llbracket \Pi_m \rrbracket$. □

If a GSO sentence is in Datalog, then whether or not the problem describes a CSP can be characterised by a syntactic description of the Datalog program (Proposition 4.5). We say that a Datalog rule $\psi_0 :- \psi_1, \ldots, \psi_m$ is *connected* if $\{\psi_1, \ldots, \psi_m\}$ cannot be partitioned into two non-empty subsets with disjoint sets of variables. The following holds for Datalog in general.

Proposition 4.4. *Let Π be a connected Datalog program. Then $[\![\Pi]\!]$ equals the complement of a CSP.*

PROOF. By Remark 1.2, it suffices to show that the complement of $[\![\Pi]\!]$ is closed under disjoint unions. So let $\mathfrak{A}$ and $\mathfrak{B}$ be structures where Π does not derive the goal predicate. Let $\mathfrak{A}'$ and $\mathfrak{B}'$ be the structure computed by Π on $\mathfrak{A}$ and on $\mathfrak{B}$, respectively. If the body of a rule of Π holds on $\mathfrak{A}' \uplus \mathfrak{B}'$, then all variables must take values from $\mathfrak{A}'$, or all variables must take values from $\mathfrak{B}'$, because the rule is connected. Hence, Π does not derive new facts on $\mathfrak{A}' \uplus \mathfrak{B}'$, and in particular does not derive the goal predicate. Hence $\mathfrak{A} \uplus \mathfrak{B}$ belongs to the complement of $[\![\Pi]\!]$ we well. □

In the specific situation that a Datalog program describes a problem in GSO, Proposition 4.4 has a converse.

Proposition 4.5. *Suppose that Π is a Datalog program such that $[\![\Pi]\!]$ is in GSO and equals the complement of a CSP. Then there exists a* connected *Datalog program Π' defining $[\![\Pi]\!]$. In fact, if $\mathfrak{B}$ is the ω-categorical structure such that $[\![\Pi]\!]$ is the complement of $\mathrm{CSP}(\mathfrak{B})$ (which exists by Corollary 3.8), and if Π has width (ℓ, k), then such a program Π' can be obtained from the canonical Datalog program of width (ℓ, k) for $\mathfrak{B}$ by removing all rules that are not connected.*

PROOF. Let Π be the canonical Datalog program of width (ℓ, k) for $\mathfrak{B}$ (Definition 5). Suppose that Π contains a rule which is not connected, e.g., a rule of the form $\psi_0 :- \psi_1, \psi_2$ where ψ_1 and ψ_2 are conjunctions of atomic formulas with disjoint sets of variables. We claim that then $\psi_0 :- \psi_1$ or $\psi_0 :- \psi_2$ must be a rule of Π as well. Otherwise, for each $i \in \{1, 2\}$ there would be a substructure of $\mathfrak{A}_i$ of $\mathfrak{B}$ that satisfies $\psi_i \wedge \neg \psi_0$. Then the disjoint union of $\mathfrak{A}_1$ and $\mathfrak{A}_2$ satisfies $\psi_1 \wedge \psi_2 \wedge \neg \psi_0$, a contradiction to the definition of the canonical Datalog program of $\mathfrak{B}$. □

4.2 The Existential Pebble Game

The existence of canonical Datalog programs for GSO can be be shown via a characterisation of Datalog based on existential pebble games. An informal definition of the existential pebble game was given in the introduction; we now present a formal definition of the concept of *(positional) winning strategies for Duplicator* in this game.

Definition 6 ([14]). *A winning strategy for Duplicator for the existential (ℓ, k)-pebble game on a pair of relational τ-structures $(\mathfrak{A}, \mathfrak{B})$ is a non-empty set $\mathcal{H}$ of partial homomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ such that*

(1) *$\mathcal{H}$ is closed under restrictions;*
(2) *for all $h \in \mathcal{H}$ with $|\mathrm{dom}(h)| = d \le \ell$ 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 $a_1, \ldots, a_{k-d}$.*

We say that *Duplicator wins the existential (ℓ, k)-pebble game for $[\![\Phi]\!]$ on $\mathfrak{A}$* if there exists a countable τ-structure $\mathfrak{B}$ such that $\mathrm{CSP}(\mathfrak{B}) \cap [\![\Phi]\!] = \emptyset$ and Duplicator has a winning strategy on $(\mathfrak{A}, \mathfrak{B})$ as in Definition 6, and otherwise we say that *Spoiler wins the game*. In all our arguments about the existential pebble game, we only need this definition; in particular, we do not need the presentation of the existential pebble game as we have presented it in the introduction.

THEOREM 4.6. *Let Φ be a GSO sentence. Then $[\![\Phi]\!]$ can be defined in Datalog if and only if*

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

Proof. If $[\![\Phi]\!]$ is in Datalog, then it is closed under homomorphisms. Hence, for both directions of the proof we may apply Corollary 3.16 to Φ, which implies that there are GSO sentences $\Phi_1, \ldots, \Phi_m$ and ω-categorical structures $\mathfrak{B}_1, \ldots, \mathfrak{B}_m$ such that Φ is equivalent to $\Phi_1 \wedge \cdots \wedge \Phi_m$ and $[\![\neg\Phi_i]\!] = \mathrm{CSP}(\mathfrak{B}_i)$. First suppose that $[\![\Phi]\!]$ is in Datalog. That is, there exist $\ell, k \in \mathbb{N}$ and a Datalog program Π of width (ℓ, k) such that $[\![\Phi]\!] = [\![\Pi]\!]$. 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 $\mathfrak{B}$ is a countable structure such that $\mathrm{CSP}(\mathfrak{B}) \cap [\![\Phi]\!] = \emptyset$. Then $\mathrm{CSP}(\mathfrak{B}) \cap [\![\Phi_i]\!] = \emptyset$ for some $i \in \{1, \ldots, m\}$; otherwise, if there is a structure $\mathfrak{A}_i \in \mathrm{CSP}(\mathfrak{B}) \cap [\![\Phi_i]\!]$ for every $i \in \{1, \ldots, m\}$, then the disjoint union of $\mathfrak{A}_1, \ldots, \mathfrak{A}_m$ satisfies Φ_i since Φ_i is closed under homomorphisms, and is in $\mathrm{CSP}(\mathfrak{B})$ since $\mathrm{CSP}(\mathfrak{B})$ is closed under disjoint unions; but this is in contradiction to our assumption that $\mathrm{CSP}(\mathfrak{B}) \cap [\![\Phi]\!] = \emptyset$. Hence, $\mathrm{CSP}(\mathfrak{B}) \subseteq \mathrm{CSP}(\mathfrak{B}_i)$ and there is a homomorphism h from $\mathfrak{B}$ to $\mathfrak{B}_i$ (see [14, Lemma 2]). Note that Π is sound for the complement of $\mathrm{CSP}(\mathfrak{B}_i)$, and Π derives goal on $\mathfrak{A}$, and thus Theorem 4.1 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 $(\mathfrak{A}, \mathfrak{B})$.

Now suppose that $\mathfrak{A} \models \neg\Phi$. Hence, there exists $i \in \{1, \ldots, m\}$ 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. By Theorem 4.1, for every $i \in \{1, \ldots, m\}$ there exists a canonical Datalog program Π_i of width (ℓ, k) for $[\![\Phi_i]\!]$. Then Lemma 4.2 implies that there exists a Datalog program Π such that $[\![\Pi]\!] = [\![\Pi_1]\!] \cap \cdots \cap [\![\Pi_m]\!]$. Since each Π_i is sound for $[\![\Phi_i]\!]$, it follows that Π is sound for $[\![\Phi]\!]$. 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 $i \in \{1, \ldots, m\}$, the assumption implies that Spoiler wins the existential (ℓ, k)-pebble game on $(\mathfrak{A}, \mathfrak{B}_i)$. By Theorem 4.1, it follows that Π_i derives goal on $\mathfrak{A}$. Hence, Π derives goal on $\mathfrak{A}$. □

An example application of Theorem 4.6 for showing that a certain MSO sentence cannot be expressed in Datalog can be found in the proof of Theorem 6.5.

5 NESTED GUARDED QUERIES

In this section we revisit logics that express problems that are contained in both GSO and Datalog, and then prove that all of them fail to express all problems that lie in the intersection of GSO and Datalog. *Frontier-guarded Datalog* is a fragment of Datalog that is contained in GSO. *Guarded queries (GQ)* and the more expressive *Nested guarded queries (GQ⁺)* are extensions of frontier-guarded Datalog that have been introduced by Bourhis, Krötzsch, and Rudolph [21]; the definitions will be recalled below. GQ is also strictly more expressive than *Nested Monadically Defined Queries (Nemodeq), Monadically defined queries (Modeq)* [43], and the recently introduced *almost monadic queries* [44]. Both GQ and GQ⁺ are contained in the intersection of Datalog and GSO. We show that GQ⁺ does not contain all queries from the intersection of Datalog and MSO (and hence, neither do GQ, frontier-guarded Datalog, Nemodeq, Modeq, and almost monadic Datalog): there are CSPs whose complement is both in MSO and in Datalog, but not in GQ⁺ (Corollary 5.19).

5.1 Frontier-Guarded Datalog

A rule of a Datalog program is called *frontier-guarded* if all variables of the head appear in a single EDB atom in the rule body.[6] A Datalog program is called *frontier-guarded* if all its rules are frontier-guarded. Note that every

---

[6]The term *frontier-guarded* was originally introduced for existential rules (also referred to as tuple-generating dependencies or Datalog$^\pm$), an expressive extension of Datalog. Frontier-guarded Datalog is then simply the syntactic intersection of frontier-guarded existential rules and Datalog. The name *guarded* is reserved for rules where all variables of the rule (and not just the variables of the head) appear in a single EDB atom in the rule body.

monadic Datalog program is frontier-guarded. Also the Datalog program from the proof of Proposition 2.4 is frontier-guarded.

Proposition 5.1. *Every problem in frontier-guarded Datalog is in GSO.*

Proof. Let Π be a Datalog program. Let Φ_Π be the SO sentence obtained by

- existentially quantifying over the IDBs of Π,
- replacing each rule $\psi :- \phi_1, \ldots, \phi_m$ of Π by the conjunct $\forall \bar{x}(\psi \lor \neg\phi_1 \lor \cdots \lor \neg\phi_m)$ of Φ,
- additionally adding the conjunct $\neg\mathsf{goal}$ to Φ_Π.

Clearly, Π derives goal on a finite structure $\mathfrak{A}$ if and only if $\mathfrak{A}$ satisfies $\neg\Phi_\Pi$.

We will now show that, if Π is frontier-guarded, then Φ_Π can be expressed by a GSO sentence. To this end, we establish that Φ_Π has the same meaning under standard and under guarded semantics, from which the desired result follows via proposition 2.3. Toward a contradiction, assume some structure $\mathfrak{A}$ satisfies Φ_Π under standard but not under guarded semantics (as Φ_Π is preceded only by existentially quantified relation symbols, this is the only possibility). Assume that instantiating the IDBs with the relations $R_1, \ldots, R_k$ over A witnesses that $\mathfrak{A}$ satisfies Φ_Π under standard semantics. Obtain now the guarded relations $R'_1, \ldots, R'_k$ by removing all unguarded tuples from $R_1, \ldots, R_k$, respectively. The desired contradiction is now obtained by arguing that instantiating the IDBs with the relations $R'_1, \ldots, R'_k$ witnesses that $\mathfrak{A}$ satisfies Φ_Π under guarded semantics: First, we note that $\neg\mathsf{goal}$ is still satisfied. Second, for any rule $\psi :- \phi_1, \ldots, \phi_m$ of Π, fixing any variable assignment of $\bar{x}$ with elements of A, we obtain that the truth of $\psi \lor \neg\phi_1 \lor \cdots \lor \neg\phi_m$ under the instantiation with $R_1, \ldots, R_k$ implies its truth under the instantiation with $R'_1, \ldots, R'_k$: On the one hand, should any $\neg\phi_i$ be true, this is immediate (both for EDB and IDB atoms). On the other hand, in case all $\neg\phi_i$ are false, then so is the one $\neg\phi_j$ where $\phi_j = \phi_j(\bar{y})$ is the rule's "frontier guard", i.e., it consists of an EDB atom where $\bar{y}$ contains at least all the variables of $\bar{z}$ from $\psi(\bar{z}) = \psi$. Truth of $\phi_j(\bar{y})$ under the chosen assignment implies that the tuple assigned to $\bar{y}$ is guarded and therefore also the tuple assigned to $\bar{z}$ must be guarded. Yet, then, truth of $\psi(\bar{z})$ under the two IDB instantiations coincides, hence $\psi(\bar{z})$ must hold. □

However, not every problem in GSO which is in Datalog can be expressed by a frontier-guarded Datalog program. To prove this we need the following definition (in contrast to Definition 5, we work with a single parameter, since this is sufficient for our purposes).

Definition 7. *Let $\mathfrak{B}$ be an ω-categorical structure with finite relational signature τ and $k \in \mathbb{N}$. Let s be the maximal arity of τ. Then the* canonical frontier-guarded Datalog program of width k for $\mathfrak{B}$ *is the subset of the canonical Datalog program of width (k, k) for $\mathfrak{B}$ which contains all rules that are frontier-guarded.*

If Π is the canonical frontier-guarded Datalog program of width k for $\mathfrak{B}$ then it follows from Proposition 5.2 below that $[\![\Pi']\!] \subseteq [\![\Pi]\!]$ for every frontier-guarded Datalog program Π' of width (k, k) which is sound for the complement of CSP($\mathfrak{B}$). We need to adapt the existential pebble game on $\mathfrak{A}$ and $\mathfrak{B}$ to the guarded setting as well. In our informal definition of the existential pebble game, we only change the rules for Spoiler: we additionally require that when Spoiler removes pebbles from $\mathfrak{A}$, then all the pebbles that remain on $\mathfrak{A}$ must be *guarded*, i.e., there must exist a tuple in a relation of $\mathfrak{A}$ such that these pebbles must be from the entries of this tuple. The resulting game will be called the *existential guarded k-pebble game*. Formally, we again work exclusively with the concept of a winning strategy $\mathcal{H}$ for Duplicator (similarly as in Definition 6). A partial homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ is called *guarded* if there exists a tuple in a relation of $\mathfrak{A}$ whose entries contain dom(h).

Definition 8. *Let $k \in \mathbb{N}$ and let τ be a finite relational signature. A* winning strategy for Duplicator for the existential guarded k-pebble game on two relational τ-structures $\mathfrak{A}, \mathfrak{B}$ *is a set $\mathcal{H}$ of partial homomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ such that $\mathcal{H}$ is closed under restrictions, and for every $S \subseteq A$ with $|S| \leq k$ and every guarded $h \in \mathcal{H}$ with dom(h) $\subseteq S$, there is an extension $h' \in \mathcal{H}$ of h with domain S.*

The following can be shown analogously to the proof of Theorem 4 in [14]. We omit the proof because all the ideas are present in the proof of a more interesting theorem that we present in full detail in Section 5.5 (Theorem 5.13).

Proposition 5.2. *Let $\mathfrak{B}$ be ω-categorical with finite relational signature τ and let $\mathfrak{A}$ be a finite τ-structure. Then for all $k \in \mathbb{N}$ the following statements are equivalent.*

- *Every sound frontier-guarded Datalog program of width k for the complement of $\mathrm{CSP}(\mathfrak{B})$ does not derive* goal *on $\mathfrak{A}$.*
- *The canonical frontier-guarded Datalog program of width k for $\mathfrak{B}$ does not derive* goal *on $\mathfrak{A}$.*
- *Duplicator has a winning strategy for the existential guarded k pebble game on $\mathfrak{A}, \mathfrak{B}$.*

Recall from Example 1.1 that $\mathrm{CSP}(\mathbb{Q}; <)$ is expressible in MSO; it is easy to see that its complement can be defined by a Datalog program (of width (2,3)).

Proposition 5.3. *The complement of $\mathrm{CSP}(\mathbb{Q}; <)$ cannot be defined by a frontier-guarded Datalog program.*

PROOF. Let $k \in \mathbb{N}$. By Proposition 5.2, it suffices to show that there exists a structure $\mathfrak{A}$ which has no homomorphism to $(\mathbb{Q}; <)$ but Duplicator has a winning strategy for the existential guarded k pebble game on $\mathfrak{A}, (\mathbb{Q}; <)$. Let $\mathfrak{A}$ be a directed cycle of length at least $k + 1$. Let $\mathcal{H}$ be the set of *all* partial homomorphisms h from $\mathfrak{A}$ to $(\mathbb{Q}; <)$ with a domain of size at most k. We claim that $\mathcal{H}$ is a winning strategy for the existential guarded (ℓ, k)-pebble game. Clearly, $\mathcal{H}$ is closed under restrictions. Now, let h be a partial homomorphism whose domain is contained in a tuple from a relation of $\mathfrak{A}$, that is, contained in $\{a_1, a_2\}$ for two subsequent elements of A on the cycle. Let $a_3, \ldots, a_k$ be elements of A. Since $|A| \geq k + 1$ there must be some $b \in A \setminus \{a_1, \ldots, a_k\}$. Let $\prec$ be the order on $a_1, \ldots, a_k$ in which these elements are traversed on the cycle, starting with b. Note that $\prec$ is a linear extension of $<^{\mathfrak{A}}$. We may then extend h to a partial homomorphism h' from $\mathfrak{A}$ to $(\mathbb{Q}; <)$ such that $h(a_i) < h(a_j)$ if $a_i \prec a_j$, for all $i, j \in \{1, \ldots, k\}$. Note that $h' \in \mathcal{H}$; this finishes the proof. □

5.2 Flag and Check

A general mechanism, called *flag and check*, to turn a query language into a potentially more powerful query language has been introduced by Bourhis, Krötzsch, and Rudolph [21], generalising earlier work of Rudolph and Krötzsch [43]. The idea is best described for some fundamental constraint satisfaction problems. It is easy to see that there is no k such that $\mathrm{CSP}(\{0, 1\}; \neq)$ (i.e., graph 2-colorability) is defined by a Datalog program of width $(1, k)$. However, to decide whether a graph is 2-colorable, it suffices to show that there is no vertex v which can reach v via a path of odd length. Note that such a *'check'* can be performed by a Datalog program of width $(1, 2)$ once a *'flag'* has been put on v. Similarly, there is no $k \in \mathbb{N}$ such that $\mathrm{CSP}(\mathbb{Q}; <)$ (i.e., digraph acyclicity) can be defined by a Datalog program of width $(1, k)$ [9]. However, to decide whether a graph contains a directed cycle, it suffices to find a vertex v which can reach v by a directed path with at least one edge. Again, after v has been found and *'flagged'* such a computation can be performed by a Datalog program of width $(1, 2)$.

We now present the formal definition of flag-and-check programs of Bourhis, Krötzsch, and Rudolph [21].

Definition 9. *Let τ be a finite set of relation and constant symbols. A* flag-and-check τ-program (τ-FCP) *of arity m is a set of Datalog rules Π with EDBs from τ and the IDBs $\{$goal$, P_1, \ldots, P_k\}$ where* goal *is a distinguished predicate of arity 0. Moreover, on top of constants from τ, Π may use extra distinguished constant symbols $\lambda_1, \ldots, \lambda_m$. If $\mathfrak{A}$ is a τ-structure, then $\Pi^{\mathfrak{A}}$ is the set of all tuples $(a_1, \ldots, a_m) \in A^m$ such that Π derives* goal *on the $\tau \cup \{\lambda_1, \ldots, \lambda_m\}$-expansion $\mathfrak{A}'$ of $\mathfrak{A}$ where $\lambda_i^{\mathfrak{A}'} = a_i$ for all $i \in \{1, \ldots, m\}$.*

Several formalisms in the literature are based on flag-and-check programs:
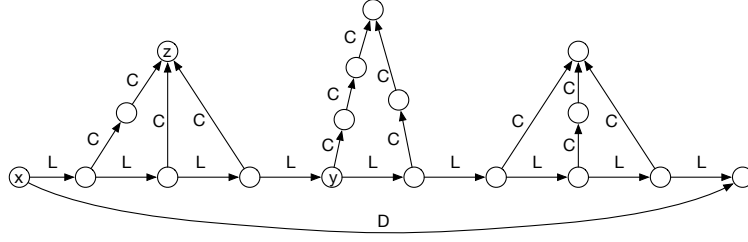
Fig. 3. An illustration of a structure that satisfies the sentence Ψ in Example 5.5.

- A *guarded τ-query (GQ)* is a query with free variables $y_1, \ldots, y_k$, for some $k \in \mathbb{N}$, of the form $\exists x_1, \ldots, x_\ell.\Pi(\bar{x}, \bar{y})$ where Π is a frontier-guarded τ-FCP of arity $m = k + \ell$, with the obvious semantics [21].
- A *monadically defined query (Modeq)* is a query with free variables $y_1, \ldots, y_k$, for some $k \in \mathbb{N}$, of the form $\exists x_1, \ldots, x_\ell.\Pi(\bar{x}, \bar{y})$ where Π is a *monadic τ-FCP* of arity $m = k + \ell$, i.e., all IDBs of Π have arity at most one [43] (we follow the presentation in [21]).

We omit the reference to τ in the notation if the reference is clear from the context. It has been shown that every problem in Modeq can be expressed in MSO [43]. Similarly, one can show that GQ is contained in GSO. This result is implicit in [21], which is why we present the proof here for the convenience of the reader.

Proposition 5.4. *Every problem in GQ is in GSO.*

PROOF. Let $k, \ell \in \mathbb{N}$ and let Π be a guarded flag-and-check program of arity $k + \ell$. We have to find a GSO sentence which is equivalent to $\exists x_1, \ldots, x_\ell.\Pi(\bar{x}, \bar{y})$. Let Φ_Π be the GSO sentence that is equivalent to Π (Proposition 5.1). Then we define Ψ to be the GSO sentence obtained from Φ_Π as follows.

- Replace each occurrence of λ_i, for $i \in \{1, \ldots, m\}$, by a fresh variable x_i.
- Existentially quantify $x_1, \ldots, x_\ell$.   □

An example of a query that can be expressed in GQ, but not in Modeq can be found in [21] (Example 2). An example of a Datalog query in MSO which cannot be expressed in Modeq has already been found in [43] (Example 6); in fact, the same query is not even expressible in GQ, as we will show in Section 5.5 (Example 5.15), so we present it in more detail in the following.

Example 5.5. *Let τ be the EDB signature consisting of the binary relation symbols C, L, and D. Let $\Phi_1(x, y)$ be the MSO τ-formula*

$$
\begin{aligned}
\forall z \exists U_2, U_3 \forall u, v \big(& (C(u, z) \Rightarrow U_3(u)) \\
\wedge\quad & (U_3(v) \wedge C(u, v) \Rightarrow U_3(u)) \\
\wedge\quad & (L(x, u) \Rightarrow U_2(u)) \\
\wedge\quad (U_2(u) \wedge U_3(u) \wedge L(u, v) \Rightarrow U_2(v)) \wedge & \neg U_2(y) \big);
\end{aligned}
$$

Figure 6 depicts a τ-structure that satisfies $\neg\Phi_1(x, y)$ for the elements of the structure labelled with x and y. Let $\Phi_2(u, v)$ be the MSO formula

$$
\exists U_1 \forall x, y \big(U_1(u) \wedge (U_1(x) \wedge \neg\Phi_1(x, y) \Rightarrow U_1(y)) \wedge \neg(U_1(x) \wedge \neg\Phi_1(x, v)) \big).
$$

Finally, let Ψ *be the MSO sentence* $\exists u, v\big(D(u,v) \wedge \neg\Phi_2(u,v)\big)$. *See Figure 3 for an illustration of a* τ-*structure that satisfies* Ψ. *It follows from the results of Rudolph and Krötzsch [43] (Example 5.10) that* Ψ *can be expressed by a Datalog program.* △

5.3 The Expressive Power of Modeqs

The Modeq formalism generalises natural concepts that have been discovered independently in the constraint satisfaction literature; this section can be skipped by impatient readers, and consulted by those interested in the relationship to constraint satisfaction.

Certain flag-and-check programs have been studied in the context of the complexity of constraint satisfaction under the name *peek arc consistency (PAC)* [11], extending the famous (hyper-) arc consistency procedure (AC; see, e.g., [10, Section 8.4]). In the following, τ denotes a finite relational signature.

Definition 10. *Let* $\mathfrak{B}$ *be a countable* ω-*categorical* τ-*structure with the orbits* $O_1, \ldots, O_n$ *and let* $b_1, \ldots, b_n$ *be representatives from these orbits. Then the PAC procedure for* $\mathrm{CSP}(\mathfrak{B})$ *derives* goal *on a* τ-*structure* $\mathfrak{A}$ *if there exists an* $a \in A$ *such that for all* $i \leq n$ *the hyperarc consistency procedure for* $\mathrm{CSP}(\mathfrak{B}, \{b_i\})$ *derives* goal *on* $(\mathfrak{A}, \{a\})$.

For example, the PAC procedure for $\mathrm{CSP}(\{0, 1\}; \neq)$ derives goal on a finite structure $\mathfrak{A}$ if and only if $\mathfrak{A}$ has no homomomorphism to $(\{0, 1\}; \neq)$; similarly, the PAC procedure solves $\mathrm{CSP}(\mathbb{Q}; <)$. We prove that the PAC procedure can be expressed as a Modeq.

Lemma 5.6. *Let* $\mathfrak{B}$ *be a countable* ω-*categorical* τ-*structure with maximal arity* p. *Then there exists a Boolean Modeq* Θ *of width* $(1, p)$ *such that* $[\![\Theta]\!]$ *equals the class of all finite* τ-*structures where the PAC procedure for* $\mathrm{CSP}(\mathfrak{B})$ *derives* goal.

Proof. By the theorem of Ryll-Nardzewski, the automorphism group of $\mathfrak{B}$ has finitely many orbits $O_1, \ldots, O_n$; choose representatives $b_1, \ldots, b_n$ from these orbits. For every $i \leq n$, let Π_i be the Datalog program for the hyperarc consistency procedure for $\mathrm{CSP}(\mathfrak{B}, \{b_i\})$ (which has width $(1, p)$; see Section 8.4 in [10]). Let C_i be the unary IDB of Π_i for the relation $\{b_i\}$, and let Π'_i be the Datalog program obtained from Π_i by adding the rule $C_i(\lambda_1)$:– (without precondition). Since expressibility by Datalog programs of width $(1, p)$ is closed under intersection (Lemma 4.2), there is a Datalog program Π of width $(1, p)$ expressing $\Pi'_1 \wedge \cdots \wedge \Pi'_n$. Also note that $\Theta := \exists x.\Pi$ holds on a τ-structure $\mathfrak{A}$ if and only if there exists an element $a \in A$ such that for every $i \leq n$ the program Π'_i derives goal on $(\mathfrak{A}, \{a\})$, which by definition is the case if and only if the PAC procedure derives goal on $\mathfrak{A}$. This shows the statement. □

The following example demonstrates that the expressivity of Modeq extends well beyond that of PAC.

Example 5.7. *The following is based on an example from [25] of a structure such that the complement of the CSP of this structure cannot be solved by PAC, but can be solved by* singleton arc consistency (SAC).[7] *Below we will show that the mentioned class is even in Modeq. Let* $\mathfrak{B}$ *be the structure with domain* $\{-1, 0, 1, 2\}$ *and relations*

$$R^{\mathfrak{B}} := \{-1, 0, 1, 2\}^2 \setminus \{(-1, -1)\}$$
$$S^{\mathfrak{B}} := \{(0, 1), (1, 2), (2, 0), (-1, -1)\}$$

and let $\mathfrak{A}$ *be the structure with the domain* $\{a, b, c, a', b', c'\}$ *and the relations*

$$R^{\mathfrak{A}} := \{(a, a')\}$$
$$S^{\mathfrak{A}} := \{(a, b), (b, c), (a, c), (a', b'), (b', c'), (a', c')\}$$

---

[7]We do not need the notion of singleton arc consistency in this article and therefore refrain from giving a full definition and refer the interested reader to [25].

Then there is no homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$, but the PAC procedure for $\mathrm{CSP}(\mathfrak{B})$ does not derive goal *on $\mathfrak{A}$ [25]. However, the problem can be defined by the Modeq $\exists x_1, x_2. \Pi$ where Π is the following monadic flag-and-check program of arity two.*

$$U_0(\lambda_1) :- \qquad\qquad\qquad\qquad\qquad U_0(\lambda_2) :-$$
$$U_1(z) :- U_0(y), S(y, z) \qquad\qquad U_2(y) :- U_0(z), S(y, z)$$
$$U_2(z) :- U_1(y), S(y, z) \qquad\qquad U_0(y) :- U_1(z), S(y, z)$$
$$U_0(z) :- U_2(y), S(y, z) \qquad\qquad U_1(y) :- U_2(z), S(y, z)$$
$$\text{goal} :- U_0(y), U_1(y), R(y, z), U_0(z), U_1(z)$$

Claim. $\exists x_1, x_2. \Pi$ *evaluates to true on a finite $\{R, S\}$-structure $\mathfrak{A}$ if and only if there is no homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$. First suppose that there are $a_1, a_2 \in A$ such that Π derives* goal *on the $\{\lambda_1, \lambda_2\}$-expansion $\mathfrak{A}'$ of $\mathfrak{A}$ where $\lambda_i^{\mathfrak{A}_1} = a_i$ for $i \in \{1, 2\}$. This means that there are $b_1, b_2 \in A$ such that $(b_1, b_2) \in R^{\mathfrak{A}}$, and there are paths of net length 0 and 1 from a_1 to b_1, and paths of net length 0 and 1 from a_2 to b_2. In general, the existence of paths of net length 0 and 1 between two vertices implies that both have to be mapped to -1. Therefore, any homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ must map both b_1 and b_2 to -1, which is impossible since $(b_1, b_2) \in R^{\mathfrak{A}}$.*

Now suppose that for any $a_1, a_2 \in A$ the program Π does not derive goal *on the structure $\mathfrak{A}'$ defined above. That is, for every $(b_1, b_2) \in R^{\mathfrak{A}}$ there exists $i \in \{1, 2\}$ such that the for all $u \in A$ that are connected to b_i in the graph defined by $S^{\mathfrak{A}}$ there is no path of net length 0 and no path of net length 1 to b_i. Hence, there must exist a path of net length 2 from u to b_i. We then define $h(u) := b_i + 2 \mod 3$. For all other $u \in A$, we define $h(u) := -1$. It is straightforward to verify that h is a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$.* $\triangle$

5.4 Nested Queries

The expressive power of Datalog fragments can sometimes be increased by considering *nesting*, which is a concept introduced in [43] and further studied in [21]. The idea is that in nested queries we allow the use of other nested queries as if they were atomic formulas. Depending on the query language we start with, nesting may or may not increase the expressive power. Nested Datalog programs, for example, can be rewritten into equivalent Datalog programs without nesting. Similarly, MSO and GSO are closed under nesting. On the other hand, nested monadic queries (*Nemodeq*) are strictly more expressive than monadic queries [43], and, similarly, nested guarded queries (GQ^+) are strictly more expressive than guarded queries, as we will see below (Proposition 5.15). We mention that the *layered tree programs* studied in [24] are in fact the queries obtained from linear monadic Datalog programs with at most one EDB per rule via nesting.

Definition 11 ([21]). *Let $q, m \in \mathbb{N}$ and let τ be a finite relational signature. A q-nested (monadic, guarded) flag-and-check τ-program of arity m is defined inductively as follows. A 1-nested (monadic, guarded) τ-FCP of arity m is the same as a (monadic, guarded) τ-FCP of arity m as defined in Definition 9. A $q + 1$-nested (monadic, guarded) τ-FCP of arity m is a (monadic, guarded) τ-FCP Π of arity m that may use in rule bodies q-nested (monadic, guarded) τ-FCPs in addition to the symbols of τ. These queries are called the* subqueries *of Π at nesting level q; the subqueries of Π at nesting level j are inductively defined as the subqueries of the subqueries at nesting level $j + 1$. In the case of guarded $q + 1$-nested FCPs, the q-nested FCPs may not serve as guards. An FCP has* width m *if all its subqueries have arity at most m.*

A (q-nested, monadic, guarded, width m) τ-query is a query of the form $\exists \bar{x}. \Pi$ where Π is a (q-nested, monadic, guarded, width m) τ-FCP.

Remark 5.8. *Note that all the classes of queries defined in Definition 11 are closed under disjunction by appropriately taking the union of the rule sets, just as in the proof of Lemma 4.2 for Datalog.* $\triangle$

The class of nested guarded queries is denoted by GQ$^+$, and the class of nested monadically defined queries by Nemodeq. It is known that every nested monadically defined query is equivalent to a Datalog program [43]; the same is even true for every nested guarded query [21, Figure 1]. The following fact is implicit in [21].

THEOREM 5.9. *Let τ be a finite relational signature with maximal arity s. Let Π be a q-nested guarded FCP of arity m and EDBs τ such that all rules have at most k variables. Then there exists a Datalog program Π' of width $(m + s, m + k)$ with a distinguished IDB P_Π of arity m such that for every τ-structure $\mathfrak{A}$, a tuple $\bar{a} \in A^m$ satisfies $\Pi(\bar{a})$ if and only if $\bar{a} \in P_\Pi^{\Pi'(\mathfrak{A})}$. The program Π' can be computed in polynomial time from Π.*

PROOF. The proof follows closely the corresponding statement for Nemodeq [43]. For $q = 1$, the Datalog program Π' contains for each rule in Π a new rule obtained by

- replacing each constant λ_i by a new variable x_i,
- replacing each occurrence of goal by $P_\Pi(x_1, \ldots, x_m)$ where P_Π is a new IDB of arity m;
- replacing each atomic formula $R(\bar{y})$ by a new atomic formula $R'(x_1, \ldots, x_m, \bar{y})$ where R' is a new IDB.

For $q > 1$, the translation is defined recursively: let $\Pi_1, \ldots, \Pi_s$ be the subqueries of Π at nesting level $q - 1$. Let $\Pi'_1, \ldots, \Pi'_s$ be the Datalog programs that can be associated to these FCPs by the inductive assumption; we may assume that the IDBs for all these programs are pairwise disjoint. Then Π' is defined to be the union over the rules of $\Pi_1, \ldots, \Pi_s$ together with the rules of Π where we replace each subquery Π_i at nesting level q by P_{Π_i}.

To verify that the Datalog program Π' defined in this way satisfies the required properties, let $\mathfrak{A}$ a τ-structure and suppose that $\bar{a} \in A^m$ satisfies $\Pi(\bar{a})$. This means that Π derives goal on the $\tau \cup \{\lambda_1, \ldots, \lambda_m\}$-expansion $\mathfrak{A}'$ of $\mathfrak{A}$ where $\lambda_i^{\mathfrak{A}'} = a_i$ for all $i \in \{1, \ldots, m\}$, that is, goal$^{\Pi(\mathfrak{A}')} = \{()\}$. It can be shown by induction over the evaluation of Datalog programs (see Remark 2.1) that this is the case if and only if $\bar{a} \in P_\Pi^{\Pi'(\mathfrak{A})}$. □

We therefore use some of the terminology that we introduced for Datalog also for Nemodeq and for GQ$^+$. It is also known that Nemodeq is contained in MSO [43].

Example 5.10. *Example 5.5 can be expressed in Nemodeq [43], and hence is in Datalog and in MSO.* △

Proposition 5.11. *Every problem in GQ$^+$ is contained in GSO.*

PROOF. This is an immediate consequence of Proposition 5.4, because GSO is closed under nesting. □

5.5 The Nested Guarded Game

In this section we present a modification of the existential guarded k-pebble game that allows us to capture the expressive power of GQ and GQ$^+$. In particular, we will prove that the query in Example 5.5 is not in GQ. We call this game the *nested guarded game*; it will also be used in the next section to prove that there are problems in the intersection of Datalog and GSO that cannot be expressed in GQ$^+$.

Again we start with an informal description of the game. Let τ be a finite relational signature. The game has three parameters, q, m, and k. There are two players, Spoiler and Duplicator, that play on a pair of τ-structures $(\mathfrak{A}, \mathfrak{B})$. Each player has k labelled pebbles, out of which m pebbles are blue, the others are red. Spoiler places the blue pebbles $\mathfrak{A}$, Duplicator answers by placing her blue pebbles on elements of $\mathfrak{B}$. Then the two players play the existential guarded $(k - m)$-pebble game with the red pebbles, with the difference that Duplicator loses if the map between all the pebbled vertices (blue and red alike) is not a partial homomorphism. At most q times, Spoiler can relocate the red pebbles (without the guard restriction). We present a formal definition of winning strategies for Duplicator in the nested guarded game; all statements about the game only involve such winning strategies for Duplicator.

Definition 12. *Let $k, q, m \in \mathbb{N}$. A winning strategy for Duplicator for the q-nested width m guarded k-pebble game on two τ-structures $(\mathfrak{A}, \mathfrak{B})$ is a sequence $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_q$ of non-empty sets of partial homomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ such that*

(1) *each of $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_q$ is closed under restriction;*

(2) *if $h \in \mathcal{H}_i$, for $i \in \{1, \ldots, q\}$, is such that there exists a tuple in a relation of $\mathfrak{A}$ whose entries contain $\mathrm{dom}(h)$, then $\mathcal{H}_i$ contains for every $S \subseteq A$ with $|S| \leq k$ and $\mathrm{dom}(h) \subseteq S$ an extension of h with domain S;*

(3) *if $h \in \mathcal{H}_i$ for $i \in \{1, \ldots, q\}$ and $S \subseteq A$ with $|S| \leq k$ is such that $\mathrm{dom}(h) \subseteq S$ and $|\mathrm{dom}(h)| \leq m$, then $\mathcal{H}_{i-1}$ contains an extension of h with domain S.*

To prove the connection between GQ^+ and the q-nested guarded k-pebble game we need an appropriate notion of canonical Boolean q-nested guarded queries.

Remark 5.12. *We present the game characterisation of GQ^+ only for such GQ^+ sentences whose sets of finite models are complements of CSPs of ω-categorical structures $\mathfrak{B}$; however, in analogy to Section 4, one can define a game that characterises the general case, using Corollary 3.16. Since this is notationally and technically heavier, but does not add new ideas or new applications in the following, we only present the special case.* △

Definition 13 (Canonical nested guarded queries). *Let $\mathfrak{B}$ be an ω-categorical structure with a finite relational signature τ and let $R \subseteq B^m$ be a relation with a primitive positive definition over $\mathfrak{B}$. The canonical q-nested guarded query Π_R of width (m, k) for $(\mathfrak{B}, R)$ is defined by induction over q as follows. For $q = 1$, let $o^1 = (o_1^1, \ldots, o_m^1), \ldots, o^s = (o_1^s, \ldots, o_m^s) \in B^m$ be representatives of all orbits of m-tuples of $\mathrm{Aut}(\mathfrak{B})$ that are not contained in R. For $i \in \{1, \ldots, s\}$ let Π_{o^i} be the canonical frontier-guarded Datalog program of width k of the ω-categorical structure $(\mathfrak{B}; o_1^i, \ldots, o_m^i)$.[8] We transform Π_{o^i} into an m-ary FCP by replacing literals of the form $o_j^i = x$ by $\lambda_j = x$. Let Π_R be obtained by taking the conjunction of all the resulting FCPs over all $i \in \{1, \ldots, s\}$ (which is again an FCP; this can be shown analogously to Lemma 4.2).*

For $q > 1$, we suppose inductively that for all $m \geq 0$ and relations S with a primitive positive definition over $\mathfrak{B}$ the canonical $q - 1$ nested guarded query Ξ_S of width (m, k) for $(\mathfrak{B}, S)$ is already defined. Replace in the canonical 1 nested guarded query of width (m, k) for $(\mathfrak{B}, R)$ every occurrence of an EDB S by Ξ_S. We define Π_R to be the resulting q-nested query.

The canonical q nested guarded query Π of width (m, k) for $\mathfrak{B}$ is obtained from the canonical q nested guarded query Π_R of width (m, k) for $(\mathfrak{B}, R)$, where R is the empty relation of arity 0, by existentially quantifying all free variables.

Definition 14 (q-nested m-bounded k-variable logic). *Let τ be a relational signature and let $q, m, k \in \mathbb{N}$. Then τ-sentences in q-nested m-bounded k-variable logic $L_q^{m,k}$ are built inductively from atomic τ-formulas and the following operation: if $\phi_1, \ldots, \phi_s$ are formulas from $L_q^{m,k}$, then*

$$\exists \bar{y} \, (\phi_1 \wedge \cdots \wedge \phi_s)$$

is in $L_q^{m,k}$ if

- *there exists an $i \in \{1, \ldots, s\}$ and an atomic formula ψ in ϕ_i whose variables contain all the variables that are not existentially quantified in $\bar{y}$,*
- *each of the formulas $\phi_1, \ldots, \phi_s$ is from $L_{q-1}^{m,k}$.*

The following result and its proof are inspired by Theorem 4.1 and its proof.

---

[8]Observe that if $(s_1, \ldots, s_m)$ and $(s_1', \ldots, s_m')$ have the same orbit in $\mathrm{Aut}(\mathfrak{B})$, then $(\mathfrak{B}, s_1, \ldots, s_m)$ and $(\mathfrak{B}, s_1', \ldots, s_m')$ have the same canonical frontier-guarded Datalog program.

THEOREM 5.13. *Let $\mathfrak{B}$ be an ω-categorical structure with finite relational signature τ and let $\mathfrak{A}$ be a finite τ-structure. Then for all $q, m, k \in \mathbb{N}$ the following statements are equivalent.*

(1) *Every q-nested guarded query of width (m, k) which is sound for the complement of $\mathrm{CSP}(\mathfrak{B})$ is false on $\mathfrak{A}$.*
(2) *The canonical q-nested guarded query of width (m, k) for the complement of $\mathrm{CSP}(\mathfrak{B})$ is false on $\mathfrak{A}$.*
(3) *Duplicator has a winning strategy for the q-nested width m guarded k-pebble game on $(\mathfrak{A}, \mathfrak{B})$.*
(4) *Every sentence from $L_q^{m,k}$ which holds in $\mathfrak{A}$ also holds in $\mathfrak{B}$.*

PROOF. For the implication from 1. to 2., it suffices to prove that the canonical q-nested guarded query Π of width (m, k) is sound for the complement of $\mathrm{CSP}(\mathfrak{B})$. Let $\mathfrak{A}$ be a finite τ-structure. Let $R \subseteq B^m$ be primitively positively definable over $\mathfrak{B}$. If $q = 1$ then $\Pi(t_1, \ldots, t_m)$ holds in $\mathfrak{A}$ if and only if for every tuple $(s_1, \ldots, s_m) \in B^m \setminus R$ the canonical frontier-guarded Datalog program for $(\mathfrak{B}, s_1, \ldots, s_m)$ derives goal. This in turn means that there exists no homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ which maps $(t_1, \ldots, t_m)$ to $(s_1, \ldots, s_m)$, because the canonical frontier-guarded Datalog program, which is a subset of the canonical Datalog program of width k, is sound by definition. By induction over the nesting depth q it follows that if $\mathfrak{A}$ satisfies the canonical q nested guarded query of with (m, k) for $\mathfrak{B}$, then there is no homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$, and hence Π is sound for $\mathrm{CSP}(\mathfrak{B})$.

For the implication from 2. to 3., let $\mathfrak{B}'$ be the expansion of $\mathfrak{B}$ by all primitive positive definable relations of arity at most m; since $\mathfrak{B}$ is ω-categorical, the structure $\mathfrak{B}'$ still has a finite relational signature τ'. We compute a winning strategy $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_q$ for Duplicator for the q nested width m guarded k pebble game on $(\mathfrak{A}, \mathfrak{B})$ using the canonical q-nested guarded query of width (m, k) for $(\mathfrak{B}', R)$ for every relation R of arity m which has a primitive positive definition in $\mathfrak{B}$. Define $\mathfrak{A}'_0$ be the expansion of $\mathfrak{A}$ with the same signature as $\mathfrak{B}'$ that contains for every m-ary relation symbol $R \in \tau' \setminus \tau$ the empty relation, i.e., $R^{\mathfrak{A}'_0} = \emptyset$. For $i \in \{0, \ldots, q-1\}$, suppose that $\mathfrak{A}'_0, \ldots, \mathfrak{A}'_i$ have already been defined. Let $\mathfrak{A}'_{i+1}$ be the expansion of $\mathfrak{A}$ with the same signature as $\mathfrak{B}'$ that contains for every m-ary relation symbol $R \in \tau' \setminus \tau$ the relation accepted by the canonical i-nested guarded query of width (m, k) for $(\mathfrak{B}', R)$ on $\mathfrak{A}'_{i-1}$. For $i \in \{0, \ldots, q\}$, let $\mathcal{H}_i$ be the set of all partial homomorphisms f from $\mathfrak{A}'_i$ to $\mathfrak{B}'$ with domain of size at most k. Note that since the canonical q nested guarded query of width (m, k) for $\mathfrak{B}$ is false on $\mathfrak{A}$, the relation symbol F of arity 0 which denotes the empty relation over $\mathfrak{B}$ (which is primitively positively definable) denotes the empty relation over $\mathfrak{A}'_q$ as well, and hence $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_q$ are non-empty. Clearly, these sets are closed under restriction.

We claim that the sequence $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_q$ also satisfies items 2. and 3. in the definition of winning strategies (Definition 12). Indeed, for item 2., let $h \in \mathcal{H}_i$, for $i \in \{1, \ldots, q\}$, be such that there exists a tuple in a relation of $\mathfrak{A}$ whose entries contain $\mathrm{dom}(h) = \{a_1, \ldots, a_j\}$, and let $S \subseteq A$ be a superset of $\mathrm{dom}(h)$ of size at most k. Consider the following frontier-guarded rule in the canonical Datalog program for $(\mathfrak{B}, h(a_1), \ldots, h(a_j))$: the body of the rule is the canonical query ϕ of the substructure of $\mathfrak{A}'_q$ induced on S. The head of the rule is $R(a_1, \ldots, a_j)$ where $R \in \tau'$ is such that $R^{\mathfrak{B}'}$ is the projection of the relation defined by ϕ in $\mathfrak{B}'$ to $a_1, \ldots, a_j$. By the assumption that there exists a tuple in a relation of $\mathfrak{A}$ whose entries contain $\{a_1, \ldots, a_j\}$, this rule is indeed frontier guarded. Since the rule applies to $\mathfrak{A}'$ we obtain that $(a_1, \ldots, a_j) \in R^{\mathfrak{A}'_q}$. By the definition of $\mathcal{H}_i$, we have that $(h(a_1), \ldots, h(a_j)) \in R^{\mathfrak{B}'}$. By the definition of $R^{\mathfrak{B}'}$ we can find elements in B' for the other variables of ϕ so that ϕ is satisfied, and this shows that h can be extended to a partial homomorphism from $\mathfrak{A}'_q$ to $\mathfrak{B}'$ defined on all of S.

We show item 3. by induction over $i \in \{1, \ldots, q\}$. Let $h \in \mathcal{H}_i$ be such that $\mathrm{dom}(h)$ has size at most m and let $S \subseteq A$ be a superset of $\mathrm{dom}(h) = \{a_1, \ldots, a_j\}$ of size $j \le k$. We have to show that $\mathcal{H}_{i-1}$ contains an extension of h with domain S. Let ϕ be the primitive positive formula obtained from the canonical query of the substructure $\mathfrak{A}_S$ of $\mathfrak{A}'_i$ induced on S by existentially quantifying all variables except the ones in the domain of h. Then ϕ is equivalent to a primitive positive formula over the signature of $\mathfrak{B}$; let R be the relation symbol of $\mathfrak{B}'$ for the relation defined by ϕ in $\mathfrak{B}$. For any $(t_1, \ldots, t_j) \in B^j \setminus R^{\mathfrak{B}'}$ the canonical q-nested query derives goal since the canonical query of $\mathfrak{A}_S$ together with $a_1 = t_1, \ldots, a_j = t_j$ is unsatisfiable in $(\mathfrak{B}, t_1, \ldots, t_j)$. Hence, $(a_1, \ldots, a_j) \in R^{\mathfrak{A}'}$. In this

case, $(h(a_1), \ldots, h(a_j)) \in R^{\mathfrak{B}'}$ since h is a partial homomorphism on $\{a_1, \ldots, a_j\}$. Therefore, $(h(a_1), \ldots, h(a_j))$ satisfies ϕ and the witnesses for the existentially quantified variables of ϕ provide an extension of h to a partial homomorphism from $\mathfrak{A}'_{i+1}$ to $\mathfrak{B}$ which is defined on all of S.

3. implies 4.: We show by induction over the syntactic structure of $L_q^{m,k}$ formulas that if $\phi(v_1, \ldots, v_m)$ is an $L_q^{m,k}$ formula, then for all $h \in \mathcal{H}_q$ and all elements $a_1, \ldots, a_m$ from the domain of h, if $\mathfrak{A}$ satisfies $\phi(a_1, \ldots, a_m)$, then $\mathfrak{B}$ satisfies $\phi(h(a_1), \ldots, h(a_m))$. For $m = 0$ this implies 4. The base case of the induction is obvious because atomic formulas are preserved by homomorphisms. Now suppose that ϕ is of the form $\exists \bar{y}(\phi_1 \wedge \cdots \wedge \phi_s)$ where $\phi_1, \ldots, \phi_s$ are formulas from $L_q^{m,k}$. Let $\bar{c}$ be a tuple providing witnesses for the variables $\bar{y}$ that shows that $\phi(a_1, \ldots, a_m)$ holds in $\mathfrak{A}$.

If there exists an $i \in \{1, \ldots, s\}$ and an atomic formula $\psi(\bar{z})$ in ϕ_i whose variables contain all the variables that are not existentially quantified in $\bar{y}$, then $h(\bar{z})$ is a tuple in a relation of $\mathfrak{A}$ that contains the domain of h. Hence, h has an extension h' whose domain contains the entries of $\bar{c}$, because $h \in \mathcal{H}_q$. Then $h'(\bar{c})$ provides the witnesses for $\bar{y}$ that show that $\phi(h(a_1), \ldots, h(a_m))$ holds in $\mathfrak{B}$, because $\phi_i(h(a_1), \ldots, h(a_m), h'(\bar{c}))$ holds by inductive assumption for every $i \leq s$. If each of the formulas $\phi_1, \ldots, \phi_s$ is even from $L_{q-1}^{m,k}$, then we consider the extension h' of h from $\mathcal{H}_{q-1}$ whose domain also contains the entries of $\bar{c}$; such an extension exists because $h \in \mathcal{H}_q$. By the inductive assumption, $\mathfrak{A}$ satisfies $\phi_i(h'(a_1), \ldots, h'(a_m), h'(\bar{c}))$ for every $i \leq s$. This concludes the induction.

Finally, 4. implies 1.: suppose that there is a q nested guarded query ϕ of width (m, k) which is sound for the complement of $\mathrm{CSP}(\mathfrak{B})$ and true on $\mathfrak{A}$. We translate the evaluation of ϕ on $\mathfrak{A}$ into a sentence from $L_q^{m,k}$ which is true on $\mathfrak{A}$ but false on $\mathfrak{B}$. Suppose that ϕ is of the form $\exists \bar{x}.\Pi$ where Π is a q-nested guarded FCP, and let $\bar{a}$ be elements of A for the variables in $\bar{x}$ that show that ϕ is true in $\mathfrak{A}$.

For each IDB R of Π of arity k and every k-tuple $\bar{a}$ such that Π derives $R(\bar{a})$ on $\mathfrak{A}$ we define an $L_q^{m,k}$-formula $\psi(x_1, \ldots, x_k)$ which holds on $\bar{a}$ in $\mathfrak{A}$ and such that Π derives $R(\bar{x})$ on the canonical database of $\psi(\bar{x})$ (after transforming it into prenex normal form) by induction over the evaluation of Π on $\mathfrak{A}$. Let $\theta(x_1, \ldots, x_k, \bar{y})$ be the body of the rule of Π that derived $R(\bar{a})$ on $(\bar{a}, \bar{b})$. For every atomic formula $S(\bar{z})$ of θ we may assume by inductive assumption that there exists an $L_q^{m,k}$-formula $\chi(\bar{z})$ which holds on the respective entries $\bar{c}$ of $(\bar{a}, \bar{b})$ in $\mathfrak{A}$ and such that Π derives $S(\bar{c})$ on the canonical database of $\chi(\bar{z})$. Then we define ψ to be the conjunction of all these formulas where all variables $\bar{y}$ are existentially quantified. This formula clearly holds on $\bar{a}$ in $\mathfrak{A}$, and Π derives $R(\bar{x})$ on the canonical database of ψ. The rule of Π that derives $R(\bar{a})$ must be guarded, and hence contains an atomic formula whose variables contain $x_1, \ldots, x_k$, so the formula ψ is indeed an $L_q^{m,k}$-formula. Then the formula that we obtain for the IDB R of Π that denotes B^k in $\mathfrak{B}'$ by existentially quantifying all variables is true on $\mathfrak{A}$ but false on $\mathfrak{B}$. $\qquad \square$

Corollary 5.14. *Let $\mathfrak{B}$ be an ω-categorical structure with a finite relational signature τ and let $q, m \geq 1$. Then the complement of $\mathrm{CSP}(\mathfrak{B})$ cannot be expressed by a q-nested width m guarded query if and only if for all $k \geq 1$ there exists a finite τ-structure $\mathfrak{A} \notin \mathrm{CSP}(\mathfrak{B})$ such that Duplicator has a winning strategy for the q-nested width m guarded k-pebble game on $(\mathfrak{A}, \mathfrak{B})$.*

Proof. For the backwards direction, let Π be a q-nested guarded query of width (m, k) which is sound for the complement of $\mathrm{CSP}(\mathfrak{B})$. By assumption, there exists $\mathfrak{A} \notin \mathrm{CSP}(\mathfrak{B})$ such that Duplicator has a winning strategy for the q nested width m guarded k pebble game on $(\mathfrak{A}, \mathfrak{B})$. By the implication from 3. to 1. in Theorem 5.13, the query Π is false on $\mathfrak{A}$. But then Π does not express the complement of $\mathrm{CSP}(\mathfrak{B})$.

For the forwards direction, we show the contraposition. We assume that there exists a $k \geq 1$ such that for every $\mathfrak{A} \notin \mathrm{CSP}(\mathfrak{B})$ Duplicator has no winning strategy for the q nested width m guarded k pebble game on $(\mathfrak{A}, \mathfrak{B})$. It suffices to prove that the canonical q nested guarded query Π of width (m, k) for the complement of $\mathrm{CSP}(\mathfrak{B})$ expresses the complement of $\mathrm{CSP}(\mathfrak{B})$. Let $\mathfrak{A}$ be an instance of $\mathrm{CSP}(\mathfrak{B})$. If $\mathfrak{A}$ is satisfiable then Π is false

on $\mathfrak{A}$, because Π is sound for the complement of CSP($\mathfrak{B}$) as shown in the proof of the implication 1. implies 2. in Theorem 5.13. If $\mathfrak{A}$ is unsatisfiable then the contraposition of the implication from 2. to 3. in Theorem 5.13 implies that Π is true on $\mathfrak{A}$. Hence, Π expresses the complement of CSP($\mathfrak{B}$). □

To illustrate the power of our game characterisation of GQ, we show that there are problems in the intersection of Datalog and MSO that are inexpressible in GQ.

Proposition 5.15. *The class of finite structures described by the MSO sentence Ψ from Example 5.5 is (in Datalog, but) not expressible in GQ.*

PROOF. It is easy to see that Ψ describes a class C of finite structures whose complement is closed under disjoint unions, and hence is the complement of a CSP by Remark 1.2. By Corollary 3.16 there exists an ω-categorical structure $\mathfrak{B}$ such that the complement of CSP($\mathfrak{B}$) equals C.

To prove the inexpressibility in GQ, we use Corollary 5.14 for the special case $q = 1$. Let $m, k \geq 1$. Let $\phi_1(x_0, x_k)$ be the formula

$$\exists x_1, \ldots, x_{k-1} \bigwedge_{i \in \{0, \ldots, k-1\}} C(x_i, x_{i+1}).$$

Let $\phi_2(y_0, y_k)$ be formula $\exists z, y_1, \ldots, y_{k-1} \bigwedge_{i \in \{1, \ldots, k-1\}} \big(\phi_1(y_i, z) \wedge L(y_i, y_{i+1})\big)$. Let ϕ be the formula $\bigwedge_{i \in \{0, \ldots, m\}} \phi_2(z_i, z_{i+1}) \wedge D(z_0, z_{m+1})$, rewritten in prenex normal form, and let $\mathfrak{A}$ be the canonical database of ϕ.

Claim 1. $\mathfrak{A}$ satisfies Ψ, and hence is an unsatisfiable instance of CSP($\mathfrak{B}$). First note that $(z_0, z_1), (z_1, z_2), \ldots, (z_m, z_{m+1})$ are precisely the pairs in A^2 that do not satisfy the formula Φ_1 from definition of Ψ in Example 5.5. Then the vertices z_0, z_{m+1} play the role of u and v: to see that $\Phi_2(u, v)$ does not hold in $\mathfrak{A}$, let $U_1 \subseteq A$ be such that it satisfies the first two conjuncts of Φ_2 for all $x, y \in A$. Then U_1 must contain z_0 by the first conjunct, and z_1 by the second conjunct, and inductively by the second conjunct it must contain all of $z_0, z_1, \ldots, z_{m+1}$, and hence v. But then the last disjunct $\neg(U_1(x) \wedge \neg\Phi_1(x, v))$ is not satisfied for $x = z_m$.

Claim 2. Duplicator has a winning strategy $\mathcal{H}_0, \mathcal{H}_1$ for the 1-nested width m guarded k pebble game on $\mathfrak{A}$ and $\mathfrak{B}$: we set $\mathcal{H}_0$ to be the set of all partial homomorphisms h from $\mathfrak{A}$ to $\mathfrak{B}$ such that $\text{dom}(h) \leq m$. Set $\mathcal{H}_1$ to be the set of all partial homomorphisms h from $\mathfrak{A}$ to $\mathfrak{B}$ such that if $\text{dom}(h)$ contains the variable z from a conjunct $\phi_2(y_0, y_k)$ of ϕ, and $\phi_2'(\bar{u})$ is the formula obtained from the rewriting of ϕ_2 into prenex normal form and existentially quantifying all variables except for the ones that are in the domain of h, then $h(\bar{u})$ satisfies ϕ_2' in $\mathfrak{B}$. Note that $\text{dom}(h)$ cannot contain the variable z for all of the conjuncts of ϕ of the form $\phi_2(y_0, y_k)$, and hence $\mathcal{H}_1$ is non-empty. It is straightforward to verify that $\mathcal{H}_0, \mathcal{H}_1$ satisfies the three items from Definition 12.

The statement is an immediate consequence of Corollary 5.14 for $q = 1$ and the two claims. □

5.6 Separation in GSO

In this section we present an example of a GSO sentence which is in Datalog, but cannot be expressed in GQ$^+$. The GSO sentence describes the complement of the CSP of the following structure $\mathfrak{B}$, which is fixed throughout this section. Define $\mathfrak{B} := (\mathbb{N}; D, R)$ where

$$D := \{(x, y) \in \mathbb{N}^2 \mid x \neq y\}$$
$$R := \{(x, y, z) \in \mathbb{N}^3 \mid x = y \Rightarrow y = z\}.$$

Clearly, the structure $\mathfrak{B}$ is ω-categorical. The structure $\mathfrak{B}$ plays a prominent role in the classification of structures preserved by all permutations [13] and the complexity of quantified constraint satisfaction problems for equality constraints [12, 45]. The proof of the following proposition uses ideas from Example 2.2.

Proposition 5.16. *There is a GSO sentence that expresses* CSP($\mathfrak{B}$).

PROOF. Let $\tau = \{D, R\}$ be the signature of $\mathfrak{B}$. Let S be a new ternary and X a new unary relation symbol. Let χ_X be the formula which specifies that

- X is non-empty,
- if x, y, z are such that $S(x, y, z)$, then $y \in X$ if and only if $z \in X$, and
- X is minimal with respect to the properties above (the details are similar as in Example 2.2).

Let ϕ be the $\tau \cup \{S, X\}$-sentence which expresses that there exists $S \subseteq R$ such that for all X that satisfy χ_X we have

(1) for all x, y, if $D(x, y)$ then $x \notin X$ or $y \notin X$, and
(2) for all x, y, z, if $R(x, y, z)$ but not $S(x, y, z)$, then $x \notin X$ or $y \notin X$.

We claim that a finite structure $\mathfrak{A}$ satisfies ϕ if and only if $\mathfrak{A}$ has a homomorphism to $\mathfrak{B}$. If h is a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$, then we define $S \subseteq R$ to be the relation that contains $(x, y, z) \in R$ if and only if $h(x) = h(y)$. Let $X \subseteq A$ be such that χ_X is satisfied. We have to show that 1. and 2. above are satisfied. Let $\mathfrak{A}'$ be the $\{E, D\}$-structure with domain A and the relations

$$E^{\mathfrak{A}'} \coloneqq \bigcup_{(x,y,z) \in S} \{(x, y), (y, z)\}$$

$$D^{\mathfrak{A}'} \coloneqq D^{\mathfrak{A}} \cup \{(x, y) \mid \exists z (R(x, y, z) \wedge \neg S(x, y, z))\}$$

Note that h is a homomorphism from $\mathfrak{A}'$ to structure $(\mathbb{N}; E, D)$ from Example 2.2. Recall from Example 2.2 that any homomorphism from $\mathfrak{A}'$ to $(\mathbb{N}; E, D)$ must be constant on X. Hence, if $(x, y) \in D$ then $h(x) \neq h(y)$ and hence we must have $x \notin X$ or $y \notin X$. To see that 2. is satisfied as well, note that if $(x, y, z) \in R \setminus S$, then $(x, y) \in D^{\mathfrak{A}'}$, and hence $h(x) \neq h(y)$ by the definition of S, which again implies that $x \notin X$ or $y \notin X$.

Now suppose that $\mathfrak{A}$ satisfies ϕ. Let $S \subseteq R$ be the ternary relation witnessing this. Define the structure $\mathfrak{A}'$ as above. As in Example 2.2 we may argue that there exists a homomorphism from $\mathfrak{A}'$ to $(\mathbb{N}; E, D)$, which is a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ as well. □

Proposition 5.17. *The complement of* $\mathrm{CSP}(\mathfrak{B})$ *cannot be expressed in* GQ^+.

PROOF. Let $q, m \geq 1$. We use Corollary 5.14 to show that $\mathrm{CSP}(\mathfrak{B})$ cannot be expressed by a q-nested width m guarded query. Let $k \geq 1$. We construct an unsatisfiable instance $\mathfrak{A}$ of $\mathrm{CSP}(\mathfrak{B})$ such that Duplicator wins the q nested width m guarded k-pebble game on $(\mathfrak{A}, \mathfrak{B})$. Without loss of generality we may assume that $m \leq k$. It will be convenient to specify $\mathfrak{A}$ as the canonical database of a primitive positive $\{D, R\}$-formula ϕ. Let $\phi_0(x, y)$ be the formula $R(x, x, y)$, and for $i > 0$ let $\phi_i(x, y)$ be the formula

$$\exists x_0, \ldots, x_k \left(x = x_0 \wedge \bigwedge_{j=0}^{k-1} \phi_{i-1}(x_j, x_{j+1}) \wedge R(x_0, x_k, y) \right).$$

Let ϕ be the formula $\phi_s(x, y) \wedge D(x, y)$ where $s \coloneqq q + 1$. Let $\mathfrak{A}$ be the canonical database of ϕ after transforming ϕ into prenex normal form. Note that every variable of $\phi_i(x, y)$ except x and x_0 appears exactly once as the final argument in a conjunct that involves R.

Claim. $\mathfrak{A}$ is an unsatisfiable instance of $\mathrm{CSP}(\mathfrak{B})$. This follows from the fact that $\phi_i(x, y)$ implies $x = y$, for every $i \leq s$, which can be shown by a straightforward induction over i.

Claim. Duplicator has a winning strategy in the q nested width m guarded k-pebble game on $(\mathfrak{A}, \mathfrak{B})$.

Let $\mathcal{H}_0$ be the set of all partial homomorphisms h from substructures of $\mathfrak{A}$ to $\mathfrak{B}$ with domain size at most k, and let $\mathcal{H}_i$, for all $i \in \{1, \ldots, q\}$, be the set of all partial homomorphisms h from substructures of $\mathfrak{A}$ to $\mathfrak{B}$ with domain size at most k such that $h(u) = h(v)$, for $u, v \in \mathrm{dom}(h)$, if and only if both u and v

- are part of the same subformula of ϕ of the form ϕ_{i-1}, or

- are elements of different subformulas of ϕ of the form ϕ_{i-1} but dom(h) contains all the variables $x_r, \ldots, x_s, x_{s+1}$ from ϕ_i for some $0 \leq r < s < k$ and u appears in the subformula $\phi_{i-1}(x_r, x_{r+1})$ and v appears in the subformula $\phi_{i-1}(x_s, x_{s+1})$.

Since dom$(h) \leq k$ it is impossible that $r = 0$ and $s = k - 1$; hence, if x_0 and x_k are from dom(h), then we will have $h(x_0) \neq h(x_k)$.

Clearly, $\mathcal{H}_i$ is closed under restrictions and non-empty. Now suppose that $h \in \mathcal{H}_i$ is such that there exists a tuple a in a relation of $\mathfrak{A}$ whose entries contain dom(h). Let $S \subseteq A$ be a superset of dom(h) of cardinality at most k. If this relation is D, then a equals (x, y), we may clearly extend h to some $h' \in \mathcal{H}_i$ with $h'(x) \neq h'(y)$. Otherwise, the relation must be R and a equals (x_0, x_k, y) for some variables x_0, x_k, y from a subformula of ϕ of the form $\phi_j(x, y)$ for some $j \leq i$. Note that $h(x_0) = h(x_k) = h(y)$ if $j < i$, and $h(x_0) \neq h(x_k)$ and $h(x_k) \neq h(y)$ otherwise. It is easy to see that in both cases we may extend h to some $h' \in \mathcal{H}_i$ with dom$(h') = S$. Finally, if $h \in \mathcal{H}_i$ for $i \in \{1, \ldots, q\}$ and $S \subseteq A$ with $|S| \leq k$ is such that dom$(h) \subseteq S$, then $\mathcal{H}_{i-1}$ contains an extension h' of h with domain S: if $x \in S$ appears in a copy of ϕ_{i-1} whose variables intersect dom(h), we set $h'(x)$ to the value taken by these variables under h. If dom(h) contains all the variables $x_r, \ldots, x_s, x_{s+1}$ from ϕ_i for some $0 \leq r < s < k$ and x appears in the subformula $\phi_{i-1}(x_s, x_{s+1})$ and some variable of $\phi_{i-1}(x_r, x_{r+1})$ appears in dom(h), then we set $h'(x)$ to the value taken by this variable under h. Otherwise, pick $h'(x)$ to be a new value. Hence, $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_q$ is indeed a winning strategy for Duplicator for the q nested width m guarded k pebble game on $(\mathfrak{A}, \mathfrak{B})$. □

Proposition 5.18. *The complement of* CSP$(\mathfrak{B})$ *can be expressed in Datalog.*

PROOF. It is easy to see that the complement of CSP$(\mathfrak{B})$ can be defined by a Datalog program of width $(2, 3)$, because for a given τ-structure $\mathfrak{A}$ it suffices to

- compute the smallest reflexive, symmetric, and transitive binary relation T on all elements participating in R such that for every $(x, y, z) \in R^{\mathfrak{A}}$ with $(x, y) \in T$ we have $(y, z) \in T$:

$$
\begin{aligned}
T(y, z) &:- T(x, y), R(x, y, z) & \qquad T(x, x) &:- R(x, y, z) \\
T(y, x) &:- T(x, y) & T(y, y) &:- R(x, y, z) \\
T(x, z) &:- T(x, y), T(y, z) & T(z, z) &:- R(x, y, z),
\end{aligned}
$$

- and then to check whether T and D contain a common pair of elements or D has a loop:

$$
\text{goal} :- T(x, y), D(x, y) \qquad \qquad \text{goal} :- D(x, x). \qquad \qquad \qquad \square
$$

Corollary 5.19. *The intersection of GSO and Datalog contains problems that cannot be expressed in* GQ^+.

PROOF. Since GSO is closed under negation, Proposition 5.16 implies that the complement of CSP$(\mathfrak{B})$ is in GSO, and Proposition 5.17 shows that CSP$(\mathfrak{B})$ is not in GQ^+. □

We will see a strengthening of this result in Corollary 5.24.

5.7 Separation in MSO

In this section, we will prove that there are problems in MSO that can be expressed in Datalog, but that cannot be expressed in Nemodeq. The MSO sentence describes the complement of the CSP of a structure $\mathfrak{C}$ which is closely related to the structure $\mathfrak{B}$ from Section 5.6. Throughout this section, let $\mathfrak{C}$ be the $\{D, R\}$-structure $(C; D^{\mathfrak{C}}, R^{\mathfrak{C}})$

where

$$C := \mathbb{N} \,\dot{\cup}\, \mathbb{N}^3$$
$$D^{\mathfrak{C}} := \{(x, y) \in C^2 \mid x, y \in \mathbb{N}, \ x \neq y\}$$
$$R^{\mathfrak{C}} := \{(x, y, z, t) \in C^4 \mid x, y, z \in \mathbb{N}, \ t = \langle x, y, z \rangle, \ x = y \Rightarrow y = z\}.$$

That is, the domain C contains two sorts of elements: all natural numbers *and* all triples of natural numbers.[9] In words, the binary relation $D^{\mathfrak{C}}$ expresses disequality between natural numbers, whereas the quaternary relation $R^{\mathfrak{C}}$ consists of quadruples where the first three components are natural numbers while the last component is the triple built from the first three components. Among all such quadruples, $R^{\mathfrak{C}}$ contains those where the first three components are identical whenever the first two components are. An alternative formulation would be to say that $R^{\mathfrak{C}}$ contains all quadruples of the form $(x, y, z, \langle x, y, z \rangle)$ except those where $x = y$ and $y \neq z$.

Proposition 5.20. *The complement of* $\mathrm{CSP}(\mathfrak{C})$ *is in Datalog.*

Proof. The idea of the proof is analoguous to that of Proposition 5.18. Consider the following Datalog program Π

$$T(y, z) :- T(x, y), R(x, y, z, t) \qquad\qquad T(x, x') :- R(x, y, z, t), R(x', y', z', t)$$
$$T(y, x) :- T(x, y) \qquad\qquad T(y, y') :- R(x, y, z, t), R(x', y', z', t)$$
$$T(x, z) :- T(x, y), T(y, z) \qquad\qquad T(z, z') :- R(x, y, z, t), R(x', y', z', t)$$
$$\text{goal} :- T(x, y), D(x, y) \qquad\qquad T(x, x) :- D(x, y)$$
$$\text{goal} :- T(t, t), R(x, y, z, t) \qquad\qquad T(y, y) :- D(x, y)$$

Now assume $\mathfrak{A}$ is such that Π does not derive goal and let $\mathfrak{A}' = \Pi(\mathfrak{A})$. Let

$$A^* = \{a_1, a_2, a_3 \in A \mid (a_1, a_2, a_3, a_4) \in R^{\mathfrak{A}}\} \cup \{a_1, a_2 \in A \mid (a_1, a_2) \in D^{\mathfrak{A}}\}.$$

Thanks to $\mathfrak{A}' = \Pi(\mathfrak{A})$ and $\Pi(\mathfrak{A})$ not deriving goal, we observe the following:

(1) $T^{\mathfrak{A}'}$ is an equivalence relation on A^* and is disjoint from $D^{\mathfrak{A}}$,
(2) A^* is disjoint from $\{a_4 \in A \mid (a_1, a_2, a_3, a_4) \in R^{\mathfrak{A}}\}$,
(3) for any $(a_1, a_2, a_3, a) \in R^{\mathfrak{A}}$ and $(a'_1, a'_2, a'_3, a) \in R^{\mathfrak{A}}$, we have that a_i and a'_i are in the same $T^{\mathfrak{A}'}$-equivalence class for any $i \in \{1, 2, 3\}$.

We now define a homomorphism h from $\mathfrak{A}$ to $\mathfrak{C}$. Pick an injective function $h^* \colon A^*_{/T^{\mathfrak{A}'}} \to \mathbb{N}$ mapping $T^{\mathfrak{A}'}$-equivalence classes to natural numbers. For every $a \in A^*$, we let $h(a) = h^*([a]_{T^{\mathfrak{A}'}})$ and for every $a \in A \setminus A^*$, we let $h(a) \in \mathbb{N}^3$ be such that, whenever $(a_1, a_2, a_3, a) \in R^{\mathfrak{A}}$ for some $a_1, a_2, a_3 \in A^*$ then $h(a) = \langle h(a_1), h(a_2), h(a_3) \rangle$ (thanks to item 3 above, this choice is always possible). It can now be readily checked that h is a homomorphism from $\mathfrak{A}$ to $\mathfrak{C}$. □

We will now argue that $\mathrm{CSP}(\mathfrak{C})$ can be expressed in MSO. We first provide a characterisation for a given $\mathfrak{A}$ being in $\mathrm{CSP}(\mathfrak{C})$.

Lemma 5.21. *A finite* $\{D, R\}$*-structure* $\mathfrak{A}$ *is in* $\mathrm{CSP}(\mathfrak{C})$ *if and only if the following conditions are satisfied:*

(1) *There is a partition* (A', A'') *of* A *such that* A' *consists of exactly those elements of* A *occurring in the fourth position of some tuple from* $R^{\mathfrak{A}}$*, while* A'' *contains at least all elements of* A *occurring in* $D^{\mathfrak{A}}$ *or in any of the the first three positions of some tuple from* $R^{\mathfrak{A}}$*.*

---

[9] For the triples *inside C*, we will use the notation $\langle n_1, n_2, n_3 \rangle$, to better distinguish them from tuples *over C* which we will continue to enclose in round parentheses $(\ldots)$.

(2) *The binary relation $D^{\mathfrak{A}}$ is disjoint from the smallest equivalence relation $\approx$ over A'' that satisfies the following:*

    (2a) *For any $(a_1, b_1, c_1, d) \in R^{\mathfrak{A}}$ and $(a_2, b_2, c_2, d) \in R^{\mathfrak{A}}$, we have $a_1 \approx a_2$ as well as $b_1 \approx b_2$ and $c_1 \approx c_2$.*

    (2b) *For any $(a, b, c, d) \in R^{\mathfrak{A}}$ with $a \approx b$, we also have $b \approx c$.*

PROOF. For the "if" part, we provide a homomorphism $h : A \to C$ defined by

$$a \mapsto \begin{cases} f([a]_{\approx}) & \text{if } a \in A'' \\ \langle f([b]_{\approx}), f([c]_{\approx}), f([d]_{\approx}) \rangle & \text{whenever } (b, c, d, a) \in R^{\mathfrak{A}} \end{cases}$$

where $f \colon A''/_{\approx} \to \mathbb{N}$ is an arbitrary injection from the $\approx$-equivalence classes of A'' to the natural numbers. Note that (2a) ensures well-definedness of the second case.

For the "only if" part, assume a homomorphism h from $\mathfrak{A}$ to $\mathfrak{C}$. Without loss of generality we may assume h to be such that it maps any $a \in A$ not participating in any of the relations to an arbitrary element of $\mathbb{N}$. We then let $A'' = h^{-1}(\mathbb{N})$ and $A' = A \setminus A''$, which by assumption satisfy Condition (1). We define an equivalence relation $\mathring{\approx}$ over A'' by letting $a \mathring{\approx} b$ iff $h(a) = h(b)$. We observe that $\mathring{\approx}$ must be disjoint from $D^{\mathfrak{A}}$ and that $\approx \subseteq \mathring{\approx}$. Therefore $\approx$ must be disjoint from $D^{\mathfrak{A}}$, warranting Condition (2). □

We will construct an MSO sentence that describes $\mathrm{CSP}(\mathfrak{C})$ in a stepwise fashion. As a "witness" of the existence of a homomorphism h from $\mathfrak{A}$ to $\mathfrak{C}$, we will employ a set variable X which is supposed to correspond to the set

$$\{a \in A \mid h(a) = \langle n, n, n \rangle \in \mathbb{N}^3\}.$$

If X represented this set, $\mathfrak{A}$ would necessarily need to satisfy the MSO formula

$$\varphi_{\text{well}}(X) \coloneqq \varphi_{\text{sort}}(X) \wedge \varphi_{\text{triag}}(X) \wedge \varphi_{\text{compat}}(X)$$

whose components are specified as follows:

The first conjunct φ_{sort} prevents "type clashes" between numbers and triples (where the set variable T is used to identify the elements that h maps to triples):

$$\varphi_{\text{sort}}(X) \coloneqq \exists T. X \subseteq T \wedge \forall x, y, z, t \big(R(x, y, z, t) \Rightarrow \neg T(x) \wedge \neg T(y) \wedge \neg T(z) \wedge T(t)\big)$$
$$\wedge \big(D(x, y) \Rightarrow \neg T(x) \wedge \neg T(y)\big).$$

The second conjunct φ_{triag} ensures that X covers those $a \in A$ with $h(a) = \langle n_1, n_2, n_3 \rangle$ where $n_1 = n_2 = n_3$ is warranted because of $n_1 = n_2$ being known:

$$\varphi_{\text{triag}}(X) \coloneqq \forall x, y, z, t. R(x, y, z, t) \wedge \varphi_{\text{Eq}}(x, y, X) \Rightarrow X(t)$$

where $\varphi_{\text{Eq}}(x, y, X)$ is an MSO formula with two additional free first-order variables which (presuming the choice of X as above) holds for every pair $(a_1, a_2) \in A^2$ for which the above Datalog program would derive the Eq predicate:

$$\varphi_{\text{Eq}}(x, y, X) \coloneqq \forall Y. \big(Y(x) \wedge \varphi_{\text{propag}}(Y, X) \Rightarrow Y(y)\big)$$

with the following formula $\varphi_{\text{propag}}(Y, X)$ enforcing that Y is a set which is closed under "derived equalities":

$$\forall x_1, \ldots, x_6, t \big(R(x_1, x_2, x_3, t) \wedge R(x_4, x_5, x_6, t)\big) \Longrightarrow \bigwedge_{1 \leq i \leq 3} \big(Y(x_i) \Leftrightarrow Y(x_{i+3})\big)$$
$$\wedge \; \forall x_1, x_2, x_3, t \big(R(x_1, x_2, x_3, t) \wedge X(t)\big) \Longrightarrow \bigwedge_{1 \leq i < j \leq 3} \big(Y(x_i) \Leftrightarrow Y(x_j)\big).$$

Finally, the third conjunct $\varphi_{\text{compat}}(X)$ ensures that the "derivable equalities" are not in conflict with the disequalities imposed through D:

$$\varphi_{\text{compat}}(X) \coloneqq \forall x, y \big(\varphi_{\text{Eq}}(x, y, X) \Rightarrow \neg D(x, y)\big).$$

Proposition 5.22. *A finite $\{D, R\}$-structure $\mathfrak{A}$ is in $\mathrm{CSP}(\mathfrak{C})$ if and only if it satisfies the MSO sentence $\exists X. \varphi_{\text{well}}(X)$.*

PROOF. For the "only if" direction, we make a formal argument along the intuitions provided above. Assume a homomorphism h from $\mathfrak{A}$ to $\mathfrak{C}$. We show that the assignment $X \mapsto A^=$ with

$$A^= := \{a \in A \mid (n, n, n, h(a)) \in R^{\mathfrak{A}}, n \in \mathbb{N}\}$$

validates $\varphi_{\text{well}}(X)$:

It validates $\varphi_{\text{sort}}(X)$ because one can choose $T \mapsto h^{-1}(\mathbb{N}^3)$ whereas in order to satisfy the premises of any implication inside $\varphi_{\text{sort}}(X)$, all first order variables but t must be instantiated with elements of $h^{-1}(\mathbb{N})$.

Validation of $\varphi_{\text{triag}}(X)$ can be obtained from the fact that, for any $(a_x, a_y, a_z, a_t) \in R^{\mathfrak{A}}$ satisfying $h(a_x) = h(a_y)$ it must also hold that $h(a_y) = h(a_z)$ by virtue of h being a homomorphism into $\mathfrak{C}$ – where it remains to show that $\varphi_{\text{Eq}}(a_x, a_y, A^=)$ indeed implies $h(a_x) = h(a_y)$. To see that this is true, consider the case where Y is mapped to the set $A_x = h^{-1}(h(a_x))$. Then the implication's consequence indeed ensures $h(a_x) = h(a_y)$, provided that the premise is found to be true. The premise's first conjunct holds by construction, whence it remains to show that $\varphi_{\text{propag}}(A_x)$ also holds true, which – in view of our chosen assignment for X – again follows from the fact that h is a homomorphism into $\mathfrak{C}$.

Finally, validation of $\varphi_{\text{compat}}(X)$ is established by recalling that $\varphi_{\text{Eq}}(a_x, a_y, A^=)$ implies $h(a_x) = h(a_y)$ (see above) and therefore $(h(a_x), h(a_y)) \notin D^{\mathfrak{C}}$ by definition.

For the "if" direction, assume that some assignment $X \mapsto A_X \subseteq A$ validates $\varphi_{\text{well}}(X)$ for $\mathfrak{A}$. We will use Lemma 5.21 to argue that $\mathfrak{A}$ is in CSP($\mathfrak{C}$). Let $A'' = \{a_4 \mid (a_1, a_2, a_3, a_4) \in R^{\mathfrak{A}}\}$ and let $A' = A \setminus A''$. Then, by virtue of φ_{sort}, the partition (A', A'') satisfies Condition (1) of Lemma 5.21. Satisfaction of the Condition (2) would be an immediate consequence of $\varphi_{\text{compat}}(X)$ if one could establish that the binary relation $Eq_{A_X} := \{(a_1, a_2) \mid \varphi_{\text{Eq}}(a_1, a_2, A_X)\}$ contains $\approx$. This is achieved by arguing that Eq_{A_X} is an equivalence relation and satisfies the two subitems in Lemma 5.21 defining $\approx$. The former is a result of the structure of φ_{Eq} as an undirected reachability check. The latter is enforced through φ_{propag} and φ_{triag} which together ensure the two subconditions (2a) and (2b). $\qquad\square$

The proof of the following proposition is similar to the proof of Proposition 5.17.

Proposition 5.23. *The complement of* CSP($\mathfrak{C}$) *cannot be expressed in* GQ^+.

PROOF. Let $q, m, k \in \mathbb{N}$ be such that $q, m \geq 1$ and $k \geq m$. We construct an unsatisfiable instance $\mathfrak{A}$ of CSP($\mathfrak{C}$) such that Duplicator wins the q nested width m guarded k-pebble game on $(\mathfrak{A}, \mathfrak{C})$. Let ϕ_i' be as defined in the proof of Proposition 5.17, but replacing $R(x, y, z)$ by $R(x, y, z, t)$ where t is a fresh variable. Let $\mathfrak{A}$ be the canonical database of $\phi := \phi_{q+1}(x, y) \wedge D(x, y)$ after transforming it into prenex normal form. Similarly as in the proof of Proposition 5.17, it is straightforward to show that $\mathfrak{A}$ does not have a homomorphism to $\mathfrak{C}$. We need to show that Duplicator has a winning strategy in the q nested width m guarded k-pebble game on $(\mathfrak{A}, \mathfrak{C})$. Let $\mathcal{H}_0$ be the set of all partial homomorphisms h from substructures of $\mathfrak{A}$ to $\mathfrak{C}$ with domain size at most k. Let $\mathcal{H}_i$, for all $i \in \{1, \ldots, q\}$, be the set of all partial homomorphisms h from substructures of $\mathfrak{A}$ to $\mathfrak{C}$ with domain size at most k defined as follows. If $u, v \in \text{dom}(h)$ are variables of ϕ that do *not* appear as the last argument of any atomic formula $R(x, y, z, t)$ in ϕ, then we require that $h(u) = h(v)$ if and only if both u and v are part of the same subformula of $\phi_{q+1}(x, y) \wedge D(x, y)$ of the form ϕ_{i-1}, or $\text{dom}(h)$ contains all the variables $x_r, \ldots, x_s, x_{s+1}$ from ϕ_i for some $0 \leq r < s < k$ and u appears in the subformula $\phi_{i-1}(x_r, x_{r+1})$ and v appears in the subformula $\phi_{i-1}(x_s, x_{s+1})$. If $t, t' \in \text{dom}(h)$ are such that ϕ contains conjuncts $R(x, y, z, t)$ and $R(x', y', z', t')$, then these conjuncts are unique by the construction of ϕ; in this case, $h(t) = h(t')$ if and only if $h(x) = h(x')$, $h(y) = h(y')$, and $h(z) = h(z')$. Otherwise, $h(t)$ must be distinct from all other points in the image of h.

Since $\text{dom}(h) \leq k$ it is impossible that $r = 0$ and $s = k - 1$; hence, if x_0 and x_k are from $\text{dom}(h)$, then $h(x_0) \neq h(x_k)$. Clearly, $\mathcal{H}_i$ is closed under restrictions and non-empty. Now suppose that $h \in \mathcal{H}_i$ is such that there exists a tuple a in a relation of $\mathfrak{A}$ whose entries contain $\text{dom}(h)$. Let $S \subseteq A$ be a superset of $\text{dom}(h)$ of

cardinality at most k. If this relation is D, then a equals (x, y) and we may clearly extend h to some $h' \in \mathcal{H}_i$ with $h'(x) \neq h'(y)$. Otherwise, the relation must be R and a equals (x_0, x_k, y, t) for some variables x_0, x_k, y, t from a subformula ϕ of the form $\phi_j(x, y)$ for some $j \leq i$. Note that $h(x_0) = h(x_k) = h(y)$ if $j < i$, and $h(x_0) \neq h(x_k)$ and $h(x_k) \neq h(y)$ otherwise. It is easy to see that in both cases we may extend h to some $h' \in \mathcal{H}_i$ with domain S. Finally, if $h \in \mathcal{H}_i$ for $i \in \{1, \ldots, q\}$ and $S \subseteq A$ with $|S| \leq k$ is such that $\mathrm{dom}(h) \subseteq S$, then $\mathcal{H}_{i-1}$ contains an extension h' of h with domain S: if $x \in S$ appears in a copy of ϕ_{i-1} whose variables intersect $\mathrm{dom}(h)$, we set $h'(x)$ to the value taken by these variables under h. If $\mathrm{dom}(h)$ contains all the variables $x_r, \ldots, x_s, x_{s+1}$ from ϕ_i for some $0 \leq r < s < k$ and x appears in the subformula $\phi_{i-1}(x_s, x_{s+1})$ and some variable of $\phi_{i-1}(x_r, x_{r+1})$ appears in $\mathrm{dom}(h)$, then we set $h'(x)$ to the value taken by this variable under h. Otherwise, pick $h'(x)$ to be a new value. Hence, $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_q$ is indeed a winning strategy for Duplicator for the q nested width m guarded k pebble game on $(\mathfrak{A}, \mathfrak{C})$. □

Note that the following is a strengthening of Corollary 5.19.

Corollary 5.24. *The intersection of MSO and Datalog contains problems that cannot be expressed in GQ^+.*

PROOF. We already know that the complement of $\mathrm{CSP}(\mathfrak{C})$ is in Datalog (Proposition 5.20), but not in GQ^+ (Proposition 5.23). Proposition 5.22 shows that $\mathrm{CSP}(\mathfrak{B})$ is in MSO; then, as MSO is closed under negation, the complement of $\mathrm{CSP}(\mathfrak{B})$ is MSO-expressible as well. □

6 A CONP-COMPLETE CSP IN MSO

How tame is the intersection of MSO and Datalog? We have seen that every CSP which is in MSO and in Datalog can be expressed as the CSP of an ω-categorical structure (Corollary 3.8). It would be interesting to know whether this can be strengthened further by showing that these problems can even be expressed as CSPs for a particularly well-behaved class of structures, namely for *reducts of finitely bounded homogeneous structures* (the definition of this class can be found below). It has been conjectured that every CSP for a structure from this class is in P or NP-complete [20]; we will refer to this conjecture as the *infinite-domain dichotomy conjecture*. This conjecture generalises the finite-domain complexity dichotomy that was conjectured by Feder and Vardi [32] and proved by Bulatov [22] and by Zhuk [46].

Definition 15. *A relational structure is called* homogeneous *if every isomorphism between finite substructures can be extended to an automorphism.*

Homogeneous structures with a *finite* relational signature (and all of their reducts) are ω-categorical. Reducts $\mathfrak{B}$ of homogeneous structures with finite relational signature are a particularly well-behaved subclass of the class of ω-categorical structures. See [4] for an example of a result that only holds for such $\mathfrak{B}$, but not for ω-categorical structures in general. However, there are uncountably many countable homogeneous digraphs with pairwise distinct CSP, and it follows that there are homogeneous digraphs with undecidable CSPs [19].

To define a class of structures whose CSPs have better computational properties we need the following concept. The *age* of a structure $\mathfrak{B}$ is the class of all finite structures that embed into $\mathfrak{B}$. A structure $\mathfrak{B}$ is called *finitely bounded* if there exists a finite set $\mathcal{F}$ of finite structures such that a finite structure $\mathfrak{A}$ belongs to the age of $\mathfrak{B}$ if and only if no structure in $\mathcal{F}$ embeds into $\mathfrak{A}$.

Example 6.1. *The structure $\mathfrak{B}$ from Section 5.6 is finitely bounded and homogeneous.* △

Clearly, the CSP of finitely bounded structures $\mathfrak{B}$ is in NP. It is also easy to see that every CSP of a reduct of a finitely bounded structure is in NP as well.

Question 6.2. *Is it true that every CSP in MSO is the CSP for a reduct of a finitely bounded homogeneous structure?*

In this section, we do not answer this question, but at least we can 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 3.1). In particular, the class of CSPs that can be expressed in MSO does not fall into the scope of the mentioned infinite-domain CSP dichotomy conjecture.

A *tournament* is a directed graph such that for any pair of vertices u and v, there is an edge from u to v or an edge from v to u, but not both. 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 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 C. It has been shown in [16] that CSP($\mathfrak{H}$) is coNP-complete.

Proposition 6.3. CSP($\mathfrak{H}$) *can be expressed in MSO.*

PROOF. We have to find an MSO sentence Φ 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 $\mathfrak{S}$ if and only if $(X^{\mathfrak{S}}; E^{\mathfrak{S}})$ 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 $\forall x \colon \neg E(x, x) \wedge \forall X \colon \neg \Phi$ 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: we set s to 0, t to $n + 1$, we set A to the set of even and B to the set of odd numbers in $\{1, \dots, n + 1\}$. 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 $f(s) \coloneqq 0$, $f(a) \coloneqq 1$, 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 $f(a') \coloneqq 3$. 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 $f(b') \coloneqq 4$. Continuing in this manner, we eventually define f on all of X and find an isomorphism with $\mathfrak{T}_{|X|-1}$. □

This shows that, unless NP = coNP, CSP($\mathfrak{H}$) cannot be expressed as CSP($\mathfrak{B}$) for some reduct of a finitely bounded structure, because such CSPs are in NP.

Corollary 6.4. *If NP ≠ coNP, then there are CSPs in MSO that cannot be expressed as CSPs of finitely bounded structures.*

We do not know how to show this statement without complexity-theoretic assumptions, even if we just want to rule out that $CSP(\mathfrak{H})$ can be expressed as $CSP(\mathfrak{B})$ for some reduct of a finitely bounded *homogeneous* structure. However, using the tools from Section 4.2 we can rule out that the complement of $CSP(\mathfrak{H})$ can be expressed in Datalog. Note that this insight is obtained without resorting to any complexity-theoretic assumptions.

THEOREM 6.5. *Let Φ be the MSO sentence from the proof of Proposition 6.3. Then $[\![\Phi]\!]$ cannot be expressed in Datalog.*

PROOF. We use Theorem 4.6; it suffices to find for all ℓ, k a finite model $\mathfrak{A}$ of Φ such that Duplicator wins the existential (ℓ, k)-pebble game for $[\![\Phi]\!]$ on $\mathfrak{A}$. Simply choose $\mathfrak{A}$ to be the tournament $\mathfrak{T}_{k+1}$; then clearly $\mathfrak{A} \models \neg\Phi$. Note that $CSP(\mathfrak{H}) \cap [\![\Phi]\!] = \emptyset$, so it suffices to find a winning strategy $\mathcal{H}$ for Duplicator for the existential (ℓ, k)-pebble game on $(\mathfrak{A}, \mathfrak{H})$.

Let $\mathcal{H}$ be the set of all partial embeddings h from $\mathfrak{A}$ to $\mathfrak{B}$ such that $\text{dom}(h) \leq k$. Then $\mathcal{H}$ is clearly non-empty and closed under restrictions. To verify that $\mathcal{H}$ is a winning strategy for Duplicator we still need to show that for all $h \in \mathcal{H}$ with $\text{dom}(h) = \{a_{k-d+1}, \dots, a_k\}$ and for all $a_1, \dots, a_{k-d} \in A$ there is an extension $h' \in \mathcal{H}$ of h such that h' is also defined on $a_1, \dots, a_{k-d}$. Let $\mathfrak{A}'$ be the substructure of $\mathfrak{A}$ with the domain $\{a_1, \dots, a_k\}$. Since $|A'| < |A|$, there exists an embedding of $\mathfrak{A}'$ into $\mathfrak{H}$; we identify the elements of A' with their image under this embedding, so that $\mathfrak{A}'$ is now a substructure of $\mathfrak{H}$, and h is an isomorphism between finite substructures of $\mathfrak{H}$. By the homogeneity of $\mathfrak{H}$, there exists an automorphism α of $\mathfrak{H}$ which extends h. Then the restriction of α to $a_1, \dots, a_k$ is the desired extension h' of h in $\mathcal{H}$. □

7 CONCLUSION, OPEN PROBLEMS, AND PROSPECT

In this article, we provided a game-theoretic characterisation of the 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.

Our results imply that the so-called universal-algebraic approach, which has eventually led to the classification of finite-domain CSPs in Datalog [5], can be applied to study problems that are simultaneously in GSO and in Datalog (see [41]). They might also pave the way towards a syntactic characterisation of GSO ∩ Datalog and of MSO ∩ Datalog; however, we showed that the class of nested guarded queries from [21] is not expressive enough for this purpose. See Figure 4 for an overview of the considered logics, their inclusions, and references to examples that show that the inclusions are strict.

We state some concrete open problems in the context.

(1) Is there a *finite* structure whose CSP is in Datalog, but not in Nemodeq? Also the expressive power of frontier-guarded Datalog, GQ, and GQ⁺ for finite-domain CSPs appears to be unexplored.
(2) It is known that *singleton linear arc consistency (SLAC)* [39] captures the intersection of finite-domain CSPs with Datalog. It would be interesting to define an appropriate notion of singleton linear arc consistency for CSPs for ω-categorical structures; can it be used to precisely characterise the intersection of MSO and Datalog?
(3) Is there a CSP of a reduct of a finitely bounded homogeneous structure which is not expressible in GSO?
(4) Question 6.2: Is every CSP in MSO ∩ Datalog the CSP of a reduct of a finitely bounded homogeneous structure?
(5) Is every CSP in existential MSO also in MMSNP?

SO

$\mathrm{CSP}(K_3)$ | Expl. 3.17

Thm. 5.9

Datalog   GSO

Expl. 3.17 | Prop. 2.4

Prop. 5.11

GQ^+ MSO

Prop. 2.4 | $\mathrm{CSP}(K_3)$

Prop. 5.15   [43]

GQ   Nemodeq

Prop. 2.4

Prop. 5.3 | Prop. 5.15

Frontier-Guarded Datalog   Modeq
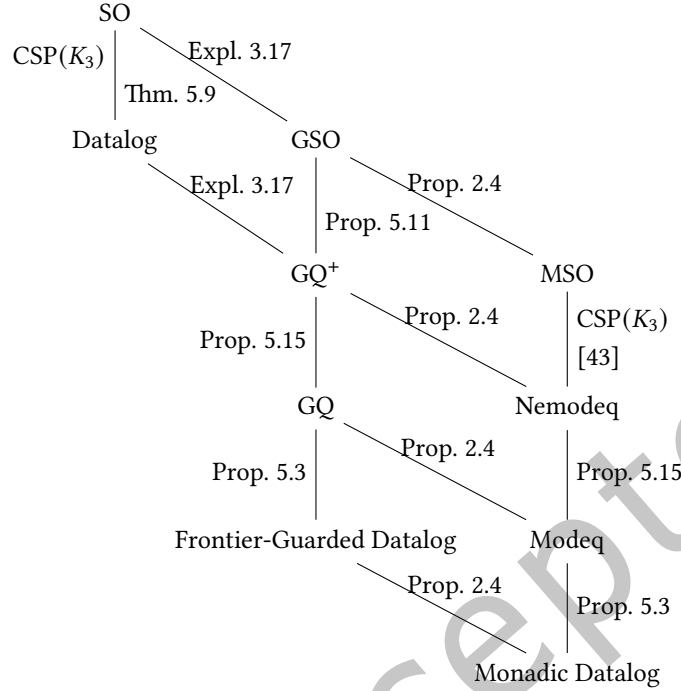
Prop. 2.4

Prop. 5.3

Monadic Datalog

Fig. 4. A Hasse Diagram of the logical formalisms considered in this article, ordered by expressive power. The red labels give references to propositions or examples in this text that show that the respective inclusion is strict.

We are also confident that our results will advance the understanding of CSPs (the complements of) which are obtained as the homomorphism-closure of the set of some theory's finite models. For example, the homomorphism-closures of the model sets of guarded- and guarded-negation-theories have recently been found to be GSO-expressible [15] so, by virtue of our results, we immediately know they must be (complements of) ω-categorical CSPs.

8 ACKNOWLEDGMENTS

REFERENCES

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. URL: http://webdam.inria.fr/Alice/.

[2] Hajnal Andréka, István Németi, and Johan van Benthem. Modal languages and bounded fragments of predicate logic. *J. Philos. Log.*, 27(3):217–274, 1998. doi:10.1023/A:1004275029985.

[3] Albert Atserias, Andrei A. Bulatov, and Anuj Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410(18):1666–1683, 2009.

[4] Libor Barto, Michael Kompatscher, Miroslav Olšák, Trung Van Pham, and Michael Pinsker. The equivalence of two dichotomy conjectures for infinite domain constraint satisfaction problems. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science – LICS'17*, 2017. Preprint arXiv:1612.07551.

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

[6] Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter. Ontology-based data access: A study through disjunctive datalog, csp, and MMSNP. *ACM Transactions on Database Systems*, 39(4):33:1–33:44, 2014. doi:10.1145/2661643.

[7] Achim Blumensath. Monadic second-order logic. Lecture Notes, 2020.

[8] Achim Blumensath. Monadic Second-Order Model Theory. Lecture Notes, https://www.fi.muni.cz/ blumens/MSO2.pdf, 2025.

[9] Manuel Bodirsky. Constraint satisfaction with infinite domains. Dissertation, Humboldt-Universität zu Berlin, 2004.

[10] Manuel Bodirsky. *Complexity of Infinite-Domain Constraint Satisfaction*. Lecture Notes in Logic (52). Cambridge University Press, Cambridge, United Kingdom; New York, NY, 2021. doi:10.1017/9781107337534.

[11] Manuel Bodirsky and Hubie Chen. Peek arc consistency. *Theor. Comput. Sci.*, 411(2):445–453, 2010.

[12] Manuel Bodirsky and Hubie Chen. Quantified equality constraints. *SIAM Journal on Computing*, 39(8):3682–3699, 2010. A preliminary version appeared in the proceedings of LICS'07.

[13] Manuel Bodirsky, Hubie Chen, and Michael Pinsker. The reducts of equality up to primitive positive interdefinability. *Journal of Symbolic Logic*, 75(4):1249–1292, 2010. doi:10.2178/jsl/1286198146.

[14] Manuel Bodirsky and Víctor 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).

[15] Manuel Bodirsky, Thomas Feller, Simon Knäuer, and Sebastian Rudolph. On logics and homomorphism closure. In *Proceedings of the Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE, 2021. Preprint https://arxiv.org/abs/2104.11955.

[16] Manuel Bodirsky and Martin Grohe. Non-dichotomies in constraint satisfaction complexity. In Luca Aceto, Ivan Damgard, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, 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.

[17] 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.

[18] Manuel Bodirsky, Simon Knäuer, and Sebastian Rudolph. Datalog-expressibility for monadic and guarded second-order logic. In *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, pages 120:1–120:17, 2021. Preprint available at https://arxiv.org/abs/2010.05677. doi:10.4230/LIPIcs.ICALP.2021.120.

[19] Manuel Bodirsky and Jaroslav Nešetřil. Constraint satisfaction with countable homogeneous templates. *Journal of Logic and Computation*, 16(3):359–373, 2006. doi:10.1093/logcom/exi083.

[20] Manuel Bodirsky, Michael Pinsker, and András Pongrácz. Projective clone homomorphisms. *Journal of Symbolic Logic*, 86(1):148–161, 2021.

[21] 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.

[22] 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.

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

[24] Catarina Carvalho, Víctor Dalmau, and Andrei Krokhin. CSP duality and trees of bounded pathwidth. *Theoretical Computer Science*, 411:3188–3208, 2010.

[25] Hubie Chen, Víctor Dalmau, and Berit Grußien. Arc consistency and friends. *J. Log. Comput.*, 23(1):87–108, 2013.

[26] Gregory Cherlin, Saharon Shelah, and Niangdong Shi. Universal graphs with forbidden subgraphs and algebraic closure. *Advances in Applied Mathematics*, 22:454–491, 1999.

[27] 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.

[28] 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.

[29] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. *ACM Comput. Surv.*, 33(3):374–425, 2001. doi:10.1145/502807.502810.

[30] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. doi:10.1007/978-1-4612-0515-9.

[31] 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.

[32] 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.

[33] Cristina Feier, Antti Kuusisto, and Carsten Lutz. Rewritability in monadic disjunctive datalog, mmsnp, and expressive description logics. *Logical Methods in Computer Science*, 15(2), 2019. doi:10.23638/LMCS-15(2:15)2019.

[34] Thomas Feller, Tim S. Lyon, Piotr Ostropolski-Nalewaja, and Sebastian Rudolph. Decidability of querying first-order theories via countermodels of finite width. *CoRR*, abs/2304.06348, 2023. accepted for publication at LMCS. arXiv:2304.06348, doi:10.48550/arXiv.2304.06348.

[35] Thomas Feller, Tim S. Lyon, Piotr Ostropolski-Nalewaja, and Sebastian Rudolph. Finite-cliquewidth sets of existential rules: Toward a general criterion for decidable yet highly expressive querying. In Floris Geerts and Brecht Vandevoort, editors, *26th International Conference on Database Theory, ICDT 2023, March 28-31, 2023, Ioannina, Greece*, volume 255 of *LIPIcs*, pages 18:1–18:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICDT.2023.18.

[36] Erich Grädel, Colin Hirsch, and Martin Otto. Back and forth between guarded and modal logics. *ACM Trans. Comput. Log.*, 3(3):418–463, 2002.

[37] Wilfrid Hodges. *A shorter model theory*. Cambridge University Press, Cambridge, 1997.

[38] 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.

[39] Marcin Kozik. Weak consistency notions for all the CSPs of bounded width. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 633–641, 2016.

[40] Leonid Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[41] Antoine Mottet, Tomás Nagy, Michael Pinsker, and Michal Wrona. Smooth approximations and relational width collapses. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 138:1–138:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.138.

[42] Benjamin Rossman. Homomorphism preservation theorems. *Journal of the ACM*, 55(3), 2008.

[43] 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.

[44] Balder ten Cate, Víctor Dalmau, and Jakub Oprsal. Right-adjoints for datalog programs. In Graham Cormode and Michael Shekelyan, editors, *27th International Conference on Database Theory, ICDT 2024, March 25-28, 2024, Paestum, Italy*, volume 290 of *LIPIcs*, pages 10:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[45] Dmitriy Zhuk and Barnaby Martin. The complete classification for quantified equality constraints. *CoRR*, abs/2104.00406, 2021. URL: https://arxiv.org/abs/2104.00406, arXiv:2104.00406.

[46] 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.