LINEAR DATALOG AND BOUNDED PATH DUALITY OF RELATIONAL STRUCTURES

VÍCTOR DALMAU

Departament de Tecnologia, Universitat Pompeu Fabra,, Estació de França, Passeig de la Circum-val.lacio 8. Barcelona 08003, Spain, Fax: +34 93 542 24 49
e-mail address: victor.dalmau@tecn.upf.es

ABSTRACT. In this paper we systematically investigate the connections between logics with a finite number of variables, structures of bounded pathwidth, and linear Datalog Programs. We prove that, in the context of Constraint Satisfaction Problems, all these concepts correspond to different mathematical embodiments of a unique robust notion that we call bounded path duality. We also study the computational complexity implications of the notion of bounded path duality. We show that every constraint satisfaction problem CSP(**B**) with bounded path duality is solvable in NL and that this notion explains in a uniform way all families of CSPs known to be in NL. Finally, we use the results developed in the paper to identify new problems in NL.

1. INTRODUCTION

The constraint satisfaction problem provides a framework in which it is possible to express, in a natural way, many combinatorial problems encountered in artificial intelligence and elsewhere. A constraint satisfaction problem is represented by a set of variables, a domain of values for each variable, and a set of constraints between variables. The aim of a constraint satisfaction problem is then to find an assignment of values to the variables that satisfies the constraints.

Solving a general constraint satisfaction problem is known to be NP-complete [3, 29]. One of the main approaches pursued by researchers in artificial intelligence and computational complexity to tackle this problem has been the identification of tractable cases obtained by imposing restrictions in the constraints (see [4, 6, 9, 10, 12, 13, 21, 22, 23, 30, 31, 37, 38]).

Recently [12] (see also [20]) it has been observed that the constraint satisfaction problem can be recast as the following fundamental algebraic problem: given two finite relational structures **A** and **B**, is there a homomorphism from **A** to **B**? In this framework, the problem of identifying which restrictions in the constraints guarantee tractability is equivalent to

deciding for which structures $\mathbf{B}$, the homomorphism problem when only $\mathbf{A}$ is part of the input, denoted by CSP($\mathbf{B}$), is solvable in polynomial time. In this paper we study this framework.

One of the simplest examples of constraint satisfaction problems of this form is obtained when $\mathbf{B}$ is fixed to be a k-clique $\mathbf{K}_k$ for some $k \geq 2$. In this case, an instance CSP($\mathbf{K}_k$) is again a graph $\mathbf{A}$ and the question of existence of an homomorphism from $\mathbf{A}$ to $\mathbf{B}$ is equivalent to deciding whether $\mathbf{A}$ is k-colorable.

Feder and Vardi [12] introduced one general condition for tractability of CSP($\mathbf{B}$) that accounts for many of the tractable cases of CSP($\mathbf{B}$). More precisely, Feder and Vardi observed that for many polynomial-time solvable Constraint Satisfaction Problems of the form CSP($\mathbf{B}$) there is a *Datalog Program* that defines the complement of CSP($\mathbf{B}$).

In order to illustrate this consider now the particular case when $\mathbf{B}$ is $\mathbf{K}_2$, that is, $\mathbf{B}$ is a graph with two nodes and an edge between them. As it has been observed before, CSP($\mathbf{K}_2$) is the set of all 2-colorable graphs. It is well-known that a graph is 2-colorable iff it does not have odd cycles. The following Datalog Program asserts that the graph $\mathbf{A} = (V, E)$ contains an odd cycle:

$$
\begin{aligned}
P(x,y) &:-\ E(x,y) \\
P(x,y) &:-\ P(x,z), E(z,u), E(u,y) \\
Q &:-\ P(x,x)
\end{aligned}
$$

This Datalog Programs has three rules. These rules are a recursive specification of two predicates P, and Q, called IDBs. Predicate $P(x,y)$ holds whether there exists a path in $\mathbf{A}$ of odd length from x to y and predicate Q, which acts as goal predicate, holds if there exists a cycle of odd length .

Since the seminal results of Feder and Vardi, the language Datalog has played a prominent role in the study of the complexity of the CSP. In particular, several connections between Datalog Programs and well-established notions developed in the area of constraint satisfaction problems and graph homomorphism have been explored.

One of this notions is that of bounded treewidth duality initially introduced in the area of $\mathbf{H}$-coloring [17], which can be reformulated as the constraint satisfaction problem CSP($\mathbf{H}$) where $\mathbf{H}$ is a graph. In a $\mathbf{H}$-coloring problem we are given a graph $\mathbf{G}$ as an input and we are asked where there exists an homomorphism from $\mathbf{G}$ to $\mathbf{H}$. It has been observed [17] that the vast majority of the tractable cases of CSP($\mathbf{H}$) have an obstruction set of bounded treewidth. An obstruction set of CSP($\mathbf{H}$) is any set S of graphs not homomorphic to $\mathbf{H}$ such that for every graph $\mathbf{G}$ not homomorphic to $\mathbf{H}$ there exist a graph in S that is homomorphic to $\mathbf{G}$. Every graph $\mathbf{H}$ having an obstruction set of bounded treewidth is said to have bounded treewidth duality. Bounded treewidth duality turns out to be equivalent to definability in Datalog, as shown in [12]. More precisely, a graph $\mathbf{H}$, has bounded treewidth duality if and only if the complement of CSP($\mathbf{H}$) is definable in Datalog. The relationship between Constraint Satisfaction Problems, Datalog and structures of bounded treewidth has been further investigated in [27], [25], and [8].

The goal of the present paper is to take a closer look inside the structure of Datalog Programs. In particular, we are interested in *linear* Datalog Programs [1], which are Datalog Programs in which every rule has at most one IDB in the body. Many constraint satisfaction problems solvable with a Datalog Program are indeed solvable by a linear Datalog Program. In fact, the Datalog Program defining NON-2-COLORABILITY presented above is a linear Datalog Program. It is well known that problems solvable by a linear Datalog Program are

solvable in non-deterministic logarithmic space. This class has received a lot of interest in complexity theory. In particular, it is known that $NL \subseteq NC$ and, therefore, problems in NL are highly parallelizable.

In this paper we embark on a systematic study of the relationship between linear Datalog Programs, finite variable logics and structures of bounded pathwidth. We prove several different but equivalent characterizations of definability in linear Datalog. Most of them are in the realm of logic. In particular, we show that for every constraint satisfaction problem of the form $CSP(\mathbf{B})$, $\neg CSP(\mathbf{B})$ is definable in linear Datalog if and only if $CSP(\mathbf{B})$ is definable in the logic M^ω, which is defined to be the subset of the infinitary logic $L^\omega_{\infty,\omega}$ in which only existential quantification, infinitary disjunction and a certain restricted infinitary conjunction is allowed. Definability in linear Datalog is also shown to be equivalent to definability in restricted Krom SNP, which is defined to be the set of all existential second order sentences with a universal first-order part in prenex CNF form, in which we require every clause of the first-order part to contain only negated relation symbols and at most one positive and one negative second order-variable. We also obtain a combinatorial reformulation of definability in linear Datalog. More precisely, we show that for every structure $\mathbf{B}$, the complement of $CSP(\mathbf{B})$ is definable in linear Datalog if and only if $CSP(\mathbf{B})$ has an obstruction set of bounded pathwidth. All of these different but equivalent reformulations of the same notion seem to provide some evidence that the class of problems definable is linear Datalog is an interesting and robust class.

Finally we use our results to investigate which constraints satisfaction problems are in NL. Despite the large amount of tractable cases of constraint satisfaction problems identified so far, very few subclasses of constraint satisfaction problems are known to be in NL. To our knowledge, the only families of CSP problems known to be in NL are the class of bijunctive satisfiability problems [34], including 2-SAT, which later on was generalized to the class of implicational constraints [23] (see also [20]), the class of implicative Hitting-Set Bounded (see [5]) originally defined (although with a different name) in [34], and the class of posets with constants invariant under a near-unanimity operation [28]. First we observe that all this families of problems are particular cases of bounded path duality problems. Consequently, the results in our paper provide an uniform explanation of all known constraint satisfaction problems known to be in NL. Finally, we identify some new families of constraint satisfaction problems solvable in NL.

2. Basic Definitions

Most of the terminology introduced in this section is fairly standard. We basically follow [15]. A *vocabulary* is a finite set of relation symbols or predicates. In the following τ always denotes a vocabulary. Every relation symbol R in τ has an *arity* $r = \rho(R) \geq 0$ associated to it. We also say that R is an r-ary relation symbol.

A τ-structure $\mathbf{A}$ consists of a set A, called the *universe* of $\mathbf{A}$, and a relation $R^{\mathbf{A}} \subseteq A^r$ for every relation symbol $R \in \tau$ where r is the arity of R. Unless otherwise stated we will assume that we are dealing with *finite* structures, i.e., structures with a finite universe. Throughout the paper we use the same boldface and slanted capital letters to denote a structure and its universe, respectively.

Let $\mathbf{A}$ and $\mathbf{B}$ be τ-structures. We say that $\mathbf{B}$ is a substructure of $\mathbf{A}$, denoted by $\mathbf{B} \subseteq \mathbf{A}$, if $B \subseteq A$ and for every $R \in \tau$, $R^{\mathbf{B}} \subseteq R^{\mathbf{A}}$. If $\mathbf{A}$ is a τ-structure and $B \subseteq A$, then $\mathbf{A}_{|B}$ denotes the substructure induced by $\mathbf{A}$ on B, i.e., the τ-structure $\mathbf{B}$ with universe B and $R^{\mathbf{B}} = R^{\mathbf{A}} \cap B^r$ for every r-ary $R \in \tau$.

Let $\mathbf{A}$ and $\mathbf{B}$ be τ-structures. We denote by $\mathbf{A} \cup \mathbf{B}$ the τ-structure with universe $A \cup B$ and such that for all $R \in \tau$, $R^{\mathbf{A} \cup \mathbf{B}} = R^{\mathbf{A}} \cup R^{\mathbf{B}}$.

A *homomorphism* from a τ-structure $\mathbf{A}$ to a τ-structure $\mathbf{B}$ is a mapping $h : A \to B$ such that for every r-ary $R \in \tau$ and every $\langle a_1, \ldots, a_r \rangle \in R^{\mathbf{A}}$, we have $\langle h(a_1), \ldots, h(a_r) \rangle \in R^{\mathbf{B}}$. We denote this by $\mathbf{A} \xrightarrow{h} \mathbf{B}$. We say that $\mathbf{A}$ homomorphically maps to $\mathbf{B}$, and denote this by $\mathbf{A} \longrightarrow \mathbf{B}$ iff there exists some homomorphism from $\mathbf{A}$ to $\mathbf{B}$. We denote by $\mathrm{hom}(\mathbf{A}, \mathbf{B})$ the set of all homomorphisms from $\mathbf{A}$ to $\mathbf{B}$.

We will assume by convention that for every set B there exists one mapping $\lambda : \emptyset \to B$. Consequently, if $\mathbf{A}$ is a structure with an empty universe then $\{\lambda\} = \mathrm{hom}(\mathbf{A}, \mathbf{B})$.

Let $a_1, \ldots, a_m$ be elements in A and let $b_1, \ldots, b_m$ be elements in B. We shall write $\mathbf{A}, a_1, \ldots, a_m \longrightarrow \mathbf{B}, b_1, \ldots, b_m$ to denote that there exists some homomorphism h from $\mathbf{A}$ to $\mathbf{B}$ such that $h(a_i) = b_i$, $1 \le i \le m$.

Let $\mathbf{A}$ be a τ structure and let $\tau' \subseteq \tau$. We denote by $\mathbf{A}[\tau']$ the τ'-structure such that for every $R \in \tau'$, $R^{\mathbf{A}[\tau']} = R^{\mathbf{A}}$. Similarly, if $\mathcal{C}$ is a collection of τ-structures we denote by $\mathcal{C}[\tau']$ the set $\{\mathbf{A}[\tau'] : \mathbf{A} \in \mathcal{C}\}$. STR denotes the class of all structures and consequently, $\mathrm{STR}[\tau]$ denotes the class of all τ-structures.

Finally, $\mathrm{CSP}(\mathbf{B})$ is defined to be the set of all structures $\mathbf{A}$ such that $\mathbf{A} \longrightarrow \mathbf{B}$.

3. Infinitary Logic

The following definition is borrowed from [24]. Let τ be a vocabulary containing only relational symbols and let $\{v_1, \ldots, v_n, \ldots\}$ be a countable set of variables. The class $L_{\infty,\omega}$ of *infinitary formulas* over τ is the smallest collection of formulas such that

- it contains all first-order formulas over τ.
- if φ is a formula of $L_{\infty,\omega}$ then so is $\neg\varphi$.
- if φ is a formula of $L_{\infty,\omega}$ and v_i is a variable, then $(\forall v_i)\varphi$ and $(\exists v_i)\varphi$ are also formulas of $L_{\infty,\omega}$.
- if Ψ is a set (possibly infinite) of $L_{\infty,\omega}$ formulas, then $\bigvee \Psi$ and $\bigwedge \Psi$ are also formulas of $L_{\infty,\omega}$.

The first subscript of $L_{\infty,\omega}$ indicates that conjunctions and disjunctions can be taken over arbitrary infinite sets and the second that only finite quantifier blocks are allowed.

The concept of a *free variable* in a formula of $L_{\infty,\omega}$ is defined in the same way as for first-order logic. We use the notation $\varphi(u_1, \ldots, u_m, \ldots)$ to denote that $u_1, \ldots, u_m, \ldots$ are different and that φ is a formula of $L_{\infty,\omega}$ whose free variables are among the variables $u_1, \ldots, u_m, \ldots$. A *sentence* of $L_{\infty,\omega}$ is a formula φ of $L_{\infty,\omega}$ with no free variables. The semantics of $L_{\infty,\omega}$ is a direct extension of the semantics of first-order logic, with $\bigvee \Psi$ interpreted as a disjunction over all formulas in Ψ and $\bigwedge \Psi$ interpreted as a conjunction. If $\mathbf{A}$ is a structure over τ and $a_1, \ldots, a_m, \ldots$ is a sequence of (not necessarily different) elements from the universe of $\mathbf{A}$, then we write

$$\mathbf{A}, a_1, \ldots, a_m, \ldots, \models \varphi(u_1, \ldots, u_m, \ldots)$$

to denote that the structure $\mathbf{A}$ *satisfies* the formula φ of $L_{\infty,\omega}$ when each variable u_i is interpreted by the element a_i.

Let k be a non-negative integer. The *infinitary logic with k variables*, denoted by $L_{\infty,\omega}^k$ consists of all formulas of $L_{\infty,\omega}$ with at most k distinct variables. Let L^k be the collection of all formulas of $L_{\infty,\omega}^k$, that are obtained from atomic formulas using infinitary disjunctions, infinitary conjunctions, and existential quantification only.

Let j be a non-negative integer. We will denote by *j-restricted infinitary conjunction* the infinitary conjunction $\bigwedge \Psi$ when Ψ is a collection of $L_{\infty,\omega}$ formulas such that (a) every formula with more than j free variables is quantifier-free and (b) at most one formula in Ψ having quantifiers is not a sentence. If furthermore the set Ψ is finite then we will call it *j-restricted conjunction*.

Let $0 \leq j \leq k$ be non-negative integers. Let $M^{j,k}$ ($N^{j,k}$) be the collection of all formulas of $L_{\infty,\omega}^k$, that are obtained from atomic formulas using infinitary disjunction, j-restricted infinitary conjunction (j-restricted conjunction), and existential quantification only. Finally, let $O^{j,k}$ be the collection of all formulas of $L_{\infty,\omega}^k$, that are obtained from atomic formulas using j-restricted conjunction, and existential quantification only. We also put

$$M^\omega = \bigcup_{0 \leq j \leq k} M^{j,k}, \ \ N^\omega = \bigcup_{0 \leq j \leq k} N^{j,k}, \ \ \text{and } O^\omega = \bigcup_{0 \leq j \leq k} O^{j,k}$$

The following example illustrates the expressive power of these logics.

Example 1 (Paths, Bipartiteness). Assume that the vocabulary consists of a unique binary relation E, and let $\varphi_n(x,y)$, $n \geq 1$ be the first order formula asserting that there exists a path or length n from x to y. The naive way to write $\varphi_n(x,y)$ requires $n+1$ variables, namely

$$\exists x_1 \exists x_2 \ldots \exists x_{n-1} E(x,x_1) \wedge E(x_1,x_2) \wedge \cdots \wedge E(x_{n-1},y)$$

It is well-known that φ_n is equivalent to a formula in $L_{\infty,\omega}^3$ (in fact in L^3). To see this, put $\varphi_1(x,y) \equiv E(x,y)$ and assume, by induction on n, that $\varphi_{n-1}(x,y)$ is equivalent to a formula in $L_{\infty,\omega}^3$. Then

$$\varphi_n(x,y) = \exists z[E(x,z) \wedge \exists x((z=x) \wedge \varphi_{n-1}(x,y))]$$

A closer look at $\varphi_n(x,y)$ reveals that every conjunction used in the definition of $\varphi_n(x,y)$ is j-restricted and hence we can conclude that $\varphi_n(x,y)$ is in $O^{2,3}$ (and hence in $N^{2,3}$ and $M^{2,3}$).

Finally, let $\mathcal{C}$ be the set of all finite τ-structures that, interpreted as graphs, are bipartite. It is well known that a graph is bipartite if and only if does not contain odd cycles. Therefore, the set $\mathcal{C}$ is defined by the the formula

$$\exists x \bigwedge_{n \geq 0} \varphi_{2n+1}(x,x)$$

As we shall see later (Theorem 1) the expressive power of this logics is the same regardless on whether or not we allow the use of equalities ($=$) in the formulas. By convention we shall assume that, unless otherwise explicitly stated, formulas do not contain equalities.

We finish this section by stating without proof a very simple fact about these logics that will be used intensively in our proofs.

Proposition 1. Let $\mathbf{A}$, $\mathbf{B}$ be τ-structures such that $\mathbf{A} \longrightarrow \mathbf{B}$. For every sentence φ in $L_{\infty,\omega}^\omega$ that does not contain universal quantification nor negation such that $\mathbf{A} \models \varphi$ we have that $\mathbf{B} \models \varphi$.

4. Quasi-Orderings and Pathwidth of Relational Structures

A *quasi-ordering* on a set S is a reflexive and transitive relation $\leq$ on S. Let $\langle S, \leq \rangle$ be a quasi-ordered set. Let S', S'' be subsets of S. We say that S' is a *filter* if it is closed under $\leq$ upward; that is, if $x \in S'$ and $x \leq y$, then $y \in S'$. The *filter generated* by S'' is

the set $F(S'') = \{y \in S : \exists x \in S'' \; x \leq y\}$. We say that S' is an *ideal* if it is closed under $\leq$ downward; that is, if $x \in S'$ and $y \leq x$, then $y \in S'$. The *ideal generated* by S'' is the set $I(S'') = \{y \in S : \exists x \in S'' \; y \leq x\}$. Observe that every subset S' of S and its complement $S \backslash S'$ satisfy the following relation: S' is an ideal iff $S \backslash S'$ is a filter.

Let I be an ideal of $\langle S, \leq \rangle$. We say that a set $O \subseteq S$ forms an *obstruction set* for I if

$$x \in I \text{ iff } \forall y \in O(y \nleq x)$$

That is, O is an obstruction set for I if I is the complement of $F(O)$

Let τ be a vocabulary. The set of τ-structures, $\mathrm{STR}[\tau]$, is quasi-ordered by the homomorphism relation. In consequence, a set C of τ-structures is an ideal of $\langle \mathrm{STR}[\tau], \longrightarrow \rangle$ if

$$\mathbf{B} \in C, \mathbf{A} \longrightarrow \mathbf{B} \implies \mathbf{A} \in C$$

Observe that for any relational structure $\mathbf{B}$, $\mathrm{CSP}(\mathbf{B}) = I(\mathbf{B})$.

Let us define a notion of pathwidth relative to relational structures, which is the natural generalization of the notion of pathwidth over graphs, introduced by Robertson and Seymour [33]. We follow the lines of previous generalizations of similar notions as treewidth. For reasons that will be made clear later it is desirable to parameterize the ordinary notion of pathwidth to capture a finer structure. For this purpose we will consider not only the maximum size of any set of the path-decomposition but also the maximum size of its pairwise intersection.

Definition 1. Let $\mathbf{A}$ be a τ-structure. A *path-decomposition* of $\mathbf{A}$ is a collection $S_1, \ldots, S_n$, $S_i \subseteq A$ such that:

(1) for every r-ary relation symbol R in τ and every $\langle a_1, \ldots, a_r \rangle \in R^{\mathbf{A}}$, there exists $1 \leq i \leq n$ such that $\{a_1, \ldots, a_r\} \subseteq S_i$.

(2) if $a \in S_i \cap S_j$, then $a \in S_l$ for all $i \leq l \leq j$.

The *width* of the path-decomposition is defined to be the pair $\langle \max\{|S_i \cap S_{i+1}| : 1 \leq i \leq n-1\}, \max\{|S_i| : 1 \leq i \leq n\} \rangle$. We say that a structure $\mathbf{A}$ has *pathwidth* at most (j, k) if it has a path decomposition of width (j, k).

The concept of pathwidth relative to relational structures introduced here is intended to be a natural generalization of the notion of pathwidth defined over graphs [33]. However there are some points in which our definition is not standard. First, in the ordinary notion of pathwidth over graphs, as defined in [33], the width of a path-decomposition is defined as $\max\{|S_i| : 1 \leq i \leq n\} - 1$. We are interested in a more fine-grained analysis that motivates the consideration, not only of the cardinality of the sets but also the cardinality of the intersection of two consecutive sets in the intersection. Furthermore, it is convenient for us not to subtract 1, as it is customary, as then this subtraction would had to be carried over all the paper.

In order to delineate even more the precise relationship between the notion of pathwidth as it is usually defined over graphs and the notion of pathwidth of relational structures introduced in this paper we remark the following equivalence.

Let $\mathbf{A}$ be a relational structure. The the following numbers are equal:

- the pathwidth of the Gaifman graph of $\mathbf{A}$ plus one.
- the minimum k such that $\mathbf{A}$ has pathwidth at most $(k-1, k)$

We say that a set of structures C has pathwidth at most (j, k) if every structure $\mathbf{A}$ in C has pathwidth at most (j, k).

5. Pebble-Relation Games

In this section we will introduce a game, called (j, k)-pebble-relation game, that captures expressibility in $M^{j,k}$.

Let S_1 and S_2 be two (not necessarily finite) sets. A *relation* T with domain S_1 and range S_2 is a collection of functions with domain S_1 and range S_2. Remark: some confusion can arise from the fact that generally (and in this paper) the name relation is used with another meaning; for example, an r-ary relation over B is a subset of B^r. Both concepts are perfectly consistent, since an r-ary relation over B is, indeed, a relation in our sense with domain $\{1, \ldots, r\}$ and range B.

Let f be a function with domain S_1 and range S_2, and let S_1' be a subset of its domain S_1. We will denote by $f_{|S_1'}$ the restriction of f to S_1. Similarly, let T be a relation with domain S_1 and range S_2, and let S_1' be a subset of its domain S_1. We will denote by $T_{|S_1'}$ the relation with domain S_1' and range S_2 that contains $f_{|S_1'}$ for every $f \in T$. For every relation T we denote by $\mathrm{dom}(T)$ the domain of T. We have two relations with domain $\emptyset$: the relation $\{\lambda\}$ and the relation $\emptyset$.

Let $0 \le j \le k$ be non-negative integers and let $\mathbf{A}$ and $\mathbf{B}$ be (not necessarily finite) τ-structures. The (j, k)-pebble-relation $((j, k)$-PR) game on $\mathbf{A}$ and $\mathbf{B}$ is played between two players, the *Spoiler* and the *Duplicator*. A configuration of the game consists of a relation T with domain $I = \{a_1, \ldots, a_{k'}\} \subseteq A$, $k' \le k$ and range B such that every function f in T is a homomorphism from $\mathbf{A}_{|I}$ to $\mathbf{B}$.

Initially $I = \emptyset$ and T contains the (unique) homomorphism from $\mathbf{A}_{|\emptyset}$ to $\mathbf{B}$, that is, λ. Each round of the game consists of a move from the Spoiler and a move from the Duplicator. Intuitively, the Spoiler has control on the domain I of T, which can be regarded as placing some pebbles on the elements of A that constitute I, whereas the Duplicator decides the content of T after the domain I has been set by the Spoiler. There are two types of rounds: *shrinking* rounds and *blowing* rounds.

Let T^n be the configuration after the n-th round. The spoiler decides whether the following round is a blowing or shrinking round.

- If the $(n + 1)$-th round is a shrinking round, the Spoiler sets I^{n+1} (the domain of T^{n+1}) to be a subset of the domain I^n of T^n. The Duplicator responds by projecting every function in T^n onto the subdomain defined by I^{n+1}, that is, $T^{n+1} = T^n_{|I^{n+1}}$.
- A blowing round only can be performed if $|I^n| \le j$. In this case the Spoiler sets I^{n+1} to be a superset of I^n with $|I^{n+1}| \le k$. The duplicator responds by providing a T^{n+1} with domain I^{n+1} such that $T^{n+1}_{|I^n} \subseteq T^n$. That is, T^{n+1} should contain some extensions of functions in T^n over the domain I^{n+1} (recall that any such extension must be a homomorphism from $\mathbf{A}_{|I^{n+1}}$ to $\mathbf{B}$).

The Spoiler wins the game if the response of the Duplicator sets T^{n+1} to $\emptyset$, i.e., the Duplicator could not extend successfully any of the functions. Otherwise, the game resumes. The Duplicator wins the game if he has an strategy that allows him to continue playing "forever", i.e., if the Spoiler can never win a round of the game.

Now, we will present an algebraic characterization of the (j, k)-PR game.

Definition 2. Let $0 \le j \le k$ be non-negative integers and let $\mathbf{A}$ and $\mathbf{B}$ be (not necessarily finite) τ-structures. We say that *the Duplicator has a winning strategy for the (j, k)-pebble-relation game* on $\mathbf{A}$ and $\mathbf{B}$ if there is a nonempty family $\mathcal{H}$ of relations such that:

(a) every relation T has range B and domain I for some $I \subseteq A$ with $|I| \le k$.

(b) for every relation T in $\mathcal{H}$ with domain I, $\emptyset \neq T$ and $T \subseteq \mathrm{hom}(\mathbf{A}_{|I}, \mathbf{B})$

(c) $\mathcal{H}$ is closed under restrictions: for every T in $\mathcal{H}$ with domain I and every $I' \subseteq I$, we have that $T_{|I'} \in \mathcal{H}$.

(d) $\mathcal{H}$ has the (j, k)-forth property: for every relation T in $\mathcal{H}$ with domain I with $|I| \leq j$ and every superset I' of I with $|I'| \leq k$, there exists some relation T' in $\mathcal{H}$ with domain I' such that $T'_{|I} \subseteq T$.

Furthermore, if $\mathcal{H}$ satisfies the following condition we say that $\mathcal{H}$ is a *strict* winning strategy for the (j, k)-pebble-relation game.

(d') $\mathcal{H}$ has the strict (j, k)-forth property: for every relation T in $\mathcal{H}$ with domain I with $|I| \leq j$ and every superset I' of I with $|I'| \leq k$, the relation with domain I' given by $\{h \in \mathrm{hom}(\mathbf{A}_{|I'}, \mathbf{B}) : h_{|I} \in T\}$ belongs to $\mathcal{H}$.

The intuition behind the definition of a winning strategy is that every relation in a winning strategy corresponds to a winning configuration for the Duplicator in the game.

We need an auxiliary definition that is used a number of times in the proofs.

Definition 3. Let $\mathbf{A}$ be a τ-structure, let $a_1, \ldots, a_k$ be (not necessarily different) elements of A, and let $v_1, \ldots, v_k$ be variables. We denote by $\Theta(\mathbf{A}, a_1, \ldots, a_k)(v_1, \ldots, v_k)$ the formula in $O^{k,k}$, *with equality*, with variables among $v_1, \ldots, v_k$ defined by

$$\bigwedge_{R \in \tau} \bigwedge_{\langle a_{l_1}, \ldots, a_{l_{\rho(R)}} \rangle \in R^{\mathbf{A}}} R(v_{l_1}, \ldots, v_{l_{\rho(R)}}) \wedge \bigwedge_{1 \leq i < j \leq k, a_i = a_j} (v_i = v_j)$$

Notice that if all the elements $a_1, \ldots, a_k$ are different then we do not need the equality. The following properties of $\Theta(\mathbf{A}, a_1, \ldots, a_k)(v_1, \ldots, v_k)$ will be very helpful.

Proposition 2. Let $\mathbf{A}$ be a τ-structure, let $a_1, \ldots, a_k$ be elements of A and let θ be the formula $\Theta(\mathbf{A}, a_1, \ldots, a_k)(v_1, \ldots, v_k)$. Then we have

- For every τ-structure $\mathbf{B}$ and every $b_1, \ldots, b_k \in B$, we have that

$$\mathbf{B}, b_1, \ldots, b_k \models \theta(v_1, \ldots, v_k) \text{ iff } \mathbf{A}_{|\{a_1, \ldots, a_k\}}, a_1, \ldots, a_k \longrightarrow \mathbf{B}, b_1, \ldots, b_k$$

- For every quantifier-free formula $\varphi(v_1, \ldots, v_k)$ in L^k which statisfies $\mathbf{A}, a_1, \ldots, a_k \models \varphi(v_1, \ldots, v_k)$ we have that $\theta(v_1, \ldots, v_k)$ implies $\varphi(v_1, \ldots, v_k)$, i.e., for every $\mathbf{B}$ and every $b_1, \ldots, b_k \in B$,

$$\mathbf{B}, b_1, \ldots, b_k \models (\theta \Rightarrow \varphi)(v_1, \ldots, v_k)$$

The proof of these two simple facts about $\Theta(\mathbf{A}, a_1, \ldots, a_k)(v_1, \ldots, v_k)$ follows the lines of the proof of a well-known result, due to Chandra and Merlin [2] which states that conjunctive query evaluation, conjunctive query containment, and deciding the existence of a homomorphism are essentially the same problem.

The following results show that pebble-relation games, expressiveness in the existential positive fragment of finite-variable infinitary logic with restricted conjunction, and obstruction sets of bounded pathwidth are equivalent mathematical embodiments of the same concept.

Theorem 1. Let $0 \leq j \leq k$ be non-negative integers and let $\mathbf{A}$ and $\mathbf{B}$ be (not necessarily finite) τ-structures. The following statements are equivalent:

(1) The Duplicator has a winning strategy $\mathcal{H}$ for the (j, k)-PR game on $\mathbf{A}$ and $\mathbf{B}$.

(2) For every sentence φ in $M^{j,k}$ such that $\mathbf{A} \models \varphi$ we have that $\mathbf{B} \models \varphi$.

(3) For every sentence φ in $N^{j,k}$ such that $\mathbf{A} \models \varphi$ we have that $\mathbf{B} \models \varphi$.
(4) For every sentence φ in $O^{j,k}$ such that $\mathbf{A} \models \varphi$ we have that $\mathbf{B} \models \varphi$.
(5) Every τ-structure $\mathbf{P}$ with pathwidth at most (j,k) that homomorphically maps to $\mathbf{A}$ also homomorphically maps to $\mathbf{B}$.

The equivalences hold even if we allow the use of equalities in the formulas.

Proof. **(1)→(2)** Let $\mathcal{H}$ be a winning strategy for the Duplicator. We shall show, by induction on the construction of $M^{j,k}$ formulas, that if $\varphi(v_1, \ldots, v_m)$ is a formula of $M^{j,k}$ *with equalities* where the free variables of φ are among $v_1, \ldots, v_m$, then the following property (*) holds:

(*) For any elements $a_1, \ldots, a_m$ in A such that $\mathbf{A}, a_1, \ldots, a_m \models \varphi(v_1, \ldots, v_m)$, there exists some $T \in \mathcal{H}$ with domain $I \subseteq \{a_1, \ldots, a_m\}$ such that for every $h : \{a_1, \ldots, a_m\} \to B$ such that $h_{|I} \in T$ we have

$$\mathbf{B}, h(a_1), \ldots, h(a_m) \models \varphi(v_1, \ldots, v_m).$$

First, it is easy to see that if $\varphi(v_1, \ldots, v_m)$ is a quantifier-free formula in $M^{j,k}$ such that $\mathbf{A}, a_1, \ldots, a_m \models \varphi(v_1, \ldots, v_m)$ and $h : \{a_1, \ldots, a_m\} \to B$ is any homomorphism from $\mathbf{A}_{|\{a_1, \ldots, a_m\}}$ to $\mathbf{B}$ then $\mathbf{B}, h(a_1), \ldots, h(a_m) \models \varphi(v_1, \ldots, v_m)$.

To see it, let define θ as $\Theta(\mathbf{A}, a_1, \ldots, a_m)(v_1, \ldots, v_m)$. By Proposition 2 we have $\mathbf{B}, h(a_1), \ldots, h(a_m) \models \theta(v_1, \ldots, v_m) \Rightarrow \varphi(v_1, \ldots, v_m)$. Consequently, We can infer that $\mathbf{B}, h(a_1), \ldots, h(a_m) \models \varphi(v_1, \ldots, v_m)$ from the fact that $\mathbf{B}, h(a_1), \ldots, h(a_m) \models \theta(v_1, \ldots, v_m)$.

Thus, any relation $T \in \mathcal{H}$ with domain $\{a_1, \ldots, a_m\}$ would satisfy condition (*). The inductive step for infinitary disjunction $\bigvee$ is straightforward using the induction hypothesis.

Assume that the formula $\varphi(v_1, \ldots, v_m)$ is of the form $\bigwedge \Psi$ where Ψ is a collection of formulas in $M^{j,k}$. We have to show that there exists some $T \in \mathcal{H}$ with domain $I \subseteq \{a_1, \ldots, a_m\}$ such that for every $h : \{a_1, \ldots, a_m\} \to B$ with $h_{|I} \in T$ we have $\mathbf{B}, h(a_1), \ldots, h(a_m) \models \varphi(v_1, \ldots, v_m)$. Assume that there exists some formula γ in Ψ which is quantified but not a sentence (such formula if exists must be unique). Since γ has at most j variables we have that $\gamma = \gamma(v_{i_1}, \ldots, v_{i_{j'}})$ where $j' \leq j$. Thus, $\mathbf{A}, a_{i_1}, \ldots, a_{i_{j'}} \models \gamma(v_{i_1}, \ldots, v_{i'_j})$, and by the induction hypothesis there exists some $T' \in \mathcal{H}$ with domain $I' \subseteq \{a_{i_1}, \ldots, a_{i_{j'}}\}$ such that for all $h : \{a_{i_1}, \ldots, a_{i_{j'}}\} \to B$ such that $h_{|I'} \in T'$, we have $\mathbf{B}, h(a_{i_1}), \ldots, h(a_{i_{j'}}) \models \gamma(v_{i_1}, \ldots, v_{i_{j'}})$. If such formula does not exists then set $I' = \emptyset$ and $T' = \{\lambda\}$.

By the (j,k)-forth property, there exists some relation T with domain $\{a_1, \ldots, a_m\}$ such that $T_{|I'} \subseteq T'$. We shall see that T satisfies the desired condition. Let $h : \{a_1, \ldots, a_m\} \to B$ such that $h \in T$, and let $\psi(v_1, \ldots, v_m)$ be any formula in Ψ. We have to study three cases:

- If $\psi(v_1, \ldots, v_m)$ is quantifier-free then since $h \in \mathrm{hom}(\mathbf{A}_{|\{a_1, \ldots, a_m\}}, \mathbf{B})$, we have that $\mathbf{B}, h(a_1), \ldots, h(a_m) \models \psi(v_1, \ldots, v_m)$.
- If $\psi(v_1, \ldots, v_m)$ is a sentence then by induction hypothesis we have $\mathbf{B} \models \psi$ and thus, in consequence, $\mathbf{B}, h(a_1), \ldots, h(a_m) \models \psi(v_1, \ldots, v_m)$.
- Otherwise, we have $\psi(v_1, \ldots, v_m) = \gamma(v_{i_1}, \ldots, v_{i_{j'}})$. Since $T_{|I'} \subseteq T'$, we have $\mathbf{B}, h(a_{i_1}), \ldots, h(a_{i_{j'}}) \models \gamma(v_{i_1}, \ldots, v_{i_{j'}})$.

With respect to the existential quantification we shall distinguish two cases, depending on whether or not the variable quantified is contained in $v_1, \ldots, v_m$.

First, assume that the formula $\varphi(v_1, \ldots, v_m)$ is of the form $(\exists v)\psi(v_1, \ldots, v_m, v)$ (i.e., $v \notin \{v_1, \ldots, v_m\}$). Thus, there exists some $a \in A$ such that $\mathbf{A}, a_1, \ldots, a_m, a \models \psi(v_1, \ldots, v_m, v)$. Consequently, by the induction hypothesis, there exists some $T' \in \mathcal{H}$ with domain $I' \subseteq$

$\{a_1, \ldots, a_m, a\}$ such that for every $h : \{a_1, \ldots, a_m, a\} \to B$ such that $h_{|I'} \in T'$ we have $\mathbf{B}, h(a_1), \ldots, h(a_m), h(a) \models \psi(v_1, \ldots, v_m, v)$. We consider two cases:

- If $a \in \{a_1, \ldots, a_m\}$ then we set $I = I'$ and $T = T'$. Let $h : \{a_1, \ldots, a_m\} \to B$ such that $h_{|I} \in T$. In consequence, we have $\mathbf{B}, h(a_1), \ldots, h(a_m), h(a) \models \psi(v_1, \ldots, v_m, v)$, and, in consequence, $\mathbf{B}, h(a_1), \ldots, h(a_m) \models \psi(v_1, \ldots, v_m)$.
- Otherwise, we set $I = I' \backslash \{a\}$ and $T = T'_{|I}$ (notice that if $a \notin I'$ then $I = I'$ and $T = T'$). Let $h : \{a_1, \ldots, a_m\} \to B$ be a mapping such that $h_{|I} \in T$ and let $h' : \{a_1, \ldots, a_m, a\}$ be an extension of $h_{|I}$ such that $h'_{|I'} \in T'$ (such extension always exists since $T = T'_{|I}$). Thus $\mathbf{B}, h'(a_1), \ldots, h'(a_m), h'(a) \models \psi(v_1, \ldots, v_m)$ and consequently $\mathbf{B}, h(a_1), \ldots, h(a_m) \models \psi(v_1, \ldots, v_m)$.

Secondly, assume that $\varphi(v_1, \ldots, v_m)$ is of the form $(\exists v)\psi(v_1, \ldots, v_m)$. We can assume without loss of generality that $\{v_1, \ldots, v_{m'}, v\} = \{v_1, \ldots, v_m\}$ for $m' = m - 1$.

We have that $\mathbf{A}, a_1, \ldots, a_{m'} \models \varphi(v_1, \ldots, v_{m'})$. Thus, we are in the previous case and, in consequence, there exists some relation T with domain $I \subseteq \{a_1, \ldots, a_{m'}\}$ such that for every $h' : \{a_1, \ldots, a_{m'}\} \to B$ with $h'_{|I} \in T$ we have $\mathbf{B}, h'(a_1), \ldots, h'(a_{m'}) \models \varphi(v_1, \ldots, v_{m'})$. Finally, it is easy to see that T satisfies property (*): Let $h : \{a_1, \ldots, a_m\} \to B$ such that $h_{|I} \in T$ and let $h' = h_{|\{a_1, \ldots, a_{m'}\}}$. Since $h_{|I} = h'_{|I} \in T$ we have that $\mathbf{B}, h'(a_1), \ldots, h'(a_{m'}) \models \varphi(v_1, \ldots, v_{m'})$ and hence $\mathbf{B}, h(a_1), \ldots, h(a_m) \models \varphi(v_1, \ldots, v_m)$.

(2)→(3) and **(3)→(4)**. Straightforward since $O^{j,k} \subseteq N^{j,k} \subseteq M^{j,k}$.

(4)→(5) It will be convenient to assume that the path decomposition has a sort of canonical form. We will say that a path-decomposition $S_1, \ldots, S_n$ of a structure $\mathbf{P}$ is *canonical* if: (a) $S_n = \emptyset$ and (b) for every $1 \leq i \leq n - 1$ we have that $S_i \subseteq S_{i+1}$ or $S_i \supseteq S_{i+1}$. It is easy to verify that if a structure $\mathbf{P}$ has pathwidth at most (j, k), $0 \leq j \leq k$, then it has a canonical path-decomposition of width (j, k).

Let $\mathbf{P}$ be a structure and let $S_1, \ldots, S_n$ be a canonical path-decomposition of width (j, k) of $\mathbf{P}$ where $S_1 = \{p_1, \ldots, p_m\}$, $m \leq k$. We shall show, by induction on the size n of the decomposition, that there exists a formula $\varphi(v_1, \ldots, v_m)$ in $O^{j,k}$ with variables among $v_1, \ldots, v_k$ and whose free variables are exactly $v_1, \ldots, v_m$ such that for every structure $\mathbf{D}$ and every $d_1, \ldots, d_m \in D$ we have that

$$\mathbf{P}, p_1, \ldots, p_m \longrightarrow \mathbf{D}, d_1, \ldots, d_m,$$

if and only if

$$\mathbf{D}, d_1, \ldots, d_m \models \varphi(v_1, \ldots, v_m).$$

The result then follows from the following line of reasoning: Let $\mathbf{P}$ be any structure of pathwidth at most (j, k) and let, $S_1, \ldots, S_m$ be the canonical path-decomposition that certifies its pathwidth. Then $\emptyset, S_1, \ldots, S_m$ is also a path-decomposition of $\mathbf{P}$ and, consequently, there exists a sentence φ in $O^{j,k}$ such that for every structure $\mathbf{D}$, $\mathbf{P} \longrightarrow \mathbf{D}$ iff $\mathbf{D} \models \varphi$. Thus, if $\mathbf{P} \longrightarrow \mathbf{A}$ then $\mathbf{A} \models \varphi$ and by condition (4), $\mathbf{B} \models \varphi$, and $\mathbf{P} \longrightarrow \mathbf{B}$.

The base case in the induction ($n = 0$) is easily proved by setting φ to the formula identically true which can be obtained as the conjunction of an empty set of formulas.

For the induction step, let $S_1, \ldots, S_n, S_{n+1}$ be a path-decomposition of $\mathbf{P}$ of width (j, k). Thus $S_2, \ldots, S_n, S_{n+1}$ is a path-decomposition of $\mathbf{P}_{|S_2 \cup \cdots \cup S_{n+1}}$. We should distinguish two cases, depending on whether $S_1 \subseteq S_2$ or $S_2 \subseteq S_1$.

First assume that $S_1 \subseteq S_2$. Assume, without loss of generality that $S_1 = \{p_1, \ldots, p_m\}$ and $S_2 = \{p_1, \ldots, p_l\}$ where $0 \leq m \leq l \leq k$ and $m \leq j$. Notice that since $S_1 \subseteq S_2$, $S_2, \ldots, S_{n+1}$ is a path-decomposition of $\mathbf{P}$. Thus, by the inductive hypothesis, there exists

some formula $\varphi(v_1, \ldots, v_l)$ in $O^{j,k}$, such that for every τ-structure $\mathbf{D}$ and every $d_1, \ldots, d_l \in D$

$$\mathbf{P}, p_1, \ldots, p_l \longrightarrow \mathbf{D}, d_1, \ldots, d_l$$

if and only if

$$\mathbf{D}, d_1, \ldots, d_l \models \varphi(v_1, \ldots, v_l)$$

Consider the formula $\psi(v_1, \ldots, v_m)$ in $O^{j,k}$ with variables among $v_1, \ldots, v_k$ and whose free variables are exactly $v_1, \ldots, v_m$ defined by:

$$\psi(v_1, \ldots, v_m) = (\exists v_{m+1}, \ldots, v_l)\varphi(v_1, \ldots, v_l)$$

We shall show that ψ satisfies the desired property. For every $\mathbf{D}$ and every $d_1, \ldots, d_m \in D$ we have that

$$\mathbf{P}, p_1, \ldots, p_m \longrightarrow \mathbf{D}, d_1, \ldots, d_m,$$

if and only if

$$\exists d_{m+1}, \ldots, d_l \in D \text{ s.t. } \mathbf{P}, p_1, \ldots, p_l \longrightarrow \mathbf{D}, d_1, \ldots, d_l,$$

By the by the induction hypothesis is equivalent to

$$\exists d_{m+1}, \ldots, d_l \in D \text{ s.t. } \mathbf{D}, d_1, \ldots, d_l \models \varphi(v_1, \ldots, v_l),$$

which is equivalent to

$$\mathbf{D}, d_1, \ldots, d_m \models (\exists v_{m+1}, \ldots, v_l)\varphi(v_1, \ldots, v_l) = \psi(v_1, \ldots, v_m)$$

Assume now that $S_2 \subseteq S_1$. Let $S_1 = \{p_1, \ldots, p_m\}$ and $S_2 = \{p_1, \ldots, p_l\}$, where $0 \leq l \leq m \leq k$ and $l \leq j$. By the inductive hypothesis there exists some formula $\varphi(v_1, \ldots, v_l)$ in $O^{j,k}$, with variables among $v_1, \ldots, v_k$ and whose free variables are exactly $v_1, \ldots, v_l$ such that for every τ-structure $\mathbf{D}$ and every $d_1, \ldots, d_l \in D$

$$\mathbf{P}_{|S_2 \cup \cdots \cup S_{n+1}}, p_1, \ldots, p_l \longrightarrow \mathbf{D}, d_1, \ldots, d_l$$

if and only if

$$\mathbf{D}, d_1, \ldots, d_l \models \varphi(v_1, \ldots, v_l).$$

Let $\theta(v_1, \ldots, v_m) \in O^{j,k}$ defined by $\theta = \Theta(\mathbf{P}, p_1, \ldots, p_m)(v_1, \ldots, v_m)$. Notice that since $p_1, \ldots, p_m$ are different, θ does not contain equality. Finally consider the formula $\psi(v_1, \ldots, v_m)$ in $O^{j,k}$ defined by:

$$\psi(v_1, \ldots, v_m) = \theta(v_1, \ldots, v_m) \wedge \varphi(v_1, \ldots, v_l)$$

We shall prove that ψ satisfies the desired property. Let $\mathbf{D}$ be a τ-structure and let $d_1, \ldots, d_m$ be elements in D. Let h be any homomorphism from $\mathbf{P}$ to $\mathbf{D}$ such that $h(p_i) = d_i$ for $1 \leq i \leq m$.

Thus $h_{|S_1}$ is a homomorphism from $\mathbf{P}_{|S_1}$ to $\mathbf{D}$ and, by Proposition 2, $\mathbf{D}, d_1, \ldots, d_m \models \theta(v_1, \ldots, v_m)$. Also, we have that $h_{|S_2 \cup \cdots \cup S_{n+1}}$ is a homomorphism from $\mathbf{P}_{|S_2 \cup \cdots \cup S_{n+1}}$ to $\mathbf{D}$ and by the induction hypothesis, $\mathbf{D}, d_1, \ldots, d_l \models \varphi(v_1, \ldots, v_l)$. Putting all together we have that

$$\mathbf{D}, d_1, \ldots, d_m \models \theta(v_1, \ldots, v_m) \wedge \varphi(v_1, \ldots, v_l) = \psi(v_1, \ldots, v_m)$$

Conversely, suppose that $\mathbf{D}, d_1, \ldots, d_m \models \psi(v_1, \ldots, v_m)$. In consequence we have that $\mathbf{D}, d_1, \ldots, d_l \models \varphi(v_1, \ldots, v_l)$ and, by induction hypothesis, there exists some homomorphism h' from $\mathbf{P}_{|S_2 \cup \cdots \cup S_{n+1}}$ to $\mathbf{D}$ such that $h'(p_i) = d_i$, $1 \leq i \leq l$. Let $h : P \to D$ be the mapping defined by

$$h(p) = \begin{cases} d_i & \text{if } p = p_i \\ h'(p) & \text{otherwise} \end{cases}$$

We shall show that h defines a homomorphism from $\mathbf{P}$ to $\mathbf{D}$. First notice that since $\mathbf{D}, d_1, \ldots, d_m \models \theta(v_1, \ldots, v_m)$, $h_{|S_1}$ defines a homomorphism from $\mathbf{P}_{|S_1}$ to $\mathbf{D}$. Finally, notice that $h_{|S_2 \cup \cdots \cup S_{n+1}} = h'$ defines a homomorphism from $\mathbf{P}_{|S_2 \cup \cdots \cup S_{n+1}}$ to $\mathbf{D}$. Then we are at home: Let R be any relation symbol in τ of arity, say r, and let $\langle a_1, \ldots, a_r \rangle$ any tuple in $R^{\mathbf{P}}$. Since $S_1, \ldots, S_{n+1}$ is a path-decomposition of $\mathbf{P}$ we have that $\{a_1, \ldots, a_r\} \in S_j$ for some $j \in \{1, \ldots, n+1\}$. If $j = 1$ then we have that $\langle h(a_1), \ldots, h(a_r) \rangle$ in $R^{\mathbf{B}}$, as $h_{|S_1}$ defines a homomorphism from $\mathbf{P}_{|S_1}$ to $\mathbf{D}$. If, otherwise, $j > 1$, then $\langle h(a_1), \ldots, h(a_r) \rangle$ is in $R^{\mathbf{D}}$, as $h_{|S_2 \cup \cdots \cup S_{n+1}}$ defines a homomorphism from $\mathbf{P}_{|S_2 \cup \cdots \cup S_{n+1}}$

(5)→(1) We shall produce a winning strategy $\mathcal{H}$ for the Duplicator. For every structure $\mathbf{P}$, for every path-decomposition $S_1, \ldots, S_n$ of width (j, k) of $\mathbf{P}$, and for every mapping $h : P \to A$ such that:

(i) h is a homomorphism from $\mathbf{P}$ to $\mathbf{A}$ and
(ii) $h_{|S_1}$ is one-to-one and furthermore $(h_{|S_1})^{-1}$ is a homomorphism from $\mathbf{A}_{|h(S_1)}$ to $\mathbf{P}_{S_1}$.

The set $\mathcal{H}$ contains the relation T with domain $h(S_1)$ defined by

$$T = \{g_{|S_1} \circ (h_{|S_1})^{-1} : \mathbf{P} \xrightarrow{g} \mathbf{B}\}$$

First, notice that if $\mathbf{P}$ is the structure with universe $P = \emptyset$, and h and g are λ, then we have that $\{\lambda\} \in \mathcal{H}$, and hence, $\mathcal{H}$ is non-empty. We show now that $\mathcal{H}$ has the required properties:

- Clearly, every relation T in $\mathcal{H}$ has domain $I \subseteq A$ with $|I| \leq k$ and range B.
- We have to show that for every T with domain I and every $f \in T$, f is a homomorphism from $\mathbf{A}_{|I}$ to $\mathbf{B}$. First, assume that $f = g_{|S_1} \circ h_{|h(S_1)}^{-1}$ where g, h and S_1 are as defined above. Since $h_{|S_1}^{-1}$ is a homomorphism from $\mathbf{A}_{|h(S_1)}$ to $\mathbf{P}_{|S_1}$ and $g_{|S_1}$ is a homomorphism from $\mathbf{P}_{|S_1}$ to $\mathbf{B}$, then its composition f must be a homomorphism from $\mathbf{A}_{|h(S_1)}$ to $\mathbf{B}$. Furthermore, T is non empty, since $\mathbf{P} \longrightarrow \mathbf{A}$ implies that $\mathbf{P} \longrightarrow \mathbf{B}$.
- We have to show that $\mathcal{H}$ is closed under projection. Let T be any relation in $\mathcal{H}$ obtained from $\mathbf{P}$, $S_1, \ldots, S_n$ and h as defined above. In consequence T has domain $h(S_1)$. Let $I \subseteq h(S_1)$ and let $S_1' = h^{-1}(I)$. It is not difficult to see that $S_1', S_1, \ldots, S_n$ defines a path decomposition of $\mathbf{P}$, that $h_{|S_1'}$ is one-to-one, and that $(h_{|S_1'})^{-1}$ defines a homomorphism from $\mathbf{A}_{|I}$ to $\mathbf{P}_{|S_1'}$. Thus the relation $T' = \{g_{|S_1'} \circ (h_{|S_1'})^{-1} : \mathbf{P}_{|S_1'} \xrightarrow{g} \mathbf{B}\}$ with domain I belongs also to $\mathcal{H}$. We shall show that $T_{|I} = T'$. For every homomorphism f' from $\mathbf{A}_{|I}$ to $\mathbf{B}$, $f' \in T'$ iff there exists some homomorphism g from $\mathbf{P}$ to $\mathbf{B}$ such that $f' = g_{|S_1'} \circ h_{|S_1'}$. In consequence, $f' \in T'$ iff $f = g_{|S_1} \circ h_{S_1} \in T$. Finally, the result follows from the fact that $f_{|S_1'} = f'$.
- We have to show that $\mathcal{H}$ has the (j, k)-forth property. Let T be any relation in $\mathcal{H}$ with domain I obtained from $\mathbf{P}$, $S_1, \ldots, S_n$ and h as defined above and let I' be any superset of I with $|I'| \leq k$. Let $I' \backslash I = \{a_1', \ldots, a_l'\}$ and let $p_1', \ldots, p_l'$ elements not in P. Let $P' = P \cup \{p_1', \ldots, p_l'\}$, let $S_1' = S_1 \cup \{p_1', \ldots, p_l'\}$ and let $h' : P' \to A$ be the extension of h that maps p_i' to a_i' $(1 \leq i \leq l)$. That is

$$h'(p) = \begin{cases} a_i' & \text{if } p = p_i', 1 \leq i \leq l \\ h(p) & \text{otherwise} \end{cases}$$

We define the structure $\mathbf{P}'$ as the structure with universe P' such that for every R in τ,

$$R^{\mathbf{P}'} = R^{\mathbf{P}} \cup \{\langle p_1, \ldots, p_m \rangle : \langle h'(p_1), \ldots, h'(p_m) \rangle \in R^{\mathbf{A}}, \{p_1, \ldots, p_m\} \subseteq S'_1\}$$

Notice that by construction $S'_1, S_1, \ldots, S_n$ is a path-decomposition of width (j, k) of $\mathbf{P}'$ and that h' is a homomorphism from $\mathbf{P}'$ to $\mathbf{A}$ such that $h'_{|S'_1}$ is one-to-one and that $(h'_{|S'_1})^{-1}$ is a homomorphism from $\mathbf{A}_{|I'}$ to $\mathbf{P}_{|S'_1}$. Let T' be the relation obtained from $\mathbf{P}', S'_1, S_1, \ldots, S_n$ and h'. We shall see that $T'_{|I} \subseteq T$: Let f be any function in T'. Thus, $f = g_{S'_1} \circ (h'_{|S'_1})^{-1}$ for some homomorphism g from $\mathbf{P}'$ to $\mathbf{B}$. Thus, $g_{|S_1 \cup \cdots \cup S_n}$ defines a homomorphism from $\mathbf{P}$ to $\mathbf{B}$ and, in consequence, $f_{|I} = g_{|S_1} \circ (h_{|S_1})^{-1}$ belongs to T.

$\square$

The following theorem provides us with several alternative ways to characterize $M^{j,k}$ definability.

Theorem 2. Let $0 \leq j \leq k$ be non-negative integers and let $\mathcal{C}$ be a class of τ-structures. The following statements are equivalent:

(1) The class $\mathcal{C}$ is $M^{j,k}$-definable, i.e., there is a sentence φ of $M^{j,k}$ such that for every τ-structure we have that $\mathbf{A} \in \mathcal{C}$ iff $\mathbf{A} \models \varphi$.
(2) If $\mathbf{A}$ and $\mathbf{B}$ are τ-structures such that $\mathbf{A} \in \mathcal{C}$ and the Duplicator has a winning strategy for the (j, k)-PR game on $\mathbf{A}$ and $\mathbf{B}$, then $\mathbf{B} \in \mathcal{C}$.

Furthermore, if $\neg\mathcal{C}$ is a finitely generated ideal we also have that (1) and (2) are equivalent to the following:

(3) The class $\mathcal{C}$ is $N^{j,k}$-definable.
(4) The class $\neg\mathcal{C}$ has an obstruction set of pathwidth at most (j, k)

Proof. First, we will need the following definition.

Let $\mathcal{S}$ be a finite collection of τ-structures. We say that $\mathcal{S}$ is disjoint if for every different $\mathbf{A}, \mathbf{B} \in \mathcal{S}$, $A \cap B = \emptyset$. If $\mathcal{S}$ is disjoint we define $\oplus\mathcal{S}$ as the τ-structure whose universe is the union of the universes of all the structures in $\mathcal{S}$, and such that for every relation symbol $R \in \tau$, $R^{\oplus\mathcal{S}} = \bigcup_{\mathbf{A} \in \mathcal{S}} R^{\mathbf{A}}$. Observe that, as the universes of the structures in $\mathcal{S}$ are disjoint, if $\mathcal{S}$ has pathwidth at most (j, k) for some j, k then $\oplus\mathcal{S}$ has also pathwidth at most (j, k).

If $\mathcal{S} = \{\mathbf{A}_1, \ldots, \mathbf{A}_m\}$ we also denote $\oplus\mathcal{S}$ by $\mathbf{A}_1 \oplus \cdots \oplus \mathbf{A}_m$.

Let $\mathbf{B}$ be a τ-structure and let $\mathcal{S}$ be a set of τ-structures. It is easy to verify that $\oplus\mathcal{S} \longrightarrow \mathbf{B}$ iff $\mathbf{A} \longrightarrow \mathbf{B}$ for every $\mathbf{A} \in \mathcal{S}$.

Now we are in a position to prove Theorem 2

(1)→(2) Straightforward from Theorem 1. Assume that $\mathbf{A} \models \varphi$. Since the Duplicator has a winning strategy for the (j, k)-PR game then $\mathbf{B} \models \varphi$ and thus $\mathbf{B}$ belongs to $\mathcal{C}$.

(2)→(1) Clearly $\mathcal{C}$ is a filter and let $\neg\mathcal{C}$ be $I(\mathcal{S})$ for some set $\mathcal{S}$ of τ-structures. For every $\mathbf{B}$ in $\mathcal{S}$, let $\Psi_{\mathbf{B}}$ be the collection of sentences in $N^{j,k}$ falsified by $\mathbf{B}$ and let $\varphi_{\mathbf{B}} = \bigvee \Psi_{\mathbf{B}}$, which is a sentence of $N^{j,k}$. Let Ψ be the set containing $\varphi_{\mathbf{B}}$ for every $\mathbf{B}$ in $\mathcal{S}$, and let φ be the sentence of $M^{j,k}$ defined by $\varphi = \bigwedge \Psi$. Notice that if $\mathcal{S}$ is finite then φ is in $N^{j,k}$. We shall see that φ defines $\mathcal{C}$. Let $\mathbf{A}$ be any structure not in $\mathcal{C}$. Thus, there exists some $\mathbf{B}$ in $\mathcal{S}$ such that $\mathbf{A}$ homomorphically maps to $\mathbf{B}$. Recall that if $\mathbf{A}$ is homomorphic to $\mathbf{B}$, by Proposition 1, $\mathbf{B}$ satisfies all sentences in M^ω satisfied by $\mathbf{A}$. In consequence, $\mathbf{A} \not\models \varphi_{\mathbf{B}}$ and thus $\mathbf{A} \not\models \varphi$. Conversely, let $\mathbf{A}$ be any structure such that $\mathbf{A} \not\models \varphi$, thus $\mathbf{A} \not\models \varphi_{\mathbf{B}}$ for

some $\mathbf{B}$ in $\mathcal{S}$. Thus, every formula ψ in $N^{j,k}$ satisfied by $\mathbf{A}$ is also satisfied by $\mathbf{B}$, and, in consequence, the Duplicator has a winning strategy for the (j,k)-pebble-relation game on $\mathbf{A}$ and $\mathbf{B}$. Since $\mathbf{B} \notin \mathcal{C}$ we conclude that $\mathbf{A} \notin C$.

(3)→(1) Trivial as $N^{j,k} \subseteq M^{j,k}$.

(4)→(1) Straightforward from Theorem 1. Let $\mathbf{A} \in \mathcal{C}$ and $\mathbf{B} \notin \mathcal{C}$ be τ-structures. Since $\mathbf{A} \in \mathcal{C}$ there exists a structure $\mathbf{P}$ in the obstruction set (and therefore of pathwidth at most (j,k)) that homomorphically maps to $\mathbf{A}$. Since $\mathbf{B} \notin \mathcal{C}$, $\mathbf{P}$ does not homomorphically map to $\mathbf{B}$. In consequence, there does not exist a winning strategy for the Duplicator for the (j,k)-pebble-relation game on $\mathbf{A}$ and $\mathbf{B}$.

(2)→(4) Let $\mathbf{B}_1, \dots, \mathbf{B}_m$ be the finite set of structures that generates $\neg\mathcal{C}$. Consider the set $\mathcal{O}$ given by the collection of all structures of the form $\mathbf{P}_1 \oplus \cdots \oplus \mathbf{P}_m$ where $\mathbf{P}_i$ is a τ-structure of pathwidth at most (j,k) that does not map to $\mathbf{B}_i$, and $\mathbf{P}_i$, $1 \leq i \leq m$ have disjoint universes. Thus, every structure in $\mathcal{O}$ has pathwidth at most (j,k). We shall show that $\mathcal{O}$ is an obstruction set of $\neg\mathcal{C}$. First, observe that every structure $\mathbf{P}_1 \oplus \cdots \oplus \mathbf{P}_m$ in $\mathcal{O}$ belongs to $\mathcal{C}$, as for each $i \in \{1, \dots, m\}$, $\mathbf{P}_i \not\longrightarrow \mathbf{B}_i$. Consequently, if one of such structures $\mathbf{P}_1 \oplus \cdots \oplus \mathbf{P}_m$ is homomorphic to a given structure $\mathbf{A}$, then $\mathbf{A}$ must be in $\mathcal{C}$, as $\mathcal{C}$ is a filter. Conversely, let $\mathbf{A}$ be any structure in $\mathcal{C}$. Thus, for every $1 \leq i \leq m$, the Duplicator does not have a winning strategy for the (j,k)-pebble game on $\mathbf{A}$ and $\mathbf{B}_i$. Thus, for every $1 \leq i \leq m$ there exists some τ-structure $\mathbf{P}_i$ of pathwidth at most (j,k) such that $\mathbf{P}_i \longrightarrow \mathbf{A}$ and $\mathbf{P}_i \not\longrightarrow \mathbf{B}_i$ (we can assume without loss of generality that the universes of $\mathbf{P}_i$, $1 \leq i \leq m$ are disjoint). Thus $\mathbf{P}_1 \oplus \cdots \oplus \mathbf{P}_m$ belongs to $\mathcal{O}$ and furthermore it homomorphically maps to $\mathbf{A}$.

(2)→(3) The structure of this proof is similar to the previous case.

Let $\mathbf{B}_1, \dots, \mathbf{B}_m$ be the finite set of structures that generates $\neg\mathcal{C}$. Consider the set Ψ given by the collection of all structures of the form $\psi_1 \wedge \cdots \wedge \psi_m$ where ψ_i is a sentence in $O^{j,k}$ that is not satisfied by $\mathbf{B}_i$. Then the formula φ defined to be $\bigvee \Psi$ is in $N^{j,k}$. Let us show that φ defines $\mathcal{C}$. First observe that for every $\mathbf{B}_i$, $1 \leq i \leq m$ generating $\neg\mathcal{C}$, we have that $\mathbf{B}_i \not\models \varphi$. Consequently, as $\neg\mathcal{C}$ is an ideal, by Proposition 1, for any structure $\mathbf{A}$ not in $\mathcal{C}$, $\mathbf{A} \not\models \varphi$. Conversely, let $\mathbf{A}$ be any structure in $\mathcal{C}$. Thus, for every $1 \leq i \leq m$, the Duplicator does not have a winning strategy for the (j,k)-pebble game on $\mathbf{A}$ and $\mathbf{B}_i$. Thus, for every $1 \leq i \leq m$ there exists some sentence ψ_i in $O^{j,k}$ that is true on $\mathbf{A}$ but not true on $\mathbf{B}_i$. Thus $\psi_1 \wedge \cdots \wedge \psi_m$ is true on $\mathbf{A}$ and belongs to Ψ. Consequently, φ is true on $\mathbf{A}$. $\square$

6. Datalog Programs

Let τ be a vocabulary consisting of relational symbols. The class *SNP* [26, 32] is the set of all existential second-order sentences with a universal first-order part, i.e., sentences of the form $\exists S_1, \dots, S_l \forall v_1, \dots, v_m \varphi(v_1, \dots, v_m)$ where φ is a quantifier-free first-order formula over the vocabulary $\tau \cup \{S_1, \dots, S_l\}$ with variables among $v_1, \dots, v_m$. We will assume that φ is in CNF. We consider some restrictions that can be enforced on the class SNP. For *monotone SNP* [12], we require every occurrence of a relation symbol from τ to have negative polarity, i.e., a negation applied to it. For *j-adic SNP*, we require every second-order variable $S_i, 1 \leq i \leq l$ to have arity at most j. For *k-ary*, we require the number of variables universally quantified m to be at most k. For *Krom SNP* we require every clause of the quantifier-free first-order part φ to have at most two occurrences of a second-order variable.

Every Krom SNP formula φ over the vocabulary τ defines a class of τ-structures, namely, the set containing every τ-structure $\mathbf{A}$ such that $\mathbf{A} \models \varphi$. Furthermore, the problem

of deciding, given a τ-structure $\mathbf{A}$, whether $\mathbf{A} \models \varphi$ is solvable in NL [14]. Thus expressibility in Krom SNP is a sufficient condition for membership in NL.

For *restricted Krom SNP* we additionally require every clause of the quantifier-free first-order part φ to have at most one positive occurrence of a second-order variable and at most one negative occurrence of a second-order variable.

Theorem 3. Let $0 \leq j \leq k$ be non-negative integers and let $\mathbf{B}$ be a τ-structure. There exists a sentence φ in j-adic k-ary restricted Krom monotone SNP with equalities such that for every τ-structure $\mathbf{A}$, $\mathbf{A} \models \varphi$ iff the Duplicator has a winning strategy for the (j,k)-PR game on $\mathbf{A}$ and $\mathbf{B}$.

Proof. The proof of Theorem 3 requires some intermediate results. In a first step we define a different notion of winning strategy, called "supercomplete winning strategy" and we show that this new notion is equivalent to the notion of winning strategy introduced initially, that is, we prove that for every τ-structures $\mathbf{A}$ and $\mathbf{B}$, and for every $0 \leq j \leq k$ the Duplicator has a winning strategy for the (j,k)-pebble game if and only if it has a supercomplete winning strategy. In order to prove this we define several different but equivalent notions of winning strategy that will act a links in a chain of inferences.

The new notions of winning strategy are the following:

Definition 4. Let $0 \leq j \leq k$ be non-negative integers and let $\mathbf{A}$ and $\mathbf{B}$ be τ-structures. We say that the Duplicator has a *complete winning strategy* for the (j,k)-pebble-relation game on $\mathbf{A}$ and $\mathbf{B}$ if there is a family $\mathcal{H}$ of relations such that:

(a) every relation T has range B and domain I for some $I \subseteq A$ with $|I| \leq j$.
(b) $\mathcal{H}$ contains $\{\lambda\}$ and does not contain $\emptyset$.
(c) for every $I \subseteq A$ with $|I| \leq k$, every relation T in $\mathcal{H}$ with domain $I' \subseteq I$ and every $I'' \subseteq I$ with $|I''| \leq j$, the relation with domain I'' given by

$$\{h_{|I''} : h \in \mathrm{hom}(\mathbf{A}_{|I}, \mathbf{B}), h_{|I'} \in T\}$$

belongs to $\mathcal{H}$

Furthermore, if $\mathcal{H}$ satisfies the following condition we say that $\mathcal{H}$ is a *supercomplete winning strategy* for the (j,k)-pebble game

(d) For every T in $\mathcal{H}$ and every $T \subseteq T'$ we have that $T' \in \mathcal{H}$

The following claim states that all the different characterizations of winning strategy introduced so far are equivalent.

Claim 1. Let $0 \leq j \leq k$ be non-negative integers and let $\mathbf{A}$ and $\mathbf{B}$ be τ-structures. The following statements are equivalent:

(1) The Duplicator has a winning strategy for the (j,k)-PR game on $\mathbf{A}$ and $\mathbf{B}$.
(2) The Duplicator has a strict winning strategy for the (j,k)-PR game on $\mathbf{A}$ and $\mathbf{B}$.
(3) The Duplicator has a complete winning strategy for the (j,k)-PR game on $\mathbf{A}$ and $\mathbf{B}$.
(4) The Duplicator has a supercomplete winning strategy for the (j,k)-PR game on $\mathbf{A}$ and $\mathbf{B}$.

Proof. **(1)→(2)** Let $\mathcal{H}$ be a winning strategy for the Duplicator for the (j,k)-PR game on $\mathbf{A}$ and $\mathbf{B}$. We define $\mathcal{H}^*$ to be the set

$$\{T^* : \mathrm{dom}(T^*) \subseteq A, |\mathrm{dom}(T^*)| \leq k, T^* \subseteq \mathrm{hom}(\mathbf{A}_{\mathrm{dom}(T)}, \mathbf{B}), \exists T \in \mathcal{H} \text{ s.t. } T \subseteq T^*\}$$

We will show that $\mathcal{H}^*$ is a strict winning strategy:

- It is obvious that $\mathcal{H}^*$ satisfies conditions (a) and (b).
- $\mathcal{H}^*$ is closed under restrictions: Let T^* a relation in $\mathcal{H}^*$ with domain I and let $I' \subseteq I$. Thus, there exists some T in $\mathcal{H}$ with domain I such that $T \subseteq T^*$. Since $\mathcal{H}$ is closed under restrictions $T_{|I'} \in \mathcal{H}$. Since $T_{|I'} \subseteq T^*_{|I'}$ we have that $T^*_{|I'} \in \mathcal{H}^*$.
- $\mathcal{H}^*$ has the strict (j,k)-forth property: Let T^* be a relation in $\mathcal{H}^*$ with domain I, with $|I| \leq j$ and let I' be superset of I with $|I'| \leq k$. Thus, there exists some relation T in $\mathcal{H}$ with domain I such that $T \subseteq T^*$. Since $\mathcal{H}$ has the (j,k)-forth property there exists some relation T' in $\mathcal{H}$ with domain I' such that $T'_{|I} \subseteq T$. In consequence $T' \subseteq \{h \in \hom(\mathbf{A}_{|I'}, \mathbf{B}) : h_{|I} \in T^*\}$. Thus, the latter is in $\mathcal{H}^*$

(2)→(3) Let $\mathcal{H}$ be a strict winning strategy for the Duplicator and let $\mathcal{H}^*$ be the collection of relations in $\mathcal{H}$ with domain of size at most j. We shall show that $\mathcal{H}^*$ is a complete winning strategy. It is straightforward to show that $\mathcal{H}^*$ satisfies conditions (a) and (b) of the definition of a complete winning strategy. For condition (c), let $I \subseteq A$ with $|I| \leq k$, let T^* be a relation in $\mathcal{H}^*$ (and hence in $\mathcal{H}$) with domain $I' \subseteq I$ and let $I'' \subseteq I$ with $|I''| \leq j$. By the strict (j,k)-forth property we have that the relation of arity I given by $\{h \in \hom(\mathbf{A}_{|I}, \mathbf{B}), h_{|I'} \in T^*\}$ belongs also to $\mathcal{H}$ and so its projection to I'' which is given by $\{h_{|I''} : h \in \hom(\mathbf{A}_{|I}, \mathbf{B}), h_{|I'} \in T^*\}$. Since $|I''| \leq j$, then it also belongs to $\mathcal{H}^*$.

(3)→(4) Let $\mathcal{H}$ be a complete winning strategy and let $\mathcal{H}^*$ be the set containing every relation T^* such that there exists a relation $T \in \mathcal{H}$ with $T \subseteq T^*$. Clearly $\mathcal{H}^*$ satisfies conditions (a), (b), and (d) of a supercomplete winning strategy. For condition (c), let $I \subseteq A$ with $|I| \leq k$, let T^* be a relation in $\mathcal{H}^*$ with domain $I' \subseteq I$ and let $I'' \subseteq I$ with $|I''| \leq j$. There exists some $T \in \mathcal{H}$ such that $T \subseteq T^*$ and, in consequence $\{h_{|I''} : h \in \hom(\mathbf{A}_{|I}, \mathbf{B}), h_{|I'} \in T\}$ is a subset of $\{h_{|I''} : h \in \hom(\mathbf{A}_{|I}, \mathbf{B}), h_{|I'} \in T^*\}$ and hence the latter is in $\mathcal{H}^*$.

(4)→(1) Let $\mathcal{H}$ be a supercomplete winning strategy and let $\mathcal{H}^*$ be a set containing for every $I \subseteq A$ with $|I| \leq k$, and for every T in $\mathcal{H}$ with domain $I' \subseteq I$, and for every $I'' \subseteq I$ the relation $\{h_{|I''} : h \in \hom(\mathbf{A}_{|I}, \mathbf{B}), h_{|I'} \in T\}$. It is straightforward to show that $\mathcal{H}^*$ satisfies the conditions (a), (b), and (c) of a winning strategy. For condition (d), let T'' be any relation in $\mathcal{H}^*$ with domain I'' with $|I''| \leq j$. Thus, $T'' = \{h_{|I''} : h \in \hom(\mathbf{A}_{|I}, \mathbf{B}), h_{|I'} \in T\}$ for some T in $\mathcal{H}$ with domain I' and some $I' \subseteq I$ with $|I| \leq k$. Then, T'' is also in $\mathcal{H}$. Now, let I''' be any superset of I'' with $|I'''| \leq k$. Thus, $T''' = \{h_{|I'''} : h \in \hom(\mathbf{A}_{|I'''}, \mathbf{B}), h_{|I''} \in T''\}$ is also in $\mathcal{H}$ and $T'''_{|I''} \subseteq T''$.   □

In a second step we reformulate having a supercomplete winning strategy as the existence of a structure satisfying certain properties. Let r be a non-negative integer and let B a set. Recall that an r-ary relation over B is a relation with domain $\{1,\dots,r\}$ and range B.

Let $\{U_n : 1 \leq n \leq m\}$ be the collection of all j-ary relations over B and let $\{R_{U_n} : 1 \leq n \leq m\}$ be a collection of j-ary relation symbols, one for each relation $U_n : 1 \leq n \leq m$. We shall show that the Duplicator has a superstrict winning strategy for the (j,k)-pebble-relation game on $\mathbf{A}$ and $\mathbf{B}$ iff there exists some structure $\mathbf{A}'$ with domain A over the vocabulary $\{R_{U_n} : 1 \leq n \leq m\}$ such that:

(a) $(R_\emptyset)^{\mathbf{A}'} = \emptyset$.

(b) For every (not necessarily different) $a_1, \ldots, a_k \in A$, and for every $1 \leq l_1, \ldots, l_j \leq k$, $\langle a_{l_1}, \ldots, a_{l_j} \rangle \in (R_U)^{\mathbf{A}'}$ where

$$U = \{\langle h(a_{l_1}), \ldots, h(a_{l_j}) \rangle : h \in \hom(\mathbf{A}_{\{a_1, \ldots, a_k\}}, \mathbf{B})\}$$

(c) For every (not necessarily different) $a_1, \ldots, a_k$ elements in A, for every relation symbol $R_{U_n}, 1 \leq n \leq m$, for every $\langle a_{i_1}, \ldots, a_{i_j} \rangle \in (R_{U_n})^{\mathbf{A}'}$ and for every $1 \leq l_1, \ldots, l_j \leq k$, we have that $\langle a_{l_1}, \ldots, a_{l_j} \rangle \in (R_U)^{\mathbf{A}'}$ where

$$U = \{\langle h(a_{l_1}), \ldots, h(a_{l_j}) \rangle : h \in \hom(\mathbf{A}_{\{a_1, \ldots, a_k\}}, \mathbf{B}), \langle h(a_{i_1}), \ldots, h(a_{i_j}) \rangle \in U_n\}$$

(d) For every $1 \leq n, n' \leq k$ such that $U_n \subseteq U_{n'}$ we have $(R_{U_n})^{\mathbf{A}'} \subseteq (R_{U_{n'}})^{\mathbf{A}'}$

We shall say that $\mathbf{A}'$ is a *relational* winning strategy for the Duplicator for the (j, k)-pebble-relation game on $\mathbf{A}$ and $\mathbf{B}$.

Claim 2. Let $0 \leq j \leq k$ be non-negative integers and let $\mathbf{A}$ and $\mathbf{B}$ be τ-structures. The following statements are equivalent:

- The Duplicator has a supercomplete winning strategy for the (j, k)-pebble relation game on $\mathbf{A}$ and $\mathbf{B}$.
- The Duplicator has a relational winning strategy for the (j, k)-pebble relation game on $\mathbf{A}$ and $\mathbf{B}$.

Proof. First, we need the following definition, let $a_1, \ldots, a_l$ be a collection of (not necessarily different) elements of A and let T be a relation with domain $\{a_1, \ldots, a_l\}$. We define $U(T, a_1, \ldots, a_l)$ be the l-ary relation over B given by

$$\{\langle h(a_1), \ldots, h(a_l) \rangle : h \in T\}$$

Let $\mathcal{H}$ be a supercomplete winning strategy for the Duplicator for the (j, k)-pebble on $\mathbf{A}$ and $\mathbf{B}$ and let $\mathbf{A}'$ be the $\{R_{U_n} : 1 \leq n \leq m\}$-structure such that for every $1 \leq n \leq m$,

$$(R_{U_n})^{\mathbf{A}'} = \{ \quad \langle a_1, \ldots, a_j \rangle : a_1, \ldots, a_j \in A, \\ \exists T \in \mathcal{H}, \{a_1, \ldots, a_j\} = \mathrm{dom}(T), U(T, a_1, \ldots, a_j) \subseteq U_n\}$$

We shall show that $\mathbf{A}'$ satisfies the desired conditions:

(a) By definition every relation T in $\mathcal{H}$ is non-empty. Thus $(R_\emptyset)^{\mathbf{A}'} = \emptyset$.
(b) Let $a_1, \ldots, a_k$ be (not necessarily different) elements in A and let $1 \leq l_1, \ldots, l_j \leq k$. We shall show that $\langle a_{l_1}, \ldots, a_{l_j} \rangle \in (R_U)^{\mathbf{A}'}$ where

$$U = \{\langle h(a_{l_1}), \ldots, h(a_{l_j}) \rangle : h \in \hom(\mathbf{A}_{\{a_1, \ldots, a_k\}}, \mathbf{B})\}$$

Since $\mathcal{H}$ is an supercomplete winning strategy, then the relation

$$T = \{h_{|\{a_{l_1}, \ldots, a_{l_j}\}} : h \in \hom(\mathbf{A}_{|\{a_1, \ldots, a_k\}}, \mathbf{B}), h_{|\emptyset} \in \{\lambda\}\}$$

which is equivalent to

$$\{h_{|\{a_{l_1}, \ldots, a_{l_j}\}} : h \in \hom(\mathbf{A}_{|\{a_1, \ldots, a_k\}}, \mathbf{B})\}$$

belongs to $\mathcal{H}$. In consequence we have $\langle a_{l_1}, \ldots, a_{l_j} \rangle \in (R_{U(T, a_{l_1}, \ldots, a_{l_j})})^{\mathbf{A}'}$. Finally, we have $U(T, a_{l_1}, \ldots, a_{l_j}) = U$.

(c) Let $a_1, \ldots, a_k$ be (not necessarily different) elements in A, let R_{U_n} be a relation symbol, let $\langle a_{i_1}, \ldots, a_{i_j} \rangle \in (R_{U_n})^{\mathbf{A}'}$ and let $1 \leq l_1, \ldots, l_j \leq k$. We shall show that $\langle a_{l_1}, \ldots, a_{l_j} \rangle \in (R_U)^{\mathbf{A}'}$ where

$$U = \{\langle h(a_{l_1}), \ldots, h(a_{l_j})\rangle : h \in \mathrm{hom}(\mathbf{A}_{\{a_1,\ldots,a_k\}}, \mathbf{B}), \langle h(a_{i_1}), \ldots, h(a_{i_j})\rangle \in U_n\}$$

First, there exists some $T \in \mathcal{H}$ that has domain $\{a_{i_1}, \ldots, a_{i_j}\}$ and such that the inclusion $U(T, a_{i_1}, \ldots, a_{i_j}) \subseteq U_n$ holds. Since $\mathcal{H}$ is a supercomplete winning strategy we have that

$$\begin{aligned} T' &= \{h_{|\{a_{l_1},\ldots,a_{l_j}\}} : h \in \mathrm{hom}(\mathbf{A}_{|\{a_1,\ldots,a_k\}}, \mathbf{B}), h_{|\{a_{i_1},\ldots,a_{i_j}\}} \in T\} \\ &\subseteq \{h_{|\{a_{l_1},\ldots,a_{l_j}\}} : h \in \mathrm{hom}(\mathbf{A}_{|\{a_1,\ldots,a_k\}}, \mathbf{B}), \langle h(a_{i_1}), \ldots, h(a_{i_j})\rangle \in U_n\} \end{aligned}$$

belongs also to $\mathcal{H}$.

Thus, $\langle a_{l_1}, \ldots, a_{l_j} \rangle$ belongs to $(R_{U(T', a_{l_1}, \ldots, a_{l_j})})^{\mathbf{A}'}$ and it is not difficult to see that $U(T', a_{l_1}, \ldots, a_{l_j}) \subseteq U$.

(d) Straightforward from the definition.

For the converse, let $\mathbf{A}'$ be a $\{R_{U_n} : 1 \leq n \leq m\}$-structure satisfying (a), (b), (c), and (d). Let $\mathcal{H}$ be the collection of relations

$$\{\{\lambda\}\} \cup \{T : \exists a_1, \ldots, a_j \in A, \mathrm{dom}(T) = \{a_1, \ldots, a_j\}, \langle a_1, \ldots, a_j \rangle \in (R_{U(T,a_1,\ldots,a_j)})^{\mathbf{A}'}\}$$

We shall show that $\mathcal{H}$ is an supercomplete winning strategy for the Duplicator for the (j,k)-pebble-relation game on $\mathbf{A}$ and $\mathbf{B}$. It is immediate to show that $\mathcal{H}$ satisfies conditions (a), (b) and (d) of the definition of a superstrict winning strategy. For condition (c), let $I \subseteq A$ with $|I| \leq k$, let T be a relation in $\mathcal{H}$ with domain $I' \subseteq I$ and let $I'' \subseteq I$ with $|I''| \leq j$. We have to show that $T' = \{h_{|I''} : h \in \mathrm{hom}(\mathbf{A}_{|I}, \mathbf{B}), h_{|I'} \in T\}$ belongs to $\mathcal{H}$. We will do a case analysis.

First assume that $I'' = \emptyset$. In this case, since property (b) of the definition of supercomplete winning strategy is satisfied we have that $T' = \{\lambda\}$ which belongs to $\mathcal{H}$. Now assume that $I'' \neq \emptyset$, say $I = \{a_1, \ldots, a_k\}$ (here $a_1, \ldots, a_k$ are not necessarily different) and $I'' = \{a_{l_1}, \ldots, a_{l_j}\}$

• First consider the case $I' = \emptyset$. Thus $T = \{\lambda\}$ and $T' = \{h_{|I''} : h \in \mathrm{hom}(\mathbf{A}_{|I}, \mathbf{B})\}$. From condition (b) we have that $\langle a_{l_1}, \ldots, a_{l_j} \rangle \in (R_U)^{\mathbf{A}'}$ where

$$U = \{\langle h(a_{l_1}), \ldots, h(a_{l_j})\rangle : h \in \mathrm{hom}(\mathbf{A}_{\{a_1,\ldots,a_k\}}, \mathbf{B})\} = U(T', a_{l_1}, \ldots, a_{l_j}),$$

and thus $T' \in \mathcal{H}$.

• The case $I' \neq \emptyset$ is proven similarly. In this case we have that there exist some $a_{i_1}, \ldots, a_{i_j} \in I$ such that $I' = \{a_{i_1}, \ldots, a_{i_j}\}$ and $\langle a_{i_1}, \ldots, a_{i_j} \rangle \in (R_{U(T, a_{i_1}, \ldots, a_{i_j})})^{\mathbf{A}'}$. Thus, $T' = \{h_{|I''} : h \in \mathrm{hom}(\mathbf{A}_{|I}, \mathbf{B}), h_{|I'} \in T\}$ is identical to

$$\{h_{|\{a_{l_1},\ldots,a_{l_j}\}} : h \in \mathrm{hom}(\mathbf{A}_{|\{a_1,\ldots,a_k\}}, \mathbf{B}), \langle h(a_{i_1}), \ldots, h(a_{i_j})\rangle \in (R_{U(T,a_{i_1},\ldots,a_{i_j})})^{\mathbf{A}'}\}$$

From condition (c) of relational winning strategy we have that $\langle a_{i_1}, \ldots, a_{i_j} \rangle \in (R_U)^{\mathbf{A}'}$ where

$$\begin{aligned} U = \{\langle h(a_{l_1}), \ldots, h(a_{l_j})\rangle : \quad &h \in \mathrm{hom}(\mathbf{A}_{|\{a_1,\ldots,a_k\}}, \mathbf{B}), \\ &\langle h(a_{i_1}), \ldots, h(a_{i_j})\rangle \in (R_{U(T,a_{i_1},\ldots,a_{i_j})})^{\mathbf{A}'}\} \end{aligned}$$

which is equal to $U(T', a_{l_1}, \ldots, a_{l_j})$; henceforth, $T' \in \mathcal{H}$.

$\square$

After the equivalence of supercomplete winning strategies and relational winning strategies has been established, we shall construct a sentence φ over the vocabulary $\tau \cup \{=\}$ that tests whether such a structure $\mathbf{A}'$ (certifying the existence of a relational winning strategy) exists. This is our third (and final) component of the proof of Theorem 3.

Claim 3. For every $\mathbf{B}$ there exists a sentence φ over the vocabulary $\tau \cup \{=\}$ such that for every τ-structure $\mathbf{A}$, $\mathbf{A} \models \varphi$ iff the Duplicator has a relational winning strategy for the (j,k)-PR game on $\mathbf{A}$ and $\mathbf{B}$.

Proof. The sentence φ has a second-order predicate R_{U_i} of arity j for every j-ary relation $U_i, 1 \leq i \leq m$ over B. Thus

$$\varphi = \exists R_{U_1}, \ldots, R_{U_m} \forall v_1, \ldots, v_k \psi(v_1, \ldots, v_k),$$

where $\psi(v_1, \ldots, v_k)$ is a first-order formula over the vocabulary $\tau \cup \{=, R_{U_1}, \ldots, R_{U_m}\}$ that is quantifier-free and has variables among $v_1, \ldots, v_k$. Let us describe ψ. The formula ψ will be a first-order formula in conjunctive normal form. First we need some auxiliary definitions, let τ' be $\tau \cup \{=, R_{U_1}, \ldots, R_{U_m}\}$ and let $\mathbf{B}'$ be the τ'-structure with universe B such that for every $R \in \tau$, $R^{\mathbf{B}'} = R^{\mathbf{B}}$, $(=)^{\mathbf{B}'} = \{(b,b) : b \in B\}$, and such that for every $1 \leq i \leq m$, $(R_{U_i})^{\mathbf{B}'} = U_i$.

A disjunctive formula $\gamma(v_1, \ldots, v_k)$ is any first-order quantifier-free formula obtained as the disjunction of some (possibly negated) predicates in τ' applied to variables in $v_1, \ldots, v_k$. γ is monotone if every occurrence of a predicate in τ' is negated.

Let $\gamma(v_1, \ldots, v_k)$ be any disjunctive monotone formula over the vocabulary τ' with variables among $v_1, \ldots, v_k$ and let $1 \leq i_1, \ldots, i_j \leq k$, be a collection of indices. We define $U(\gamma, i_1, \ldots, i_j)$ as the j-ary relation over B defined by

$$\{\langle b_{i_1}, \ldots, b_{i_j} \rangle : \mathbf{B}', b_1, \ldots, b_k \not\models \gamma(v_1, \ldots, v_k)\}$$

We are now in a position to describe $\psi(v_1, \ldots, v_k)$. The formula ψ is of the form $\bigwedge \Psi$ with $\Psi = \Psi_1 \cup \Psi_2 \cup \Psi_3$ where:

- Ψ_1 contains the formula $\neg R_\emptyset(v_{l_1}, \ldots, v_{l_j})$ for every $1 \leq l_1, \ldots, l_j \leq k$.
- Ψ_2 contains for every disjunctive monotone formula γ over the vocabulary τ' with at most one occurrence of a second-order predicate and every collection of indices $i \leq i_i, \ldots, i_j \leq k$, the formula (clause) $\gamma(v_1, \ldots, v_k) \vee R_{U(\gamma, i_1, \ldots, i_j)}(v_{i_1}, \ldots, v_{i_j})$. Notice that such clause is monotone and restricted Krom, although it might contain equalities
- Ψ_3 contains for every $1 \leq n, n' \leq m$ such that $U_n \subseteq U'_n$, and for every $1 \leq l_1, \ldots, l_j \leq k$ the formula $\neg R_{U_n}(v_{l_1}, \ldots, v_{l_j}) \vee R_{U_{n'}}(v_{l_1}, \ldots, v_{l_j})$.

Informally, each one of the subsets Ψ_1, Ψ_2, Ψ_3 of Ψ encodes a condition of the definition of relational winning strategy. Indeed, it is not difficult to see that $\bigwedge \Psi_1$ is equivalent to condition (a) of relational winning strategy and that $\bigwedge \Psi_3$ formulates condition (d) of relational winning strategy. It is also possible to see, although this case is certainly more complicated, that $\bigwedge \Psi_2$ encodes exactly conditions (b) and (c) of relational winning strategy. The intuition here is that the collection of all formulas $\gamma(v_1, \ldots, v_k) \vee R_{U(\gamma, i_1, \ldots, i_j)}(v_{i_1}, \ldots, v_{i_j})$ in Ψ where γ does not contain a second-order predicate encodes (b) whereas the set of all such formulas with γ containing one occurrence of a second-order predicate encodes (c).

In the following we shall make all this more precise.

Let $\mathbf{A}'$ be a τ'-structure. We shall show that $\mathbf{A}' \models \forall v_1, \ldots, v_k \psi(v_1, \ldots, v_k)$ iff $\mathbf{A}'' = \mathbf{A}'[\{R_{U_n} : 1 \leq n \leq m\}]$ is a relational winning strategy for the Duplicator for the (j,k)-PR game on $\mathbf{A} = \mathbf{A}'[\tau]$ and $\mathbf{B}$. It is easy to observe that this implies our claim.

First, assume that $\mathbf{A}' \models \forall v_1, \ldots, v_k \psi(v_1, \ldots, v_k)$. Thus, for every $a_1, \ldots, a_k$ in A, $\mathbf{A}', a_1, \ldots, a_k \models \psi(v_1, \ldots, v_k) = \bigwedge \Psi$. Since for every $1 \leq l_1, \ldots, l_j \leq k$, $\neg R_\emptyset(v_{l_1}, \ldots, v_{l_j})$ is in Ψ we have that $\langle a_{l_1}, \ldots, a_{l_j} \rangle \notin (R_\emptyset)^{\mathbf{A}''}$. Thus $(R_\emptyset)^{\mathbf{A}''} = \emptyset$ and, in consequence, $\mathbf{A}''$ satisfies condition (a) on the definition of relational winning strategy.

We shall show that $\mathbf{A}''$ satisfies (b). Let $a_1, \ldots, a_k$ be (not necessarily different) elements of A. Let $\theta(v_1, \ldots, v_k) = \Theta(\mathbf{A}, a_1, \ldots, a_k)(v_1, \ldots, v_k)$.

Thus, we have $\mathbf{A}', a_1, \ldots, a_k \models \neg\theta(v_1, \ldots, v_k) \lor R_{U(\neg\theta, l_1, \ldots, l_j)}(v_{l_1}, \ldots, v_{l_j})$. Since by Proposition 2, $\mathbf{A}', a_1, \ldots, a_k \not\models \neg\theta(v_1, \ldots, v_k)$ we have that $\langle a_{l_1}, \ldots, a_{l_j} \rangle \in (R_{U(\neg\theta, l_1, \ldots, l_j)})^{\mathbf{A}''}$. Finally we have:

$$
\begin{aligned}
U(\neg\theta, l_1, \ldots, l_j) &= \{\langle b_{l_1}, \ldots, b_{l_j} \rangle : \mathbf{B}', b_1, \ldots, b_k \not\models \neg\theta(v_1, \ldots, v_k)\} \\
&= \{\langle b_{l_1}, \ldots, b_{l_j} \rangle : \mathbf{A}_{|\{a_1, \ldots, a_k\}}, a_1, \ldots, a_k \longrightarrow \mathbf{B}', b_1, \ldots, b_k\} \\
&= \{\langle h(a_{l_1}), \ldots, h(a_{l_j}) \rangle : h \in \mathrm{hom}(\mathbf{A}_{|\{a_1, \ldots, a_k\}}, \mathbf{B})\}
\end{aligned}
$$

We shall show that $\mathbf{A}''$ satisfies condition (c). Let $a_1, \ldots, a_k$ be (not necessarily different) elements of A, and let θ be defined as above. Let R_{U_n} be a relation symbol, let $\langle a_{i_1}, \ldots, a_{i_j} \rangle \in (R_{U_n})^{\mathbf{A}''}$ and let $1 \leq l_1, \ldots, l_j \leq k$. Let $\gamma(v_1, \ldots, v_k)$ be the disjunctive monotone formula defined by $\gamma(v_1, \ldots, v_k) = \neg\theta(v_1, \ldots, v_k) \lor \neg R_{U_n}(v_{i_1}, \ldots, v_{i_j})$. Therefore the formula $\gamma(v_1, \ldots, v_k) \lor (R_{\gamma, l_1, \ldots, l_j})(v_{l_1}, \ldots, v_{l_j})$ is in Ψ. Thus, $\mathbf{A}'', a_1, \ldots, a_k \models (R_{U(\gamma, l_1, \ldots, l_j)})(v_{l_1}, \ldots, v_{l_j})$ and in consequence $\langle a_{l_1} \ldots, a_{l_j} \rangle \in (R_{U(\gamma, l_1, \ldots, l_j)})^{\mathbf{A}''}$. Finally we have:

$$
\begin{aligned}
U(\gamma, l_1, \ldots, l_j) &= \{ \quad \langle b_{l_1} \ldots, b_{l_j} \rangle : \mathbf{B}', b_1, \ldots, b_k \not\models \gamma(v_1, \ldots, v_k)\} \\
&= \{ \quad \langle b_{l_1} \ldots, b_{l_j} \rangle : \mathbf{B}', b_1, \ldots, b_k \not\models \neg\theta(v_1, \ldots, v_k), \\
&\qquad \langle b_{i_1}, \ldots, b_{i_j} \rangle \in (R_{U_n})^{\mathbf{B}'} = U_n\} \\
&= \{ \quad \langle h(a_{l_1}), \ldots, h(a_{l_j}) \rangle : h \in \mathrm{hom}(\mathbf{A}_{|\{a_1, \ldots, a_k\}}, \mathbf{B}), \\
&\qquad \langle h(a_{i_1}), \ldots, h(a_{i_j}) \rangle \in U_n\}
\end{aligned}
$$

We shall show that $\mathbf{A}''$ satisfies condition (d). Let $1 \leq n, n' \leq m$ such that $U_n \subseteq U_{n'}$, and let $\langle a_{l_1}, \ldots, a_{l_j} \rangle \in (R_{U_n})^{\mathbf{A}''}$. Since $\neg R_{U_n}(v_{l_1}, \ldots, v_{l_j}) \lor R_{U_{n'}}(v_{l_1}, \ldots, v_{l_j})$ is in Ψ_3 we have that $\mathbf{A}', a_1, \ldots, a_k \models \neg R_{U_n}(v_{l_1}, \ldots, v_{l_j}) \lor R_{U_{n'}}(v_{l_1}, \ldots, v_{l_j})$ and, in consequence, $\langle a_{l_1}, \ldots, a_{l_j} \rangle \in (R_{U_{n'}})^{\mathbf{A}''}$.

For the converse, assume that $\mathbf{A}''$ is a relational winning strategy. We shall show that for all $a_1, \ldots, a_k$ (not necessarily different) elements in A and for every $\chi(v_1, \ldots, v_k)$ in Ψ, $\mathbf{A}', a_1, \ldots, a_k \models \chi(v_1, \ldots, v_k)$. First, if $\chi \in \Psi_1$ then $\chi = \neg R_\emptyset(v_{l_1}, \ldots, v_{l_j})$ for some $1 \leq l_1, \ldots, l_j \leq k$. In this case $\mathbf{A}', a_1, \ldots, a_k \models \chi(v_1, \ldots, v_k)$ since $\langle a_{l_1}, \ldots, a_{l_j} \rangle \notin (R_\emptyset)^{\mathbf{A}''} = \emptyset$. Secondly, if $\chi \in \Psi_2$ then we have $\chi(v_1, \ldots, v_k) = \gamma(v_1, \ldots, v_k) \lor R_{U(\gamma, l_1, \ldots, l_j)}(v_{l_1}, \ldots, v_{l_j})$ where $\gamma(v_1, \ldots, v_k)$ is a disjunctive monotone formula with at most one occurrence of a predicate in $\{R_{U_n} : 1 \leq n \leq m\}$. Assume first that $\gamma(v_1, \ldots, v_k)$ is a formula over $\tau \cup \{=\}$ (that is, it does not contain a second-order predicate). Thus, if $\mathbf{A}, a_1, \ldots, a_k \not\models \gamma(v_1, \ldots, v_k)$ implies the following. Let $\theta(v_1, \ldots, v_k) = \Theta(\mathbf{A}, a_1, \ldots, a_k)(v_1, \ldots, v_k)$. Then, by Proposition 2, $\theta(v_1, \ldots, v_k)$ implies $\neg\gamma(v_1, \ldots, v_m)$.

Thus

$$\begin{aligned} U(\gamma, l_1, \ldots, l_j) \quad &= \{\langle b_{l_1}, \ldots, b_{l_j} \rangle : \mathbf{B}', b_1, \ldots, b_k \not\models \gamma(v_1, \ldots, v_k)\} \\ &\supseteq \{\langle b_{l_1}, \ldots, b_{l_j} \rangle : \mathbf{B}', b_1, \ldots, b_k \models \theta(v_1, \ldots, v_k)\} \\ &= \{\langle b_{l_1}, \ldots, b_{l_j} \rangle : \mathbf{A}_{\{a_1,\ldots,a_k\}}, a_1, \ldots, a_k \longrightarrow \mathbf{B}, b_1, \ldots, b_k\} \\ &= \{\langle h(a_{l_1}), \ldots, h(a_{l_j}) : h \in \mathrm{hom}(\mathbf{A}_{|\{a_1,\ldots,a_k\}}, \mathbf{B})\} \end{aligned}$$

Let U be $\{\langle h(a_{l_1}), \ldots, h(a_{l_j}) : h \in \mathrm{hom}(\mathbf{A}_{|\{a_1,\ldots,a_k\}}, \mathbf{B})\}$. Since $\mathbf{A}''$ is a relational winning strategy, then $\langle a_{l_1} \ldots, a_{l_j} \rangle \in (R_U)^{\mathbf{A}''}$. Furthermore, since $U \subseteq U(\gamma, l_1, \ldots, l_j)$ we have that $\langle a_{l_1}, \ldots, a_{l_j} \rangle \in (R_{U(\gamma, l_1, \ldots, l_j)})^{\mathbf{A}''}$.

The proof when γ contains a predicate in $\{R_{U_n} : 1 \leq n \leq m\}$ is analogous. Finally it is straightforward to verify the case $\chi \in \Psi_3$ using condition (d) of relational winning strategy. ☐

Finally, the proof of Theorem 3 is a consequence of Claims 1, 2, and 3. ☐

Restricted Krom SNP formulas can be regarded alternatively as a particular type of Datalog programs called, linear Datalog Programs.

Let τ be a vocabulary. A *Datalog Program* over τ is a finite set of rules of the form

$$t_0 :\!- \ t_1, \ldots, t_m$$

where each t_i is an atomic formula $R(v_1, \ldots, v_m)$. The relational predicates that occur in the heads of the rules are the *intensional database* predicates (IDBs) and do not belong to τ, while all others are the *extensional database* predicates (EDBs) and must belong to τ. One of the IDBs is designated as the *goal* of the program. Note that IDBs may occur in the bodies of rules, and, thus, a Datalog program is a recursive specification of the IDBs with semantics obtained via least fixed-points of monotone operators (see [36]).

A Datalog Program is called linear if every rule contains at most one occurrence of a IDB in its body. Let $0 \leq j \leq k$ be non-negative integers, (j, k)-Datalog is said to be the collection of all Datalog programs in which every rule has at most k variables and at most j variables in the head.

Let us introduce with a bit more detail the semantics of Datalog Programs.

Let Q be a Datalog Program over τ, let τ_{IDB} be the set of its intensional predicates, and let $\tau' = \tau \cup \tau_{\mathrm{IDB}}$. The Datalog Program Q defines an operator $\Phi : \mathrm{STR}[\tau'] \to \mathrm{STR}[\tau']$ in the class of τ'-structures in the following way: For every τ'-structure $\mathbf{A}$, $\Phi(\mathbf{A})$ is defined to be the smallest τ'-structure with the same universe A of $\mathbf{A}$ such that:

- for every $R \in \tau'$, $R^{\mathbf{A}} \subseteq R^{\Phi(\mathbf{A})}$,
- for every rule $t_0 :\!- \ t_1, \ldots, t_m$ in Q, with variables $u_1, \ldots, u_n$, and for every interpretation $h : \{u_1, \ldots, u_n\} \to A$ such that

$$\mathbf{A}, h(u_1), \ldots, h(u_n) \models t_1 \wedge \cdots \wedge t_m,$$

we have that $\langle h(v_1), \ldots, h(v_r) \rangle \in P^{\Phi(\mathbf{A})}$ where $t_0 = P(v_1, \ldots, v_r)$

Since Φ is a monotone operator we can define its minimum fix point $\Phi^* : \mathrm{STR}[\tau'] \to \mathrm{STR}[\tau']$ as:

$$\Phi^*(\mathbf{A}) = \bigcup_{n \geq 0} \Phi^n(\mathbf{A}), \text{ for every } \mathbf{A}$$

where $\Phi^0(\mathbf{A}) = \mathbf{A}$ and $\Phi^{n+1}(\mathbf{A}) = \Phi(\Phi^n(\mathbf{A})), n \geq 0$. Alternatively, $\Phi^*(\mathbf{A})$ can be defined as the smallest τ'-structure with universe A such that $\mathbf{A} \subseteq \Phi^*(\mathbf{A})$ and $\Phi(\Phi^*(\mathbf{A})) = \Phi^*(\mathbf{A})$.

A distinguished IDB predicate P is designed to be the goal of the program. A Datalog program is a query that given a τ-structure $\mathbf{A}$, returns $P^{\Phi^*(\mathbf{A}')}$, where $\mathbf{A}'$ is the τ'-structure with the same universe of $\mathbf{A}$ and such that for all $R \in \tau$, $R^{\mathbf{A}'} = R^{\mathbf{A}}$ and for all $R \in \tau_{IDB}$, $R^{\mathbf{A}'} = \emptyset$. We say that a structure $\mathbf{A}$ is accepted by the Datalog Program Q iff $P^{\Phi^*(\mathbf{A}')} \neq \emptyset$.

Let $\mathcal{C}$ be a set of τ-structures and let Q be a Datalog Program. We say that $\mathcal{C}$ is defined by Q if for every τ-structure $\mathbf{A}$, $\mathbf{A}$ is in $\mathcal{C}$ iff $\mathbf{A}$ is accepted by Q. It is well-known and easy to see that if $\mathcal{C}$ is accepted by a Datalog Program then $\mathcal{C}$ must be a filter.

The following result relates linear Datalog programs with infinitary logics.

Theorem 4. Let $0 \le j \le k$ be non-negative integers, and let $\mathcal{C}$ be a set of τ-structures for some vocabulary τ. If $\mathcal{C}$ is definable by a linear (j,k)-Datalog Program then it is also definable in $M^{j,k}$.

Proof. Let Q be a linear (j,k)-Datalog Program. We will show that: (*) for every τ-structure $\mathbf{A}$, for every intensional predicate $R \in \tau_{\mathrm{IDB}}$, for every $n \ge 0$ and for every $\langle a_1, \ldots, a_r \rangle \in R^{\Phi^n(\mathbf{A}')}$, there exists some structure $\mathbf{B}$, some $b_1, \ldots, b_r \in B$, some path-decomposition $S_1, \ldots, S_s$ of width (j,k) of $\mathbf{B}$ with $\{b_1, \ldots, b_r\} \subseteq S_1$, such that $\langle b_1, \ldots, b_r \rangle \in R^{\Phi^n(\mathbf{B}')}$, and $\mathbf{B}, b_1, \ldots, b_r \longrightarrow \mathbf{A}, a_1, \ldots, a_r$. We will prove statement (*) by induction on n. The statement is vacuously true for $n = 0$. We will show that (*) holds for $n+1$ whenever it holds for n. Let $R \in \tau_{\mathrm{IDB}}$ be a intensional predicate and let $\langle a_1, \ldots, a_r \rangle$ be any tuple in $R^{\Phi^{n+1}(\mathbf{A}')}$. If $\langle a_1, \ldots, a_r \rangle$ is in $R^{\Phi^n(\mathbf{A}')}$ then we are done. Otherwise there exists some rule $R(y_1, \ldots, y_r) \text{ :- } t_1, \ldots, t_m$ in Q, over the variables $u_1, \ldots, u_{k'}$, $k' \le k$ and some mapping $h : \{u_1, \ldots, u_{k'}\} \to A$ such that $\Phi^n(\mathbf{A}'), h(u_1), \ldots, h(u_{k'}) \models t_1 \wedge \cdots \wedge t_m$, and $a_i = h(y_i)$, for every $1 \le i \le r$. Let us consider two cases: if the body of the rule does not contain any intensional predicate, then $\mathbf{A}, h(u_1), \ldots, h(u_{k'}) \models t_1 \wedge \cdots \wedge t_m$. Then, $\mathbf{A}_{|\{h(u_1), \ldots, h(u_{k'})\}}$ and $h(a_1), \ldots, h(a_{k'})$ satisfy statement (*). Otherwise, let us assume that the body contains one occurrence of an intensional predicate, say $t_1 = R_1(x_1, \ldots, x_l)$. In this case, since

$$\langle h(x_1), \ldots, h(x_l) \rangle \in R_1^{\Phi^n(\mathbf{A}')},$$

we can assume, by induction hypothesis, that there exists some $\mathbf{B}$ and some $b_1, \ldots, b_l$ in B, such that $\langle b_1, \ldots, b_r \rangle \in R^{\Phi^*(\mathbf{B}')}$ and $\mathbf{B}, b_1, \ldots, b_l \longrightarrow \mathbf{A}, h(x_1), \ldots, h(x_l)$. Furthermore there exists a path-decomposition $S_1, \ldots, S_s$ of $\mathbf{B}$ such that $\{b_1, \ldots, b_l\} \subseteq S_1$. We can assume, by renaming elements of B if necessary, that $b_i = h(x_i)$ for every $1 \le i \le l$ and that every other element of the universe of B does not belong to $\mathbf{A}$. Consider the structure $\mathbf{C}$ given as $\mathbf{B} \cup \mathbf{A}_{|\{h(u_1), \ldots, h(u_{k'})\}}$. By construction, $\Phi^n(\mathbf{C}'), h(u_1) \ldots, h(u_{k'}) \models t_1 \wedge \cdots \wedge t_m$ and consequently, $\langle h(x_1), \ldots, h(x_l) \rangle \in R^{\Phi^{n+1}(\mathbf{C}')}$.

Furthermore $\{h(u_1), \ldots, h(u_{k'})\}, S_1, \ldots, S_s$ is a path-decomposition of $\mathbf{C}$ of width (j,k).

In order to finish the proof, let $\mathcal{O}$ be the set containing all τ-structures of pathwidth (j,k) in $\mathcal{C}$. We shall see that $\mathcal{O}$ is an obstruction set of $\neg\mathcal{C}$. First, as $\mathcal{O}$ only contains structures in $\mathcal{C}$ and $\mathcal{C}$ is a filter, we can infer that for every structure $\mathbf{A}$, such that $\mathbf{B} \longrightarrow \mathbf{A}$ for some $\mathbf{B}$ in $\mathcal{O}$, $\mathbf{A}$ is in $\mathcal{C}$. Conversely, let $\mathbf{A}$ be any structure in $\mathcal{C}$. By the definition of acceptance by a Datalog Program, we can conclude that $P^{\Phi^*(\mathbf{A}')} \neq \emptyset$. By (*) we have that there exists some τ-structure $\mathbf{B}$ with pathwidth at most (j,k) such that $P^{\Phi^*(\mathbf{B}')} \neq \emptyset$ (and hence $\mathbf{B} \in \mathcal{O}$), and $\mathbf{B} \longrightarrow \mathbf{A}$. $\square$

Lemma 1. Let $0 \le j \le k$ be non-negative integers and let $\mathcal{C}$ be a collection of τ-structures. The two following sentences are equivalent:

(1) $\mathcal{C}$ is definable in linear (j, k)-Datalog.

(2) $\neg\mathcal{C}$ is definable in j-adic k-ary restricted Krom monotone SNP.

Furthermore, if $\neg\mathcal{C}$ is an ideal we also have that (1) and (2) are equivalent to:

(3) $\neg\mathcal{C}$ is definable in j-adic k-ary restricted Krom SNP with equalities.

Proof. To show the equivalence between **(1)** and **(2)** is straightforward but laborious. Here we will only sketch briefly the proof. To proof **(1)→(2)** we show that for any any linear (j, k)-Datalog program Q, there exists a sentence φ in j-adic k-ary restricted Krom monotone SNP, such that for every structure **A**, **A** $\models \varphi$ iff and only if Q does not accept **A** . The sentence φ is of the form $\exists R_1, \ldots, R_l \forall v_1, \ldots, v_k \psi(v_1, \ldots, v_k)$ where:

- $R_1, \ldots, R_l$ are the IDBs of the Datalog Program Q
- $v_1, \ldots, v_k$ are the variables occurring in the rules of the Datalog Program. Note: We can assume that the variables in the Datalog Program have been renamed so that every rule has its variables among $v_1, \ldots, v_k$
- $\psi(v_1, \ldots, v_k)$ is a CNF formula that is defined as

$$\psi(v_1, \ldots, v_k) = \neg P \wedge \bigwedge_{t_0 :- t_1, \ldots, t_m \in Q} t_0 \vee \neg t_1 \vee \cdots \vee \neg t_m$$

    where P is the goal predicate of Q.

It is immediate, although again laborious, to show that φ is a sentence in j-ary k-ary restricted Krom monotone SNP that is satisfied precisely by those structure that are not accepted by Q.

For the converse ((**2**)→(**1**)), let $\varphi = \exists R_1, \ldots, R_l \forall v_1, \ldots, v_k \psi(v_1, \ldots, v_k)$ be an arbitrary sentence in j-adic k-ary restricted Krom monotone SNP. We shall construct a linear (j, k)-Datalog Program Q in the following way:

- The IDBs of Q are precisely the second order predicates of φ, $R_1, \ldots, R_l$ plus a new 0-ary IDB, P, which will act as the goal predicate.
- The rules of Q are constructed from ψ in the following way. The formula ψ is a CNF and henceforth is the conjunction of several subformulas, where each one of this subformulas is the disjunction of atomic or negated atomic formulas. In fact, since we are dealing with a restricted Krom monotone SNP formula each one of this subformulas has to be of a very restricted form. More precisely, every one of the subformulas that constitute ψ has to be either the disjunction of negated atomic formulas $\neg t_1 \vee \cdots \vee \neg t_m$, or it can contain one unnegated atomic formula, that is, it has to be of the form $t_o \vee \neg t_1 \vee \cdots \vee \neg t_m$ where the underlying predicate of t_0 is an IDB.

    Datalog Program Q contains a rule per each subformula in ψ. If the subformula is of the form $\neg t_1 \vee \cdots \vee \neg t_m$ then the rule added to Q is $P :- t_1, \ldots, t_m$. Otherwise, if the subformula is of the form $t_0 \vee \neg t_1 \vee \cdots \vee \neg t_m$ then the rule is of the form $t_0 :- t_1, \ldots, t_m$.

Again it is an easy exercise to show that the Datalog Program Q defined is indeed a linear (j, k)-Datalog Program that accepts precisely those structures that falsify φ.

To see that if $\mathcal{C}$ is a filter then **(3)** implies **(1)** we make use of a result by Feder and Vardi [11] which states that for every Datalog($\neq, \neg$) Program P, that is, every Datalog Program in which we allow inequality and the negation of EDBs, that defines a filter $\mathcal{C}$, there exists a Datalog Program P', that defines the same set of structures $\mathcal{C}$. Furthermore,

a closer inspection to the proof in [11] reveals that if P is in linear (j,k)-Datalog($\neq,\neg$), then P' is in linear (j,k)-Datalog..

The proof that (**3**) implies (**1**) mimics that of (**2**) implies (**1**). As before let $\varphi = \exists R_1,\ldots,R_l \forall v_1,\ldots,v_k \psi(v_1,\ldots,v_k)$ be an arbitrary sentence in j-adic k-ary restricted Krom SNP with equalities. We shall construct a linear (j,k)-Datalog Program Q in a similar fashion:

- The IDBs of Q are precisely the second order predicates of φ, $R_1,\ldots,R_l$ plus a new 0-ary IDB, P, which will act as the goal predicate.
- The rules of Q are constructed from ψ in the following way. The formula ψ is a CNF and henceforth is the conjunction of several subformulas of the form $t_1 \vee \cdots \vee t_m$, where each t_i, $1 \leq i \leq m$ is an atomic or a negated atomic formula. Since the formula ψ is not supposed to be monotone and might contain equalities we only can assume that each one of the disjunctions $t_1 \vee \cdots \vee t_m$ of ψ has at most one positive occurrence of a second-order variable and at most one negative occurrence of a second order-variable. As before, we add to Q a rule per each disjunction $t_1 \vee \cdots \vee t_m$ in ψ. If $t_1 \vee \cdots \vee t_m$ does not contain any positive occurrence of an IDB, then the rule added to Q is $P :\!- \ \neg t_1,\ldots,\neg t_m$. Otherwise, if $t_1 \vee \cdots \vee t_m$ contains one positive occurrence of an IDB, say t_1, then the rule added to Q is $t_1 :\!- \ \neg t_2,\ldots,\neg t_m$.

Observe that the body of a rule in Q might contain equalities and negated atomic formulas $\neg R(x_1,\ldots,x_r)$ where R is an EDB. Consequently, Q is a Datalog($\neq,\neg$) Program. It is easy to see that Q is, indeed, a linear (j,k)-Datalog($\neq,\neg$) Program and that Q accepts accepts precisely those structures that falsify φ. Since $\mathcal{C}$ is a filter, by the result in [11] mentioned above there exists a linear (j,k)-Datalog Program (that is, without inequalities and negations) that defines $\mathcal{C}$.

The implication ((**2**)→(**3**)) is trivial.   □

If we are dealing with set of structures $\neg\mathcal{C}$ of the form $I(\mathbf{B})$ for some τ-structure $\mathbf{B}$, then we can combine Theorem 2, Lemma 1, Theorem 3, and Theorem 4, obtaining the main result of this paper.

Theorem 5. Let $0 \leq j \leq k$ be non-negative integer, let $\mathbf{B}$ be a τ-structure, and let $\neg\mathcal{C} = \mathrm{CSP}(\mathbf{B}) = I(\mathbf{B})$. The following sentences are equivalent:

(1) The class $\mathcal{C}$ is $M^{j,k}$-definable.
(2) The class $\mathcal{C}$ is $N^{j,k}$-definable.
(3) The class $\mathcal{C}$ is definable in linear (j,k)-Datalog.
(4) If $\mathbf{A}$ and $\mathbf{B}$ are finite structures such that $\mathbf{A} \in \mathcal{C}$ and Duplicator has a winning strategy for the (j,k)-PR game on $\mathbf{A}$ and $\mathbf{B}$, then $\mathbf{B} \in \mathcal{C}$.
(5) The class $\neg\mathcal{C}$ has an obstruction set with pathwidth at most (j,k).
(6) The class $\neg\mathcal{C}$ is definable in j-adic k-ary restricted Krom SNP with equalities.
(7) The class $\neg\mathcal{C}$ is definable in j-adic k-ary restricted Krom monotone SNP.

Let $\mathbf{B}$ be a τ structure. If $\neg\mathcal{C} = I(\mathbf{B})$ satisfies any of the conditions of Theorem 5 we say that $\mathbf{B}$ has (j,k)-*path duality*. Finally, we say that $\mathbf{B}$ has *bounded path duality* if $\mathbf{B}$ has (j,k)-path duality for some $0 \leq j < k$.

7. Applications in Computational Complexity

In this section we bring computational complexity into the picture. We shall start by proving that for every finite structure $\mathbf{B}$ with bounded path duality, CSP($\mathbf{B}$) is in NL.

This result is an immediate consequence of the following theorem, proven in [14].

Theorem 6 ([14]). Let φ be a Krom CNF formula over a vocabulary τ. The problem of deciding, given a τ-structure $\mathbf{A}$, whether $\mathbf{A} \models \varphi$, is solvable in NL.

Indeed, if $\mathbf{B}$ has bounded path duality, then CSP($\mathbf{B}$) is definable in Krom SNP. Hence we have,

Proposition 3. Let $\mathbf{B}$ be a τ-structure with bounded path duality. Then CSP($\mathbf{B}$) is in NL.

In what follows, we shall revisit all families of constraint satisfaction problems, that up to the best of our knowledge, are known to be in NL. Our goal here is to show that the notion of bounded path duality provides a unifying framework that encompasses and explains this results. In the last part of this section, we shall see how the notion of bounded path duality can be used to place new problems in NL.

7.1. **Implicational constraints.** The class of *implicational constraints* was introduced independently by Kirousis [23] and by Cooper et al. [4] (in this second reference they were named *0/1/all constraints*). A binary relation $R \subseteq A^2$ over a finite domain A is called implicational if it has one of the following three forms: (1) $B \times C$ for some $B, C \subseteq A$, (2) $\{(a, f(a)) : a \in B\}$ for some $B \subseteq A$ and some one-to-one mapping $f : B \to A$, or (3) $(\{b\} \times C) \cup (B \times \{c\})$ for some $B, C \subseteq A$, $b \in B$, and $c \in C$. A structure $\mathbf{B}$ is said to be implicational if so are all its relations. It is easy to observe that 2-SAT can be encoded as a constraint satisfaction problem CSP($\mathbf{B}_{\text{2-SAT}}$), with $\mathbf{B}_{\text{2-SAT}}$ implicational:

The signature τ of $\mathbf{B}_{\text{2-SAT}}$ contains three binary relation symbols P_0, P_1 and P_2. The universe of $\mathbf{B}$ is $\{0, 1\}$ and the values of $P_0^{\mathbf{B}_{\text{2-SAT}}}$, $P_1^{\mathbf{B}_{\text{2-SAT}}}$, and $P_2^{\mathbf{B}_{\text{2-SAT}}}$ are respectively $\{0, 1\}^2 / \{(0, 0)\}$, $\{0, 1\}^2 / \{(0, 1)\}$, and $\{0, 1\}^2 / \{(1, 1)\}$. Observe that each one of these relations is implicational.

It is well known and quite easy to see that 2-SAT and CSP($\mathbf{B}_{\text{2-SAT}}$) are logspace reducible to each other. Indeed, let φ be any arbitrary 2-CNF formula. We shall show that it is possible to construct in logarithmic space a τ-structure $\mathbf{A}$ such that $\mathbf{A}$ is homomorphic to $\mathbf{B}_{\text{2-SAT}}$ if and only if φ has a solution. The universe A of $\mathbf{A}$ is given by the variables of φ. Furthermore, $P_0^{\mathbf{A}}$ contains for each clause $(x \vee y)$ with only positive literals, the tuple (x, y). Observe that every homomorphism from $\mathbf{A}$ to $\mathbf{B}$ should set values to x and y that belong to $P_0^{\mathbf{B}_{\text{2-SAT}}} = \{0, 1\}^2 / \{0, 0\}$. Hence the values set to x and y must satisfy clause $(x \vee y)$.

A similar processing is applied to the other types of clauses. More precisely, for each clause with exactly one positive literal $(x \vee \neg y)$ (let us assume that the literals are ordered so that the positive literal comes first) we add the tuple (x, y) to $P_1^{\mathbf{A}}$ and for each clause $(\neg x \vee \neg y)$ with only negative literals we add the tuple (x, y) to $P_2^{\mathbf{A}}$. It is very easy to verify that satisfying assignments of φ correspond to homomorphisms from $\mathbf{A}$ to $\mathbf{B}_{\text{2-SAT}}$ and that this reduction can be performed in logarithmic space. The reduction from CSP($\mathbf{B}_{\text{2-SAT}}$) to 2-SAT is also very simple and we omit it.

Let $\mathbf{B}$ be any implicational structure and let $\mathbf{A}$ be any instance of CSP($\mathbf{B}$). We first recall a well-known result about constraint satisfaction problems CSP($\mathbf{B}$) with $\mathbf{B}$

implicational which can be found in [23]. If $\mathbf{A}$ is an instance of CSP($\mathbf{B}$) we define the conflict graph G associated to $\mathbf{A}$ and $\mathbf{B}$ as the digraph $G = (V, E)$, with set of nodes $V = \{(a, b) : a \in A, b \in B\} \cup \{\square\}$ and set of edges E constructed in the following way:

(a) For every predicate R in the vocabulary τ of $\mathbf{A}$, for every $b, b' \in B$ such that $R^{\mathbf{B}} \cap (\{b\} \times B) = \{(b, b')\}$, and for every tuple $(a, a') \in R^{\mathbf{A}}$ we add to E an arc from (a, b) to (a', b').

(b) For every predicate R in the vocabulary τ of $\mathbf{A}$, for every $b, b' \in B$ such that $R^{\mathbf{B}} \cap (B \times \{b'\}) = \{(b, b')\}$, and for every tuple $(a, a') \in R^{\mathbf{A}}$ we add to E an arc from (a', b') to (a, b)).

(c) For every predicate R in the vocabulary τ of $\mathbf{A}$, for every $b \in B$ such that $R^{\mathbf{B}} \cap (\{b\} \times B) = \emptyset$, and for every tuple $(a, a') \in R^{\mathbf{A}}$ we add to E an arc from (a, b) to $\square$.

(d) For every predicate R in the vocabulary τ of $\mathbf{A}$, for every $b' \in B$ such that $R^{\mathbf{B}} \cap (B \times \{b'\}) = \emptyset$, and for every tuple $(a, a') \in R^{\mathbf{A}}$ we add to E an arc from (a', b') to $\square$.

Lemma 2. $\mathbf{A}$ is homomorphic to $\mathbf{B}$ iff for every $a \in A$ there exists some $b \in B$ such that there is not a path from (a, b) to any some node in $\{(a, b') : b' \in B \; \{b\}\} \cup \{\square\}$.

Let us see that this implies that each implicational $\mathbf{B}$ has $(2, 3)$-path duality.

Lemma 3. Every implicational structure $\mathbf{B}$ has $(2, 3)$-path duality.

Proof. Let $\mathbf{A}$ be any structure not homomorphic to $\mathbf{B}$ and let G be the conflict graph associated to $\mathbf{A}$ and $\mathbf{B}$. Let $\{b_1, \ldots, b_m\}$ be the universe B of $\mathbf{B}$. We shall construct a structure $\mathbf{P}$ with pathwidth at most $(2, 3)$ that is homomorphic to $\mathbf{A}$ but is not homomorphic to $\mathbf{B}$.

By Lemma 2 there exists some $a^* \in A$ such that for all b_i, $1 \leq i \leq m$, there is a path in G from (a^*, b_i) to either (a^*, b_j) for some $j \neq i$ or $\square$. Let us denote by $(a^*, b_i) = v_1^i, v_2^i, \ldots, v_{l_i}^i$ the elements of such path. Observe that $v_{l_i}^i$ is either (a^*, b_j) for some $j \neq i$ or $\square$.

The universe P of $\mathbf{P}$ contains for each $1 \leq i \leq m$, and for each node v_k^i, $1 \leq k \leq l_i$ in the path corresponding to b_i an element w_k^i. If the same node of A appears in several paths then we make different copies of it. Finally we merge nodes w_k^i, corresponding to $v_k^i = (a^*, b_l)$ for some $1 \leq l \leq m$ into a unique node that we shall call w. Observe for all for all $1 \leq i \leq m$, w_1^i is w.

Now we shall construct the relations of $\mathbf{P}$ and, at the same time, a homomorphism h from $\mathbf{P}$ to $\mathbf{A}$.

Let i be any integer with $1 \leq i \leq m$ and let v_j^i, v_{j+1}^i be any two consecutive elements in the path corresponding to b_i. Then, (v_j^i, v_{j+1}^i) is an arc of G that must have been added according to (a), (b), (c) or (d). If (v_j^i, v_{j+1}^i) has been added according to (a) then there exists some predicate R in τ, some $b, b' \in B$ such that $R^{\mathbf{B}} \cap (\{b\} \times B) = \{(b, b')\}$, and some tuple $(a, a') \in R^{\mathbf{A}}$. Then we add to $R^{\mathbf{P}}$ the tuple (w_{j+1}^i, w_{j+1}^i). The mapping h maps v_j^i to a and v_{j+1}^i to a'. Similarly, if (v_j^i, v_{j+1}^i) has been added according to (b) then there exists some predicate R in τ, some $b, b' \in B$ such that $R^{\mathbf{B}} \cap (B \times \{b'\}) = \{(b, b')\}$, and some tuple $(a, a') \in R^{\mathbf{A}}$. In this case we add (w_{j+1}^i, w_j^i) to $R^{\mathbf{P}}$ and set $h(v_j^i) = a'$ and $h(v_{j+1}^i) = a$. The cases (c) and (d) are dealt with in a similar fashion.

By construction, h defines a homomorphism from $\mathbf{P}$ to $\mathbf{A}$. We shall now prove that $\mathbf{P}$ is not homomorphic to $\mathbf{B}$. Towards a contradiction let us assume that there exists such a

homomorphism f and let b_i be the image of w according to f. Let us consider the nodes in $\mathbf{P}$, $w = w_1^i, w_2^i, \ldots, w_{l_i}^i$ associated to the path $v_1^i, v_2^i, \ldots, v_{l_1}^i$. The first arc (v_1^i, v_2^i) of the path has been added due to one of the conditions $(a - d)$. If it was added due to (a) then there exists some predicate R in τ, some $b, b' \in B$ such that $R^{\mathbf{B}} \cap (\{b\} \times B) = \{(b, b')\}$. Furthermore (w_1^i, w_2^i) belongs to $R^{\mathbf{P}}$. Consequently, if f is a homomorphism it must map w_2^i to b'. We can do a similar reasoning for the other conditions $(b-c)$ and for the remaining nodes of the path until we reach the last element of the path $v_{l_i}^i$. Here we should consider two cases. If the arc $(v_{l_i-1}^i, v_{l_i}^i)$ has been added due to conditions (a) or (b), then can conclude that $w_{l_i}^i$ is w and $f(w) = b_j$ for some $j \neq i$, obtaining a contradiction. Otherwise, If the arc $(v_{l_i-1}^i, v_{l_i}^i)$ has been added due to conditions (c) or (d), then we can conclude that f is not a homomorphism, getting again a contradiction.

It only remains to show that $\mathbf{P}$ has pathwidth at most $(2,3)$. The key observation here is that each one of paths $v_1^i, \ldots v_{l_i}^i$ gives rise to a collection of nodes $w_1^i, \ldots, w_{l_i}^i$ of P such that the restriction $\mathbf{P}_{|\{w_1^i, \ldots, w_{l_i}^i\}}$ of $\mathbf{P}$ to $\{w_1^i, \ldots, w_{l_i}^i\}$ has the following path-decomposition: $\{w_1^i, w_2^i\}, \ldots, \{w_{l_i-1}^i, w_{l_i}^i\}$. Consequently, since w is the only common node to each such restriction we can easily conclude that $\mathbf{P}$ has the path-decomposition:

$$\{w, w_1^1, w_2^1\}, \{w, w_2^1, w_3^1\}, \ldots, \{w, w_{l_1-1}^1, w_{l_1}^1\}, \{w, w_1^2, w_2^2\}, \ldots, \{w, w_{l_m-1}^m, w_{l_m}^m\}$$

$\square$

Using a very similar line of reasoning is it possible to push this results a bit further. In particular it is possible to show [7] that every structure $\mathbf{B}$ that contains only relations invariant under an operation in the pseudovariety generated by all dual discriminator operations has also bounded path duality. A complete presentation of this result would require a lengthy introduction of the algebraic approach to CSP. Since this would lead us out of the scope of this paper we omit it and instead we refer to [7].

Another related family of constraint satisfaction problems solvable in NL is the class of all CSP($\mathbf{B}$) where $\mathbf{B}$ is a poset with constants invariant under a near-unanimity operation [28]. Again for space limitations we shall not present this result. In [28] it is shown that for every such poset $\mathbf{B}$, $\neg$ CSP($\mathbf{B}$) is definable in pos FO+TC, that is, the fragment of FO+TC in which negation and universal quantification is not allowed. It is known (folklore) that pos FO+TC and linear Datalog have the same expressive power. Consequently, we can infer that $\mathbf{B}$ has bounded path duality.

7.2. **Implicative Hitting-Set Bounded.** The class of Implicative Hitting-Set Bounded was introduced in [5]. Let k be any integer greater than 1. A Boolean relation R is in k-IHS-B+ if it can be expressed as a CNF where each clause is of the form $\neg v$, $\neg v \vee w$ or $w_1 \vee \cdots \vee w_k$ (here we do not require that all w_i's are different). Similarly, R is in k-IHS-B− if it can be expressed as a CNF where each clause is of the form w, $\neg v \vee w$ or $\neg v_1 \vee \cdots \vee \neg v_k$. A relational structure $\mathbf{B}$ is implicative Hitting-Set Bounded there exists some $k \geq 2$ such that all its relations are in k-IHS-B+ or k-IHS-B−. It is well known [5] that for every Implicative Hitting-Set Bounded structure $\mathbf{B}$, CSP($\mathbf{B}$) is solvable in NL. We shall prove that it has bounded path width duality.

Lemma 4. Let $\mathbf{B}$ be a relational structure containing only relations in k-IHS-B+ for some $k \geq 2$. Then $\mathbf{B}$ has $(k, k-1+\rho(\mathbf{B}))$-path duality. Similarly, if $\mathbf{B}$ contains only relations in k-IHB-B- then $\mathbf{B}$ has $(k, k-1+\rho(\mathbf{B}))$-path duality.

Proof. We shall consider only the case in which all relations of **B** are in k-IHS-B+. The case k-IHS-B− is completely symmetric. We need to introduce a bit of notation. Recall that given a relation T and a subset I of $\{1, \ldots, \rho(T)\}$, $T_{|I}$ denotes the restriction of T to I. We generalize slightly this definition by allowing sequences of integers instead of merely sets. More formally, let $i_1, \ldots, i_k$ be (not necessarily different) integers in $\{1, \ldots, \rho(T)\}$. By $T_{|i_1, \ldots, i_k}$ we denote the k-ary relation

$$\{(a_{i_1}, \ldots, a_{i_k}) : (a_1, \ldots, a_{\rho(T)}) \in T\}$$

Let us start by introducing a property of unsatisfiable formulas which will be of most help in providing intuition on the proof.

Lemma 5. [5] Let Φ be a CNF formula containing only clauses of the form $\neg v$, $\neg v \vee w$ or $w_1 \vee \cdots \vee w_k$. If Φ is unsatisfiable then there exists a clause of the form $w_1 \vee \cdots \vee w_k$ such that for each $i \in \{1, \ldots, k\}$ there exists a sequence of variables $w_i = v_1^i, \ldots, v_{l_i}^i$ such that $\neg v_{l_i}^i$, an $(\neg v_j^i \vee v_{j+1}^i)$ for all $j \in \{1, \ldots, l_i - 1\}$ are in Φ

In order to prove the theorem we need a generalization of this result. Let **A** be an instance of CSP(**B**) and let h be a partial mapping from the universe A to B. In what follows we shall denote a partial mapping as $(a_1 \to b_1, \ldots, a_m \to b_m)$ where $\{a_1, \ldots, a_m\}$ is the domain of the partial mapping and for each $1 \le i \le m$, a_i is map to b_i. We say that a partial mapping is *forbidden* if there exists a predicate R in the vocabulary τ of **B** and some integers $i_1, \ldots, i_m$ such that $(a_1, \ldots, a_m) \in R^{\mathbf{A}}_{|i_1, \ldots, i_m}$ and $(b_1, \ldots, b_m) \notin R^{\mathbf{B}}_{|i_1, \ldots, i_m}$.

Lemma 6. Let **A** be any unsatisfiable instance of CSP(**B**) where **B** contains only relations in k-IHS-B+. Then there exists some elements a_j^i, $1 \le i \le k$, $1 \le j \le l_i$ of A such that $(a_1^1 \to 0, \ldots, a_1^k \to 0)$, $(a_{l_i}^i \to 1)$, $1 \le i \le k$, and $(a_j^i \to 1, a_{j+1}^i \to 0)$, $1 \le i \le k$, $1 \le j \le l_i - 1$ are forbidden partial mappings

Proof. The proof of this lemma is rather straightforward. We only present here an sketch of the proof. In a first step, taking into account that every relation in **B** is in k-IHS-B+, it is possible to construct from **A** and **B**, an unsatisfiable CNF formula in which the variables are elements of A and every clause is of the form $\neg v$, $\neg v \vee w$, or $w_1 \vee \cdots \vee w_k$. Observe that we can associate to each clause $l_1 \vee \cdots \vee l_r$ in Φ a forbidden partial mapping $(v_1 \to b_1, \ldots, v_r \to b_r)$ where v_i is the variable underlying literal l_i and b_i is 0 if l_i is positive and 1 otherwise. Finally the result follows by applying Lemma 5 to formula Φ. $\square$

We are now in a position to construct a τ-structure **P** of $(k, k - 1 + \rho(\mathbf{B}))$ that is homomorphic to **A** but not homomorphic to **B**. The universe P of **P** contains for each $1 \le i \le k$ and for each element a_j^i, $1 \le j \le l_i$ in the sequence guaranteed by the previous lemma, an element c_j^i. If the same element appears more than once then we make different copies. The universe of P contains also more elements that will be added as needed.

We now construct the relations of **P**. At the same time we shall define a homomorphism h from **P** to **A**. The intuition of the construction is very similar to the proof of Lemma 3.

Let $(a_1^1 \to 0, \ldots, a_1^k \to 0)$ be the first of the forbidden partial mappings guaranteed to exist by Lemma 6. Consequently there exists some predicate symbol R in τ, some tuple $(v_1, \ldots, v_{\rho(R)}) \in R^{\mathbf{A}}$ and some $j_1, \ldots, j_k \in \{1, \ldots, \rho(R)\}$ such that $v_{j_1} = a_1^1, \ldots, v_{j_k} = a_1^k$. Then we include in $R^{\mathbf{P}}$ a tuple $(w_1, \ldots, w_{\rho(R)})$ containing c_1^i in its j_ith position, $i \in \{1, \ldots, w_k\}$ and new variables not occurring in P elsewhere. We set $h(w_l)$ to be v_l. We associate to the forbidden partial mapping $(a_1^1 \to 0, \ldots, a_1^k \to 0)$ a set $S_{(a_1^1 \to 0, \ldots, a_1^k \to 0)}$ that contains all variables $\{w_1, \ldots, w_\rho(R)\}$ (we shall use this set later to define the path

decomposition of $\mathbf{P}$). We proceed in a similar fashion for each forbidden partial mapping guaranteed by Lemma 6.

By construction h defines a homomorphism from $\mathbf{P}$ to $\mathbf{A}$. Furthermore $\mathbf{P}$ has has pathwidth at most $(k, k - 1 + \rho(\mathbf{B}))$ as it is certified by the following path-decomposition:

$$S_{(a_1^1 \to 0, \ldots, a_1^k \to 0)}, S_{(a_1^1 \to 1, a_2^1 \to 0)} \cup \{c_1^2, \ldots, c_1^k\}, \ldots, S_{(a_{l_1-1}^1 \to 1, a_{l_1}^1 \to 0)} \cup \{c_1^2, \ldots, c_1^k\},$$
$$S^{(a_{l_i}^1 \to 1)} \cup \{c_1^2, \ldots, c_1^k\}, S^{(a_1^2 \to 1, a_2^2 \to 0)} \cup \{c_1^3, \ldots, c_1^k\}, \ldots, S_{(a_{l_k-1}^k \to 1, a_{l_k}^k \to 0)}, S_{(a_{l_k}^k \to 1)}$$

Finally, we shall now see that $\mathbf{P}$ is not homomorphic to $\mathbf{B}$. Towards a contradiction, let f be such homomorphism. By the construction of $\mathbf{P}$ we can conclude that $(c_1^1 \to 0, \ldots, c_1^k \to 0)$ is a forbidden mapping of $\mathbf{P}$ and $\mathbf{B}$. Consequently, for some i, $f(c_1^i) = 1$. Consider now the sequence $c_1^i, c_2^i, \ldots, c_{l_i}^i$. By the construction of $\mathbf{P}$ for each $j \in \{1, \ldots, l_i - 1\}$, $(c_j^i \to 1, c_{j+1}^i \to 0)$ is a forbidden mapping. Consequently we can infer by induction that for every $j \in \{1, \ldots, l_i\}$, $f(c_j^i) = 1$. We get then a contradiction with the fact that $(c_{l_i}^i \to 1)$ is a forbidden mapping. $\qquad\square$

7.3. **New problems in NL.** In this section we shall use the notion of bounded path as a tool to identify some other constraint satisfaction problems in NL. We have mentioned already in Section 7.1 that every structure invariant under an operation in the pseudovariety generated by all dual discriminator operations has bounded path duality. More examples of constraint satisfaction problems CSP($\mathbf{B}$), such that $\mathbf{B}$ has bounded path duality can be found in the literature about $\mathbf{H}$-coloring which can be reformulated as the subcase of the general constraint satisfaction problem CSP($\mathbf{H}$) when $\mathbf{H}$ is a (di)graph. As mentioned in the introduction, the notion of bounded tree duality which is intimately related to the notion of path duality (and in fact, inspired it) has been deeply investigated in the field of $\mathbf{H}$-coloring. Let us recall that a (di)graph $\mathbf{H}$ has has bounded tree duality if CSP($\mathbf{B}$) has an obstruction set containing only graphs of treewidth at most k for some fixed k. Most of the tractable cases of the $\mathbf{H}$-coloring posses bounded tree duality. Indeed, a closer inspection of those results shows that most times, the obstruction set corresponding to a given (di)graph $\mathbf{H}$ contains only structures with bounded pathwidth. Consequently, as a direct consequence of Proposition 3, we can lower the complexity of this problems from P to NL with virtually no effort.

Let us revisit now some of these examples. For undirected graphs, the only case solvable in polynomial time corresponds to bipartite graphs [16]. It is fairly easy to see that the set of all odd cycles is an obstruction set of CSP($\mathbf{H}$) for any bipartite graph $\mathbf{H}$. As cycles have pathwidth at most $(2, 3)$ we can conclude that every bipartite graph $\mathbf{H}$ has $(2, 3)$-path duality. We have thus strengthened the dichotomy result of [16]: for every undirected graph $\mathbf{H}$, CSP($\mathbf{H}$) is in NL or NP-complete.

For directed graphs, it is a direct consequence of the results in [18, 19] (see also [17]) that every oriented path, directed cycle, and even more generally, every unbalanced oriented cycle has bounded path duality.

Let us present these results. An oriented path is a digraph $\mathbf{H}$ with nodes $p_0, \ldots, p_n$ such that for each $i \in \{0, \ldots, n - 1\}$ either (p_i, p_{i+1}) or (p_{i+1}, p_i) is an edge of $\mathbf{H}$ and it does not contain any other edge. It was shown in [18] that for every oriented path $\mathbf{H}$, CSP($\mathbf{H}$) has an obstruction set containing only oriented paths. It is easy to observe that the Gaifman

graph of an oriented path is a path and hence has pathwidth at most $(1, 2)$. Consequently, oriented paths have $(1, 2)$-path duality.

An oriented cycle is a digraph H with nodes $p_0, \ldots, p_n$ such that for each $i \in \{0, \ldots, n\}$, either (p_i, p_{i+1}) or (p_{i+1}, p_i) is an edge of $\mathbf{H}$ (the sum is modulo $n + 1$). If additionally all edges are in the same direction then $\mathbf{H}$ is called a directed path. More generally, if the number of forward edges is different than the number of backward edges then $\mathbf{H}$ is called unbalanced.

In [19], it is shown that for each unbalanced oriented cycle $\mathbf{H}$, CSP($\mathbf{H}$) has an obstruction set containing only oriented cycles. As oriented cycles have pathwidth at most $(2, 3)$ we can conclude that $\mathbf{H}$ has $(2, 3)$-path duality.

We conclude this section by examining the problem of deciding whether a given structure $\mathbf{B}$ has bounded path duality. In general, this problem is not known to be decidable. However, in a particular case it is possible to show decidability, by mimicking some results originally proven in [12].

Theorem 7. For every $1 \leq k$, the problem of deciding whether a τ-structure $\mathbf{B}$ has $(1, k)$-path duality is decidable.

Proof. By Theorem 3, there exists a sentence φ in 1-adic (also called monadic) monotone SNP such that for every τ-structure $\mathbf{A}$, $\mathbf{A} \models \varphi$ iff the Duplicator has a winning strategy for the $(1, k)$-PR game on $\mathbf{A}$ and $\mathbf{B}$. It is widely known (see [12]) that there exists a sentence ψ in monadic monotone SNP such that for every τ-structure $\mathbf{A}$, $\mathbf{A} \models \psi$ iff $\mathbf{A} \in \text{CSP}(\mathbf{B})$. Thus deciding $(1, k)$-path duality is equivalent to decide whether $(\varphi \implies \psi)$ is a tautology for finite structures. The latter is decidable since so is containment for monadic monotone SNP [12]. $\square$

8. Open Problems

It is unknown whether bounded path duality captures the class of CSPs solvable in NL, i.e, whether there exists a structure that does not have bounded path duality such that CSP($\mathbf{B}$) is solvable in NL. In this direction, it is not difficult to show, using some results in [14], that for every structure $\mathbf{B}$, if CSP($\mathbf{B}$) is in NL, then it can be expressed in restricted Krom monadic SNP with equality, $s(x, y)$ (true when y is the immediate successor of x in some total ordering), 0 and max (first and last elements of the total ordering respectively). Here we understand that the homomorphism condition only applies to non-built-in predicates.

Another interesting problem is to extend the class of structures $\mathbf{B}$ known to have bounded path duality via closure conditions. With that respect, it would be interesting to decide whether majority operations, or even more generally, near-unanimity operations [35] (see also [21]) guarantee bounded path duality. This result is true, in particular, for structures with domain of size at most 2.

9. Acknowledgments

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] A.K. Chandra and P.M. Merlin. Optimal implementation of conjunctive queries in relational databases. In *9th Symp. on Theory of Computing (STOC'77)*, pages 77–90, 1977.

[3] S.A. Cook. The Complexity of Theorem-Proving Procedures. In *3rd Annual ACM Symposium on Theory of Computing, (STOC'71)*, pages 151–158, 1971.

[4] M.C. Cooper, D.A. Cohen, and P.G. Jeavons. Characterizing Tractable Constraints. *Artificial Intelligence*, 65:347–361, 1994.

[5] N. Creignou, S. Khanna, and M. Sudan. *Complexity Classification of Boolean Constraint Satisfaction Problems*, volume 7 of *Monographs on Discrete Mathematics and Applications*. SIAM, 2001.

[6] V. Dalmau. A New Tractable Class of Constraint Satisfaction Problems. In *6th International Symposium on Artificial Intelligence and Mathematics*, 2000.

[7] V. Dalmau. Constraint satisfaction problems in non-deterministic logarithmic space. In *29th International Colloquium on Automata, Languages, and Programming (ICALP'02)*, pages 414–425, 2002.

[8] V. Dalmau, P.G. Kolaitis, and M. Vardi. Constraint Satisfaction Problems, Bounded Treewidth, and Finite-Variable Logics. In *8th International Conference on Principles and Practice of Constraint Programming (CP'02)*, pages 310–326.

[9] V. Dalmau and J. Pearson. Set Functions and Width 1. In *5th International Conference on Principles and Practice of Constraint Programming, (CP'99)*, volume 1713 of *Lecture Notes in Computer Science*, pages 159–173, Berlin/New York, 1999. Springer-Verlag.

[10] R. Dechter and J. Pearl. Network-based Heuristics for Constraint Satisfaction Problems. *Artificial Intelligence*, 34(1):1–38, 1988.

[11] T. Feder and M. Vardi. Homomorphism closed vs. existential positive. In *18th IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 311–320, 2003.

[12] T. Feder and M.Y. Vardi. The Computational Structure of Monotone Monadic SNP and Contraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Computing*, 28(1):57–104, 1998.

[13] E.C. Freuder. A Sufficient Condition for Backtrack-bounded Search. *Journal of the ACM*, 32:755–761, 1985.

[14] E. Grädel. Capturing Complexity Classes by Fragments of Second-Order Logic. *Theoretical Computer Science*, 101(1):35–57, 1992.

[15] M. Grohe. The Complexity of Homomorphism and Constraint Satisfaction Problems seen from the Other Side. In *Proceedings of the 44th IEEE Symposium on Foundations of Comupter Science, (FOCS'03)*, pages 552–561, 2003.

[16] P. Hell and J. Nešetřil. On the Complexity of H-coloring. *J. Comb. Theory, Series B*, 48:92–110, 1990.

[17] P. Hell, J. Nešetřil, and X. Zhu. Duality and polynomial testing of graph homomorphisms. *Trans. Amer. Math. Soc.*, 348:1281–1297, 1996.

[18] P. Hell and X. Zhu. Homomorphisms to oriented paths. *Discrete Mathematics*, 132:107–114, 1994.

[19] P. Hell and X. Zhu. The Existence of Homomorphisms to Oriented Cycles. *SIAM J. Discrete Math.*, 8, 1995.

[20] P. Jeavons. On the Algebraic Structure of Combinatorial Problems. *Theoretical Computer Science*, 200:185–204, 1998.

[21] P. Jeavons, D. Cohen, and M.C. Cooper. Constraints, Consistency and Closure. *Artificial Intelligence*, 101:251–265, 1998.

[22] P. Jeavons, D. Cohen, and M. Gyssens. Closure Properties of Constraints. *Journal of the ACM*, 44(4):527–548, July 1997.

[23] L. Kirousis. Fast Parallel Constraint Satisfaction. *Artificial Intelligence*, 64:147–160, 1993.

[24] P. G. Kolaitis and M. Vardi. On the Expressive Power of Datalog: Tools and a Case Study. *Journal of Computer and System Sciences*, 51(1):110–134, 1995.

[25] P.G. Kolaitis and M. Vardi. A Game-Theoretic Approach to Constraint Satisfaction. In *17th National Conference on Artificial Intelligence (AAAI'00)*, pages 175–181.

[26] P.G. Kolaitis and M. Vardi. The Decision Problem for the Probabilities of Higher-Order Properties. In *19th Annual ACM Symposium on Theory of Computing*, pages 425–435, 1987.

[27] P.G. Kolaitis and M. Vardi. Conjunctive-Query Containment and Constraint Satisfaction. *Journal of Computer and System Sciences*, 61(2):302–332, 2000.

[28] A. K. Krokhin and B. Larose. Solving order constraints in logarithmic space. In *20th Annual Symposium on Theoretical Aspects of Computer Science (STACS'03)*, pages 379–390, 2003.

[29] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99–118, 1977.

[30] U. Montanari. Networks of Constraints: Fundamental Properties and Applications to Picture Processing. *Information Sciences*, 7:95–132, 1974.

[31] U. Montanari and F. Rossi. Constraint Relaxation may be Perfect. *Artificial Intelligence*, 48:143–170, 1991.

[32] C. H. Papadimitriou and M. Yannakakis. Optimization, Approximation, and Complexity Classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[33] N. Robertson and P. Seymour. Graph minors. i. excluding a forest. *J. Comb. Theory Series B*, 35:39–61, 1983.

[34] T.J. Schaefer. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, (STOC'78)*, pages 216–226, 1978.

[35] A. Szendrei. Idempotent algebras with restrictions in subalgebras. *Acta Sci. Math.*, 51:251–268, 1987.

[36] J.D. Ullman. *Principles of Database and Knowledge-Base Systems*, volume II. Computer Science Press, 1989.

[37] P. van Beek and R. Dechter. On the Minimality and Decomposability of Row-convex Constraint Networks. *Journal of the ACM*, 42:543–561, 1995.

[38] P. van Hentenryck, Y. Deville, and C-M. Teng. A Generic Arc-consistency Algorithm and its Specializations. *Artificial Intelligence*, 1992.