Two new homomorphism dualities and lattice operations[*]

Catarina Carvalho
University of Lisbon
1649-003 Lisbon, Portugal
ccarvalho@cii.fc.ul.pt

Víctor Dalmau
University Pompeu Fabra
Barcelona, 08018 Spain
victor.dalmau@upf.edu

Andrei Krokhin
Durham University
Durham, DH1 3LE, UK
andrei.krokhin@durham.ac.uk

June 17, 2010

Abstract

The study of constraint satisfaction problems definable in various fragments of Datalog has recently gained considerable importance. We consider constraint satisfaction problems that are definable in the smallest natural recursive fragment of Datalog - monadic linear Datalog with at most one EDB per rule, and also in the smallest non-linear extension of this fragment. We give combinatorial and algebraic characterisations of such problems, in terms of homomorphism dualities and lattice operations, respectively. We then apply our results to study graph **H**-colouring problems.

1 Introduction

The constraint satisfaction problem (CSP) provides a framework in which it is possible to express, in a natural way, many combinatorial problems encountered in artificial intelligence and computer science. 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 in a constraint satisfaction problem is then to find an assignment of values to the variables that satisfies the constraints.

It is well known (see, e.g., [7, 16, 27]) that the constraint satisfaction problem can be recast as the following fundamental problem: given two finite relational structures **A** and **B**, is there a homomorphism from **A** to **B**? The CSP is **NP**-complete in general, and the identifying of its subproblems that have lower complexity has been a very active research direction in the last decade (see, e.g., [7, 16, 18, 27, 29]). One of the most studied restrictions on the CSP is when the structure **B** is fixed, and only **A** is part of the input. The obtained problem is denoted by CSP(**B**). Examples of such problems include k-Sat, Graph H-colouring, and Systems of Equations (e.g., linear equations).

A variety of mathematical approaches to study problems CSP(**B**) has been recently suggested. The most advanced approaches use logic, combinatorics, universal algebra, and their

[*] A preliminary version of part of this paper was published in the proceedings of LICS'08.

combinations (see [6, 7, 26, 27]). The logic programming language Datalog and its fragments are arguably some of the most important tools for solving CSPs. In fact, all problems CSP($\mathbf{B}$) that are known to be tractable can be solved via Datalog, via the "few subpowers property" [24], or via a combination of the two. Furthermore, for every problem CSP($\mathbf{B}$) that is currently known to belong to **NL** or to **LOGSPACE**, the complement of CSP($\mathbf{B}$) can be defined in linear Datalog and symmetric Datalog, respectively (see [6, 9, 13]). The algebraic approach to the CSP was recently linked with non-definability in the above fragments of Datalog [29].

The definability of CSP($\mathbf{B}$) in Datalog and its fragments is very closely related with homomorphism dualities (see, e.g., [6]) that were much studied in the context of graph homomorphisms (see [20]). Roughly, a structure $\mathbf{B}$ has duality (of some type) if the non-existence of a homomorphism from a given structure $\mathbf{A}$ to $\mathbf{B}$ can always be explained by the existence of a simple enough obstruction structure (i.e., one that homomorphically maps to $\mathbf{A}$ but not to $\mathbf{B}$). The types of dualities correspond to interpretations of the phrase "simple enough". The most important duality probably is bounded treewidth duality which is equivalent to definability in Datalog (see [6, 16]). Structures with this duality have been recently characterised in algebraic terms (see [2], also [5]). Two other well-understood dualities are finite duality [28, 32, 33] and tree duality [11, 16]. Both of these properties have nice logical, combinatorial, and algebraic characterisations (see the above papers or [6]). For example, they correspond to definability in first-order logic and in monadic Datalog, respectively. The simplest of trees are paths, and the concept of path duality was also much used to study graph homomorphism (see [21, 23]). In this paper we argue that, in the setting of general relational structures, a slightly more general notion of *caterpillar* duality is more natural, and we give concise logical, combinatorial, and algebraic characterisations of structures having this form of duality (see Section 3). We also give such characterisations of a similar new duality (which we call *jellyfish* duality) that originates from the study of Boolean CSP [8]. Interestingly, and somewhat unexpectedly, the algebraic characterisations for the new dualities involve lattice operations.

The problems CSP($\mathbf{B}$) with $\mathbf{B}$ being a digraph $\mathbf{H}$ are actively studied in graph theory under the name of $\mathbf{H}$-colouring [20]. Recently, algebraic and logical approaches to the CSP were applied to solve well-known open problems (or to give short proofs of known results) about $\mathbf{H}$-colouring (see, e.g., [1, 3, 4]). We also apply our findings to obtain new results about $\mathbf{H}$-colouring in Section 4.

2 Preliminaries

2.1 Basic definitions

A *vocabulary* is a finite set of relation symbols or predicates. In what follows, τ always denotes a vocabulary. Every relation symbol R in τ has an *arity* $r = \rho(R) > 0$ 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. All structures in this paper are assumed to be *finite*, 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.

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 $(a_1, \ldots, a_r) \in R^{\mathbf{A}}$, we have $(h(a_1), \ldots, h(a_r)) \in R^{\mathbf{B}}$.

We denote this by $h : \mathbf{A} \to \mathbf{B}$. We also say that $\mathbf{A}$ homomorphically maps to $\mathbf{B}$, and write $\mathbf{A} \to \mathbf{B}$ if there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ and $\mathbf{A} \not\to \mathbf{B}$ if there is no homomorphism. Now $\mathrm{CSP}(\mathbf{B})$ can be defined to be the class of all structures $\mathbf{A}$ such that $\mathbf{A} \to \mathbf{B}$. The class of all structures $\mathbf{A}$ such that $\mathbf{A} \not\to \mathbf{B}$ will be denoted by co-$\mathrm{CSP}(\mathbf{B})$. We now give three examples of combinatorial problems representable as $\mathrm{CSP}(\mathbf{B})$ or co-$\mathrm{CSP}(\mathbf{B})$ for a suitable structure $\mathbf{B}$; a number of other examples can be found in [6, 7, 27].

Example 1. *If $\mathbf{B}_{hc}$ is a digraph $\mathbf{H}$ then $\mathrm{CSP}(\mathbf{B}_{hc})$ is the much-studied problem, $\mathbf{H}$-COLOURING, of deciding whether there is a homomorphism from a given digraph to $\mathbf{H}$ [20]. If $\mathbf{H}$ is the complete graph $\mathbf{K}_k$ on k vertices then it is well known (and easy to see) that $\mathrm{CSP}(\mathbf{B}_{hc})$ is precisely the standard k-COLOURING problem.*

Example 2. *If $\mathbf{B}_{lhc}$ is a structure obtained from a digraph $\mathbf{H}$ by adding, for each non-empty subset U of H, a unary relation U then $\mathrm{CSP}(\mathbf{B}_{lhc})$ is exactly the* LIST $\mathbf{H}$-COLOURING *problem, in which every vertex v of the input digraph $\mathbf{G}$ gets a list L_v of vertices of $\mathbf{H}$, and the question is whether there is a homomorphism $h : \mathbf{G} \to \mathbf{H}$ such that $h(v) \in L_v$ for all $v \in G$ (see [20]).*

Example 3. *If $\mathbf{B}_r$ is the Boolean (i.e., with universe $\{0, 1\}$) structure with one binary relation $R_{\leq}^{\mathbf{B}_r}$, which is the natural order relation on $\{0, 1\}$, and two unary relations $T^{\mathbf{B}_r} = \{0\}$ and $S^{\mathbf{B}_r} = \{1\}$ then* co-$\mathrm{CSP}(\mathbf{B}_r)$ *is the (directed)* REACHABILITY *problem where one is given a digraph and two sets of vertices in it, S and T, and the question is whether there is a directed path in the graph from some vertex in S to a vertex in T.*

Two structures $\mathbf{B}_1$ and $\mathbf{B}_2$ are said to be *homomorphically equivalent* if both $\mathbf{B}_1 \to \mathbf{B}_2$ and $\mathbf{B}_2 \to \mathbf{B}_1$. Clearly, in this case we have $\mathrm{CSP}(\mathbf{B}_1) = \mathrm{CSP}(\mathbf{B}_2)$. A *retract* of a structure $\mathbf{B}$ is an induced substructure $\mathbf{B}'$ of $\mathbf{B}$ such that there is a homomorphism $h : \mathbf{B} \to \mathbf{B}'$ satisfying $h(b) = b$ for all $b \in \mathbf{B}'$. A structure is a *core* if it has no proper retracts, and a *core of a structure* is its retract that is a core. It is well known that all cores of a structure are isomorphic, so we will call any structure isomorphic to a core of $\mathbf{B}$ *the core* (of $\mathbf{B}$), denoted $\mathrm{core}(\mathbf{B})$. Note that two structures are homomorphically equivalent if and only if they have the same core (up to isomorphism).

We will now define structures that play an important role in this paper - trees and caterpillars, which are natural generalisations of the corresponding notions from graph theory, and also jellyfish structures which are new. Let $\mathbf{A}$ be a τ-structure. As in [32], the *incidence multigraph* of $\mathbf{A}$, denoted $Inc(\mathbf{A})$, is defined as the bipartite multigraph with parts A and $Block(\mathbf{A})$, where $Block(\mathbf{A})$ consists of all pairs $(R, \overline{a})$ such that $R \in \tau$ and $\overline{a} \in R^{\mathbf{A}}$, and with edges $e_{a,i,Z}$ joining $a \in A$ to $Z = (R, (a_1, \ldots, a_r)) \in Block(\mathbf{A})$ when $a_i = a$. A structure $\mathbf{A}$ is said to be a *τ-tree* (or simply a *tree*) if its incidence multigraph is a tree (in particular, it has no multiple edges). For a τ-tree $\mathbf{A}$, we say that an element of A is a *leaf* if it is incident to at most one block in $Inc(\mathbf{A})$. A block of $\mathbf{A}$ (i.e., a member of $Block(\mathbf{A})$) is said to be *pendant* if it is incident to at most one non-leaf element, and it is said to be *non-pendant* otherwise. For example, any block with a unary relation is always pendant. Observe that a structure with only one element and empty relations is a tree.

In graph theory, a caterpillar is a tree which becomes a path after all its leaves are removed. Following [30], we say that a τ-tree is a *τ-caterpillar* (or simply a caterpillar) if each of its blocks is incident to at most two non-leaf elements, and each element is incident to at most two non-pendant blocks. Informally, a τ-caterpillar has a body consisting of a chain of elements $a_1, \ldots, a_{n+1}$ with blocks $B_1, \ldots, B_n$ where B_i is incident to a_i and a_{i+1} ($i = 1, \ldots, n$), and legs of two types: (i) pendant blocks incident to exactly one of the elements $a_1, \dots, a_{n+1}$, together with some leaf elements incident to such blocks, and (ii) leaf elements incident to exactly one of the blocks $B_1, \dots, B_n$.

Example 4. *(i) If τ is the signature of digraphs then the τ-caterpillars are the* oriented caterpillars*, i.e., digraphs obtained from caterpillar graphs by orienting each edge in some way.*

(ii) Let $\mathbf{B}$ be a structure with $B = \{1, 2, \dots, 9\}$, one unary relation $R_1 = \{2, 3\}$, one binary relation $R_2 = \{(1, 2), (2, 3), (3, 4), (4, 8)\}$ and one ternary relation $R_3 = \{(3, 6, 7), (4, 5, 9)\}$. The graph $Inc(\mathbf{B})$ is shown on Fig. 1. The elements 2, 3, 4 are the non-leaves, and $(R_2, (2, 3))$ and $(R_2, (3, 4))$ are the non-pendant blocks. In particular, $\mathbf{B}$ is a caterpillar.

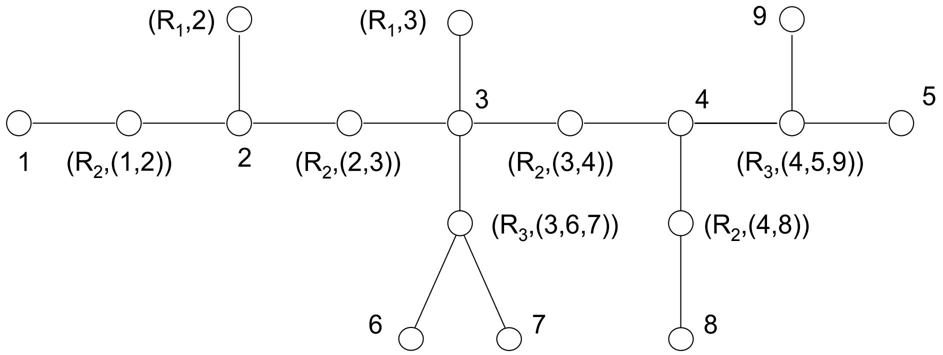


Figure 1: The incidence graph of a caterpillar structure.

We say that a non-leaf $a \in A$ of a τ-tree $\mathbf{A}$ is *extreme* if it is incident to at most one non-pendant block (i.e., it has at most one other non-leaf at distance two from it) in $Inc(\mathbf{A})$, and we say that a pendant block is extreme if either it is the only block of $\mathbf{A}$ or else it is adjacent to a non-leaf, and this (unique) non-leaf is extreme. Finally, we say that an element is *terminal* if it is isolated (i.e., does not appear in any relation in $\mathbf{A}$) or it appears in an extreme pendant block. Clearly, a caterpillar can have at most two extreme non-leaves and every extreme non-leaf is also terminal. For example in the caterpillar from Fig. 1, the extreme non-leaves are 2 and 4, the extreme pendant blocks are $R_2(1, 2)$, $R_1(2)$, $R_2(4, 8)$, and $R_3(4, 5, 9)$, and the terminal elements are 1, 2, 4, 5, 8, 9.

The following definition is new. We say that a τ-tree $\mathbf{A}$ is a *τ-jellyfish* if it is a one-element structure with empty relations or it is obtained from one tuple (in one relation) $R(\overline{a})$, called the body of the jellyfish, and a family of caterpillars by identifying one terminal element of each caterpillar with some element in the tuple $\overline{a}$ (see Fig. 2, where the curved lines depict caterpillars). Note that, since $\mathbf{A}$ is a tree, the tuple $\overline{a}$ is contained in only one relation in $\mathbf{A}$, and has no repeated components. It is easy to see that each caterpillar structure is a jellyfish structure.

2.2 Datalog

We now briefly describe the basics of Datalog (for more details, see, e.g., [9, 12, 25, 26]). Fix a vocabulary τ. A *Datalog program* is a finite set of rules of the form $t_0 :- t_1, \dots, t_n$

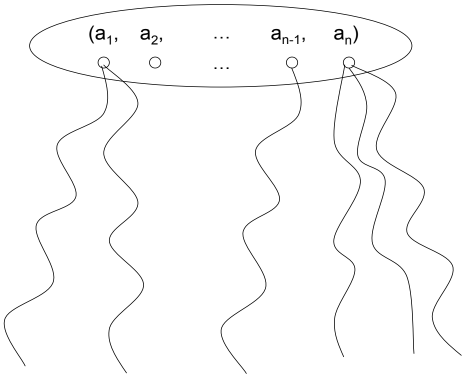


Figure 2: A jellyfish structure, schematically.

where each t_i is an atomic formula $R(x_{i_1}, \ldots, x_{i_k})$. Then t_0 is called the *head* of the rule, and the sequence $t_1, \ldots, t_n$ the *body* of the rule. The intended meaning of such a rule is that the conjunction of the atomic formulas in the body implies the formula in the head, with all variables not appearing in the head existentially quantified. The predicates occurring in the heads of the rules are not from τ and are called *IDBs* (from "intensional database predicates"), while all other predicates come from τ and are called *EDBs* (from "extensional database predicates"). One of the IDBs, which is usually 0-ary in our case, is designated as the *goal predicate* of the program. Since the IDBs may occur in the bodies of rules, each Datalog program is a recursive specification of the IDBs, with semantics obtained via least fixed-points of monotone operators. The goal predicate is assumed to be initially set to `false`, and we say that a Datalog program *accepts* a τ-structure $\mathbf{A}$ if its goal predicate evaluates to `true` on $\mathbf{A}$. In this case we also say that the program *derives* the goal predicate on $\mathbf{A}$. More generally, we say that a program *derives the fact* $I(\overline{a})$ on $\mathbf{A}$, where I is an IDB in the program and $\overline{a}$ is a tuple, if $I(\overline{a})$ holds after the program's run on $\mathbf{A}$. It is easy to see that the class of structures accepted by any Datalog program is closed under homomorphism (i.e., if $\mathbf{A} \to \mathbf{B}$ and $\mathbf{A}$ is accepted then $\mathbf{B}$ is also accepted). Hence, when using Datalog to study CSP($\mathbf{B}$), one speaks of the definability of co-CSP($\mathbf{B}$) in Datalog (or its fragments).

A rule of a Datalog program is called *linear* if it has at most one occurrence of an IDB in its body. A Datalog program is called *linear* if each of its rules is linear, *monadic* if each IDB in it is at most unary, and a $(1, k)$-Datalog program if it is monadic and every rule in it uses at most k variables. A Datalog program is *recursion-free* if the goal predicate is the only IDB in it.

We now give some examples of Datalog programs defining classes of the form co-CSP($\mathbf{B}$), more examples can be found in [6, 9, 26].

Example 5. *(i) Recall the problem* CSP($\mathbf{B}_r$) *from Example 3. It is easy to check that the following (linear monadic) program defines* co-CSP($\mathbf{B}_r$). *(It recursively computes the unary relation O containing all vertices reachable from S).*

$$\begin{array}{lll} O(X) & :- & S(X) \\ O(Y) & :- & R_{\leq}(X,Y), S(X) \\ G & :- & T(X), O(X) \end{array}$$

(ii) Let $\mathbf{B}_{ihsb-k}$ denote the Boolean structure with three relations, unary $U = \{0\}$, binary $R_{\leq}$ (same as in the previous example), and k-ary $W = \{0,1\}^k \setminus \{(0,\dots,0)\}$. These relations are basic implicative hitting-set bounded *relations, as introduced in [8]. It can be checked directly that the following program describes* co-CSP$(\mathbf{B}_{ihsb-k})$.

$$\begin{array}{lll} Z(X) & :- & U(X) \\ Z(X) & :- & R_{\leq}(X,Y), Z(Y) \\ G & :- & W(X_1,X_2,\dots,X_k), Z(X_1), Z(X_2),\dots,Z(X_k) \end{array}$$

For a given structure $\mathbf{B}$, a given fragment of Datalog, and a given (upper) bound on the number of variables in a rule, there is a standard way of constructing the *canonical program* for $\mathbf{B}$ in the given fragment of Datalog with the given bound (see, e.g., [6, 16]). We will need this construction only for fragments of monadic Datalog, so we describe it only for this case. The canonical $(1,k)$-Datalog program $\mathfrak{P}_{1,\mathfrak{k}}$ for a structure $\mathbf{B}$ is constructed as follows: let $S_0, S_1, \dots, S_p$ be an enumeration of unary relations on B (i.e., subsets of B) that can be expressed by a first-order $\exists\wedge$-formula over $\mathbf{B}$. Assume that S_0 is the empty relation. For each S_i, introduce a unary IDB I_i. Then the canonical program involves the IDBs $I_0,\dots,I_p$ and EDBs $R_1,\dots,R_n$ (precisely corresponding to the relation symbols in τ) , and contains all the rules with at most k variables that have the following property: if every I_i in the rule is replaced by S_i and every R_s by $R_s^{\mathbf{B}}$, then every assignment of elements of B to the variables that satisfies the conjunction of atomic formulas in the body must also satisfy the atomic formula in the head. Finally, declare I_0 to be the goal predicate (or equivalently include the goal predicate G along with the rule $G :- I_0(x)$). To obtain a canonical program for $\mathbf{B}$ in a given fragment of monadic Datalog, simply remove from $\mathfrak{P}_{1,\mathfrak{k}}$ all rules not belonging to the fragment.

Note that $\mathbf{B}$ is not accepted by the canonical program for itself (in any fragment of Datalog, for any k). Indeed, by construction, a derivation of G on $\mathbf{B}$ could be translated into a chain of valid implications which starts from an atomic formula and finishes with the empty (i.e., false) predicate, which is impossible. This, and the fact that any class definable in Datalog is closed under homomorphism, implies the following.

Fact 6. *If the canonical program for* $\mathbf{B}$ *in some fragment of Datalog accepts a structure* $\mathbf{A}$ *then* $\mathbf{A} \not\to \mathbf{B}$.

2.3 Polymorphisms

Let f be an n-ary operation on B, and R a relation on B. We say that f is a *polymorphism* of R if, for any tuples, $\bar{a}_1,\dots,\bar{a}_n \in R$, the tuple obtained by applying f componentwise to $\bar{a}_1,\dots,\bar{a}_n$ also belongs to R. In this case we also say that R is invariant under f.

Example 7. *It is straightforward to verify that, for the Boolean relation $OR = \{0,1\}^2 \setminus \{(0,0)\}$, the binary operation* max *on* $\{0,1\}$ *is a polymorphism, but the operation* min *is not.*

We say that f is a *polymorphism of* $\mathbf{B}$ if it is a polymorphism of each relation in $\mathbf{B}$. It is easy to check that the n-ary polymorphisms of $\mathbf{B}$ are precisely the homomorphisms from the n-th direct power $\mathbf{B}^n$ to $\mathbf{B}$. It is well known and easy to verify that composition of polymorphisms of $\mathbf{B}$ is again a polymorphism of $\mathbf{B}$ (see, e.g., [7]).

The notion of a polymorphism plays the key role in the algebraic approach to the CSP. The polymorphisms of a (core) structure are known to determine the complexity of CSP$(\mathbf{B})$

as well as definability of (the complement of) CSP($\mathbf{B}$) in Datalog and the following fragments: monadic, linear, symmetric (see [6, 29]). Many algebraic sufficient conditions for definability of co-CSP($\mathbf{B}$) in various fragments of Datalog are known (see [6]).

Let us now define several types of operations that will be used in this paper.

- An n-ary operation f on B is called *idempotent* if it satisfies the identity $f(x, \ldots, x) = x$, and it is called *conservative* if $f(x_1, \ldots, x_n) \in \{x_1, \ldots, x_n\}$ for all $x_1, \ldots, x_n \in B$.
- An n-ary operation f is called *totally symmetric* if $f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$ whenever $\{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\}$.
- An n-ary ($n \geq 3$) operation is called an *NU (near-unanimity)* operation if it satisfies the identities

$$f(y, x, \ldots, x, x) = f(x, y, \ldots, x, x) = \ldots = f(x, x, \ldots, x, y) = x.$$

- A ternary NU operation is called a *majority* operation.
- A binary associative commutative idempotent operation is called a *semilattice* operation.
- A pair of binary operations f, g on B is a pair of *lattice* operations if each of them is a semilattice operation and, in addition, they satisfy the absorption identities: $f(x, g(x, y)) = g(x, f(x, y)) = x$.

It is well known (see, e.g., [17]) that semilattice operations are in one-to-one correspondence with partial orders in which every two elements have a greatest lower bound (which is the result of applying the operation). Similarly, lattice operations are in one-to-one correspondence with partial orders in which every two elements have both a greatest lower bound and a least upper bound. The simplest example of lattice operations are the operations min and max with respect to any fixed linear order on B. It is standard practice to use infix notation for lattice operations, i.e., to write $x \sqcap y$ and $x \sqcup y$ for $f(x, y)$ and $g(x, y)$, respectively. The algebra $(B, \sqcap, \sqcup)$ is then called a lattice. A lattice is said to be *distributive* if it satisfies the identity $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$. Equivalently, a lattice is distributive if it can be represented by a family of subsets of a set with the operations interpreted as set-theoretic intersection and union (see [17]).

2.4 Dualities

A comprehensive treatment of dualities for the CSP can be found in the survey [6].

Definition 8. *A set $\mathcal{O}$ of τ-structures is called an* obstruction set *for* $\mathbf{B}$ *if, for any τ-structure* $\mathbf{A}$*, we have* $\mathbf{A} \not\to \mathbf{B}$ *if and only if* $\mathbf{A}' \to \mathbf{A}$ *for some* $\mathbf{A}' \in \mathcal{O}$.

If the set $\mathcal{O}$ can be chosen to consist of nicely behaved structures such as paths, caterpillars, trees, or structures of bounded pathwidth or of bounded treewidth, then $\mathbf{B}$ is said to have path (caterpillar, tree, bounded pathwidth, bounded treewidth, respectively) duality. The notions of bounded path- and treewidth are not defined here because they are not needed in this paper (but they can be found in [6]). A structure with a finite obstruction set is said to have finite duality.

Example 9. *Let $\mathbf{T}_n$ be the (irreflexive)* transitive tournament *on n vertices, that is, the universe of $\mathbf{T}_n$ is $\{0, 1, \ldots, n-1\}$, and the only relation is the binary relation $\{(i, j) \mid 0 \leq i < j \leq n-1\}$. Also, let $\mathbf{P}_{n+1}$ be the* directed path *on $n+1$ vertices, that is the structure with universe $\{0, 1, \ldots, n\}$ and the relation $\{(i, i+1) \mid 0 \leq i \leq n-1\}$. It is well known (see, e.g., Proposition 1.20 of [20]) and easy to show that, for any digraph $\mathbf{G}$, $\mathbf{G} \to \mathbf{T}_n$ if and only if $\mathbf{P}_{n+1} \not\to \mathbf{G}$. Hence, $\{\mathbf{P}_{n+1}\}$ is an obstruction set for $\mathbf{T}_n$, and $\mathbf{T}_n$ has finite (path) duality.*

Example 10. *An* oriented path *is a digraph obtained from a path by orienting its edges in some way. A digraph is called a* local tournament *if the set of out-neighbours of any vertex induces a tournament. For example, all transitive tournaments and all directed paths (see Example 9) are local tournaments. It was shown in [22, 23] that any digraph $\mathbf{H}$ that is an oriented path or an acyclic local tournament has an obstruction set consisting of oriented paths. Since any oriented path is a caterpillar, it follows that $\mathbf{H}$ has caterpillar duality (and even path duality). It can be shown that it does not necessarily have finite duality.*

Example 11. *It can be shown that the structure $\mathbf{B}_{ihsb-k}$ from Example 5(ii) has jellyfish duality. Any such structure has polymorphism $x \vee (y \wedge z)$. Moreover, it can be shown (see Section 7.2 of [9]) that any Boolean structure that has such polymorphism, or the dual polymorphism $x \wedge (y \vee z)$, has jellyfish duality. In fact, it was the observation that jellyfish duality and these two polymorphisms are connnected (in the Boolean case) that led us to introducing and studying jellyfish dualities in general.*

Example 12. *Let $\mathbf{B}$ have universe $B = \{1, 2, \ldots, m\}$, one binary relation $R_{\leq}$ (natural order on B), and m unary relations $U_a = \{a\}$, $1 \leq a \leq m$. For fixed $1 \leq i < j \leq m$, let $\mathbf{B}_{ij}$ be the structure obtained from $\mathbf{B}$ by removing the tuple (i, j) from $R_{\leq}$. Structures in the signature of $\mathbf{B}$ can be naturally viewed as coloured digraphs (with colours given by the unary relations).*

Then it is not hard to check that $\mathbf{B}_{ij}$ has an obstruction set consisting of all structures of the form shown on Fig. 3(a) (where the directed path has any non-negative length and the black circles denote coloured elements) with $p > q$ and, in addition, the following structures. (1) If $i+1 = j$, all structures of the form shown on Fig. 3(a) with $p = i$ and $q = i+1$, so $\mathbf{B}_{ij}$ has caterpillar duality. (2) If $1 < i < m-1$ and $j = m$, all structures of the form shown on Fig. 3(b). (3) If $i = 1$ and $2 < j < m$, all structures of the form shown on Fig. 3(c). (4) If $1 < i < j-1 < m-1$, all structures of the form shown on Fig. 3(d). Note that the structures shown on Fig. 3(b-d) are jellyfish structures (the horizontal arc being the body), so $\mathbf{B}_{ij}$ has jellyfish duality in this case (but not caterpillar duality, as we will show in Section 3.3). The remaining case $i = 1, j = m$ is a cross between the second and the third cases above, and $\mathbf{B}_{1m}$ can be easily shown to have caterpillar duality.

It is known (see [6]) that a structure $\mathbf{B}$ has one of the following forms of duality: finite, tree, bounded pathwidth, bounded treewidth if and only if co-CSP($\mathbf{B}$) is definable in the following fragments of Datalog, respectively: recursion-free, monadic, linear, full.

Structures with tree duality were characterised in several equivalent ways in [16]. To state the result, we need the following construction: for a τ-structure $\mathbf{B}$, define a τ-structure $\mathbf{U}(\mathbf{B})$ whose elements are the non-empty subsets of B, and, for each r-ary $R \in \tau$, we have $(A_1, \ldots, A_r) \in R^{\mathbf{U}(\mathbf{B})}$ if and only if, for each $j = 1, \ldots, r$ and each $a \in A_j$, there exists $(a_1, \ldots, a_r) \in R^{\mathbf{B}} \cap (A_1 \times \ldots \times A_r)$ such that $a_j = a$.

Theorem 13. *[16, 11] Let $\mathbf{B}$ be a structure. The following conditions are equivalent:*

1. *$\mathbf{B}$ has tree duality;*

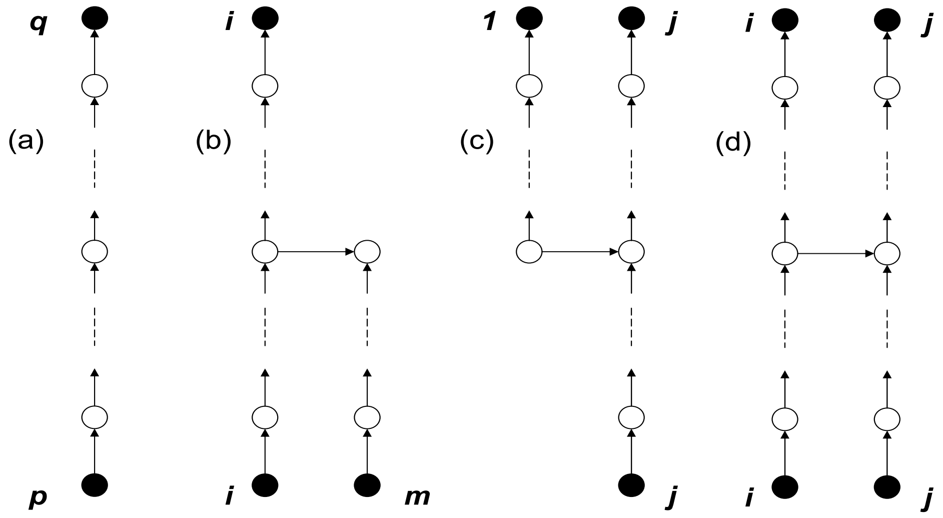


Figure 3: Obstructions for $\mathbf{B}_{ij}$.

2. co-CSP($\mathbf{B}$) *is definable by a monadic Datalog program with at most one EDB per rule;*

3. $\mathbf{U}(\mathbf{B})$ *admits a homomorphism to* $\mathbf{B}$*;*

4. *for every* $n \geq 1$*,* $\mathbf{B}$ *has an* n*-ary totally symmetric polymorphism;*

5. $\mathbf{B}$ *is homomorphically equivalent to a structure* $\mathbf{A}$ *with polymorphism* $x \sqcap y$ *for some distributive lattice* $(A, \sqcap, \sqcup)$*;*

6. $\mathbf{B}$ *is homomorphically equivalent to a structure* $\mathbf{A}'$ *with polymorphism* $x \sqcap y$ *for some semilattice* $(A', \sqcap)$*.*

It is known that any structure with finite duality has a finite obstruction set consisting of trees [32]; such structures are characterised in many equivalent ways in [28]. The situation when these trees can be chosen to be caterpillars was considered in [30].

Theorem 14 ([30])**.** *Let* $\mathbf{B}$ *be a core structure with finite duality. Then* $\mathbf{B}$ *has an obstruction set consisting of caterpillars if and only if* $\mathbf{B}$ *has a majority polymorphism.*

3 The new dualities

3.1 A characterisation

The main results of this paper are characterisations of structures with caterpillar duality and with jellyfish duality in the spirit of Theorem 13. First, we need to give some definitions.

Let k, n be positive integers. We call a (kn)-ary operation f on B *k-block symmetric* if it satisfies the following condition:

$$f(x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk}) = f(y_{11}, \ldots, y_{1k}, \ldots, y_{n1}, \ldots, y_{nk})$$

whenever $\{S_1, \ldots, S_n\} = \{T_1, \ldots, T_n\}$ where, for all i, $S_i = \{x_{i1}, \ldots, x_{ik}\}$ and $T_i = \{y_{i1}, \ldots, y_{ik}\}$. Note that if $k = 1$ or $n = 1$ then f is totally symmetric.

We will often use the following notation for k-block symmetric operations. For (not necessarily distinct) subsets $S_1, \ldots, S_n$ of B, with at most k elements each, let $f(S_1, S_2, \ldots, S_n)$ denote $f(x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk})$ where $S_i = \{x_{i1}, \ldots, x_{ik}\}$ for all i. Also, for $l \leq n$, let $f(S_1, \ldots, S_l)$ denote $f(S_1, \ldots, S_l, \ldots, S_l)$. It is clear that $f(S_1, \ldots, S_l)$ is well defined and depends neither on the order of the sets S_i nor on the number of repetitions of those sets. Therefore, we will also write $f(\mathcal{S})$ for a family of non-empty subsets $\mathcal{S} = \{S_1, \ldots, S_l\}$ to denote $f(S_1, \ldots, S_l)$.

If a k-block symmetric operation f satisfies $f(S_1, S_2, S_3, \ldots, S_l) = f(S_2, S_2, S_3, \ldots, S_l)$ whenever $S_2 \subseteq S_1$, we call it an *absorptive* k-block symmetric operation (or *k-ABS* operation, for short). The most typical example of such an operation is as follows.

Example 15. *(i) It is easy to check that, for any fixed linear order on B and any positive integers k, n, the operation*

$$f(x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk}) = \max(\min(x_{11}, \ldots, x_{1k}), \ldots, \min(x_{n1}, \ldots, x_{nk})).$$

is a k-ABS operation.

(ii) More generally, if $(B, \sqcap, \sqcup)$ is a lattice then it is easy to check that, for any k, n, the operation

$$f(x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk}) = (x_{11} \sqcap \ldots \sqcap x_{1k}) \sqcup \ldots \sqcup (x_{n1} \sqcap \ldots \sqcap x_{nk}).$$

is a k-ABS operation.

For $1 \leq m \leq r$ and an r-ary relation R, let $\mathrm{pr}_m(R) = \{a_m \mid (a_1, \ldots, a_m, \ldots, a_r) \in R\}$.

Let $\mathbf{B}$ be a τ-structure. We construct a τ-structure $\mathbf{C}(\mathbf{B})$ as follows: the elements of $\mathbf{C}(\mathbf{B})$ are the families of non-empty subsets of B; for each r-ary relation $R^{\mathbf{B}}$, we have $(\mathcal{S}^1, \mathcal{S}^2, \ldots, \mathcal{S}^r) \in R^{\mathbf{C}(\mathbf{B})}$ if, for all $j, m = 1, \ldots, r$, we have

1. $\mathrm{pr}_m(R^{\mathbf{B}}) \in \mathcal{S}^m$, and
2. $\mathrm{pr}_m(R^{\mathbf{B}} \cap (B^{j-1} \times S \times B^{r-j})) \in \mathcal{S}^m$ for each $S \in \mathcal{S}^j$.

Note that the empty family belongs to the universe of $\mathbf{C}(\mathbf{B})$, but it never appears in any tuple in a relation in this structure.

Theorem 16. *Let $\mathbf{B}$ be a structure. The following conditions are equivalent:*

1. $\mathbf{B}$ *has caterpillar duality;*
2. $\text{co-CSP}(\mathbf{B})$ *is definable by a linear monadic Datalog program with at most one EDB per rule;*
3. $\mathbf{C}(\mathbf{B})$ *admits a homomorphism to* $\mathbf{B}$*;*
4. *for every $k, n \geq 1$,* $\mathbf{B}$ *has a kn-ary k-ABS polymorphism;*
5. $\mathbf{B}$ *is homomorphically equivalent to a structure* $\mathbf{A}$ *with polymorphisms $x \sqcap y$ and $x \sqcup y$ for some distributive lattice $(A, \sqcap, \sqcup)$;*

6. $\mathbf{B}$ *is homomorphically equivalent to a structure* $\mathbf{A}'$ *with polymorphisms* $x \sqcap y$ *and* $x \sqcup y$ *for some lattice* $(A', \sqcap, \sqcup)$.

The proof of Theorem 16 will be given in Section 3.2.

We say that an operation f of arity $kn+1$ is an *extended k-ABS* operation if, (i) by fixing any value for the first variable one gets a kn-ary k-ABS operation, and, in addition, (ii) the following *replacement property* holds for all x, y and all sets $S_1, \ldots, S_n$ with $|S_i \cup \{y\}| \leq k$ and $|S_i \cup \{x\}| \leq k$ for $i = 1, \ldots, n$:

$$f(x, S_1 \cup \{y\}, \ldots, S_n \cup \{y\}) = f(y, S_1 \cup \{x\}, \ldots, S_n \cup \{x\}).$$

Note that property (i) allows us to write property (ii) in this form, by using the "block notation".

Example 17. *If* $(B, \sqcap, \sqcup)$ *is a distributive lattice then it is easy to check that, for any* k, n, *the operation*

$$f(x_0, x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk}) = x_0 \sqcap ((x_{11} \sqcap \ldots \sqcap x_{1k}) \sqcup \ldots \sqcup (x_{n1} \sqcap \ldots \sqcap x_{nk})).$$

is an extended k-ABS operation. The distributivity of the lattice is important to guarantee the replacement property.

For a structure $\mathbf{B}$, let $\mathbf{J}(\mathbf{B})$ denote the substructure of $\mathbf{C}(\mathbf{B})$ such that, for any R, a tuple $(\mathcal{S}^1, \mathcal{S}^2, \ldots, \mathcal{S}^r) \in R^{\mathbf{C}(\mathbf{B})}$ belongs to $R^{\mathbf{J}(\mathbf{B})}$ if and only if $R^{\mathbf{B}} \cap (\bigcap \mathcal{S}^1 \times \bigcap \mathcal{S}^2 \times \ldots \times \bigcap \mathcal{S}^r) \neq \emptyset$.

Theorem 18. *Let* $\mathbf{B}$ *be a structure. The following conditions are equivalent:*

1. $\mathbf{B}$ *has jellyfish duality;*
2. co-CSP($\mathbf{B}$) *is definable by a monadic Datalog program with at most one EDB per rule and such that each non-linear rule has the goal predicate in its head;*
3. $\mathbf{J}(\mathbf{B})$ *admits a homomorphism to* $\mathbf{B}$*;*
4. *for every* $k, n \geq 1$, $\mathbf{B}$ *has a* $(kn+1)$*-ary extended k-ABS polymorphism;*
5. $\mathbf{B}$ *is homomorphically equivalent to a structure* $\mathbf{A}$ *with polymorphism* $x \sqcap (y \sqcup z)$ *for some distributive lattice* $(A, \sqcap, \sqcup)$.

The proofs of Theorems 16 and 18 are very similar, and so we will prove the theorems simultaneously in the next subsection.

For the sake of brevity, we will say that a monadic Datalog program with at most one EDB per rule (as in condition (2) of Theorem 13) is a *tree program*. Similarly, we will say that a tree program which is linear (as in condition (2) of Theorem 16) is a *caterpillar program*, and a program such as the one described in condition (2) of Theorem 18 is a *jellyfish program*. For instance, the program from Example 5(i) is a caterpillar program, while the program from Example 5(ii) is a jellyfish program.

It is not hard to verify that, for any class co-CSP($\mathbf{B}$) definable by a (non-linear, in general) jellyfish program, one can construct a linear, though not necessarily monadic, Datalog program that also defines co-CSP($\mathbf{B}$). In particular, it follows, by [9], that jellyfish duality for $\mathbf{B}$ implies membership of CSP($\mathbf{B}$) in the complexity class **NL**. Previously, the presence of a majority polymorphism was the only general (i.e., applicable to any structure) sufficient algebraic condition for CSP($\mathbf{B}$) to be in **NL** [10]. The above discussion shows that $(kn+1)$-ary extended k-ABS polymorphism (where, by Remark 28, we can take $k = \rho|B|$ and $n = \rho(2^{|B|}-1)$ with ρ being the maximum of the arities of the relations in $\mathbf{B}$) provides a new such condition.

3.2 Proofs of Theorems 16 and 18

We will prove the theorems through a sequence of lemmas. In our proofs, we will actively use the canonical caterpillar (and jellyfish) program for $\mathbf{B}$, recall the definitions from Section 2.2. We will always assume that the bound on the number of variables in a rule in such a program is equal to the maximum of the arities of the relations in $\mathbf{B}$.

Lemma 19. *Let $\mathbf{D}$ be a caterpillar and let a be a terminal element in it. Then the canonical caterpillar program for $\mathbf{B}$ derives, on $\mathbf{D}$, the fact $I_j(a)$ where I_j is the IDB corresponding to the set $S_j = \{h(a) \mid h : \mathbf{D} \to \mathbf{B}\}$.*

Proof. We prove the lemma by induction on the number of nodes of the incidence graph, $Inc(\mathbf{D})$, of $\mathbf{D}$. If $\mathbf{D}$ consists of a unique node a and empty relations then $S_j = B$ and the program derives $I_j(a)$ via the rule $I_j(X) : -$(empty body).

Assume now that $Inc(\mathbf{D})$ has at least two nodes. Let $R(a_1, \dots, a_r)$ be the extreme pendant block to which a belongs, so $a = a_s$ for some $1 \leq s \leq r$. If $\mathbf{D}$ contains some other block then define a_i to be the extreme non-leaf in $\{a_1, \dots, a_r\}$; otherwise, let a_i be an arbitrary element of $\{a_1, \dots, a_r\}$. Let $\mathbf{D}'$ be the substructure of $\mathbf{D}$ obtained by removing the tuple $(a_1, \dots, a_r)$ from R and the elements of $\{a_1, \dots, a_r\} \setminus \{a_i\}$ from D (observe that in the second case this gives a structure containing only the element a_i and empty relations). Clearly a_i is a terminal element of $\mathbf{D}'$ and hence, by the inductive hypothesis, the canonical caterpillar program derives on $\mathbf{D}'$ (and hence on $\mathbf{D}$) the fact $I_t(a_i)$ where $S_t = \{h(a_i) \mid h : \mathbf{D}' \to \mathbf{B}\}$. Clearly $S_j = \mathrm{pr}_s\, R^{\mathbf{B}} \cap (B^{i-1} \times S_t \times B^{r-i})$ and hence the canonical caterpillar program contains the rule

$$I_j(X_s) : -R(X_1, \dots, X_r), I_t(X_i)$$

from which $I_j(a)$ is obtained. ■

Lemma 20. *For any structures $\mathbf{A}$ and $\mathbf{B}$, if there exists a caterpillar (jellyfish) $\mathbf{C}$ such that $\mathbf{C} \to \mathbf{A}$ and $\mathbf{C} \not\to \mathbf{B}$ then the canonical caterpillar (jellyfish, resp.) program for $\mathbf{B}$ accepts $\mathbf{A}$.*

Proof. Since the class of structures accepted by any Datalog program is closed under homomorphism, it suffices to show that $\mathbf{C}$ is accepted by the program.

The caterpillar case follows directly from Lemma 19. Let us now consider the jellyfish case. Let the tuple $(a_1, \dots, a_r) \in R$ be the body of the jellyfish $\mathbf{C}$ and, for $1 \leq i \leq r$, let $\mathbf{C}_{i1}, \dots, \mathbf{C}_{is_i}$ be the caterpillars attached to a_i. Since the canonical jellyfish program for $\mathbf{B}$ contains all rules of the canonical caterpillar one, Lemma 19 implies that, for each $1 \leq i \leq r$ and each $1 \leq j \leq s_i$, the canonical jellyfish program derives the fact $I_{ij}(a_i)$ where I_{ij} is the IDB corresponding to the set $S_{ij} = \{h(a_i) \mid h : \mathbf{C}_{ij} \to \mathbf{B}\}$. Since $\mathbf{C} \not\to \mathbf{B}$, it follows that $\{(b_1, \dots, b_r) \in R^{\mathbf{B}} \mid b_i \in S_{ij}$ for all $i = 1, \dots, r$ and all $j = 1, \dots, s_i\} = \emptyset$ and hence, by construction, the canonical jellyfish program contains the rule $G : -R(X_1, \dots, X_r), I_{11}(X_1), , \dots, I_{1s_1}(X_1), \dots, I_{r1}(X_r), \dots, I_{rs_r}(X_r)$ from which the goal predicate is derived. ■

Lemma 21. *A structure* $\mathbf{B}$ *has caterpillar (jellyfish) duality if and only if* co-CSP($\mathbf{B}$) *can be defined by a caterpillar (jellyfish, respectively) program.*

Proof. Suppose that co-CSP($\mathbf{B}$) is defined by a caterpillar program. This means that a structure $\mathbf{A}$ satisfies $\mathbf{A} \not\to \mathbf{B}$ if and only if $\mathbf{A}$ is accepted by the program.

If $\mathbf{A} \not\to \mathbf{B}$ then by the Sparse Incomparability Lemma [31] there is a structure $\mathbf{C}$ that is homomorphic to $\mathbf{A}$ but not to $\mathbf{B}$ and such that for every $R \in \tau$ and every $(a_1, \ldots, a_r) \in R^{\mathbf{C}}$, $a_i \neq a_j$ for all $1 \leq i \neq j \leq r$. Since $\mathbf{C} \not\to \mathbf{B}$ the canonical program derives the goal predicate on $\mathbf{C}$. Reading the derivation from the end to the beginning we obtain

$$\begin{array}{rl} G :- & I_0(a_0) \\ I_0(a_0) :- & R_{i_1}(\ldots, a_0, \ldots, a_i, \ldots), I_{j_1}(a_i) \\ I_{j_1}(a_i) :- & R_{i_2}(\ldots, a_i, \ldots, a_j, \ldots), I_{j_2}(a_j) \\ & \vdots \\ I_{j_{(l-1)}}(a_p) :- & R_{i_l}(\ldots, a_p, \ldots, a_q, \ldots), I_{j_l}(a_q) \\ I_{j_l}(a_q) :- & R_{i_{l+1}}(\ldots, a_q, \ldots). \end{array}$$

Note that the first two lines in this derivation can instead appear in the "merged" form $G : -R_{i_1}(\ldots, a_0, \ldots, a_i, \ldots), I_{j_1}(a_i)$. Consider a substructure $\mathbf{C}'$ of $\mathbf{C}$ such that, for any $R \in \tau$ and any $\bar{a}$, we have $\bar{a} \in R^{\mathbf{C}'}$ if and only if $R(\bar{a})$ appears in the above derivation. Now modify the structure $\mathbf{C}'$ by giving new names to the occurrences of each element in such a way that in the obtained structure we have the following:

- there is no repetition of elements in any tuple in any relation, and

- if two tuples (possibly in different relations) share an element then this element appears in the heads of all rules between the rules corresponding to the two tuples.

Call the obtained structure $\mathbf{D}$. It is clear that $\mathbf{D}$ is a caterpillar. We also have that $\mathbf{D}$ homomorphically maps to $\mathbf{C}'$ (via reverse renaming) and hence to $\mathbf{C}$ and to $\mathbf{A}$, but not to $\mathbf{B}$ because the program still derives the goal predicate on $\mathbf{D}$. Hence, $\mathbf{B}$ has caterpillar duality.

The proof for the jellyfish case is very similar. If co-CSP($\mathbf{B}$) is defined by a jellyfish program, it easy to check that the same sequence of steps as above will result in a structure $\mathbf{D}$ that is a jellyfish.

Conversely, assume that $\mathbf{B}$ has caterpillar duality, i.e., for any structure $\mathbf{A}$ we have $\mathbf{A} \to \mathbf{B}$ if and only if all caterpillars that homomorphically map to $\mathbf{A}$ also map to $\mathbf{B}$. We claim that the canonical caterpillar program for $\mathbf{B}$ defines co-CSP($\mathbf{B}$). By Fact 6, the canonical program never accepts a structure that homomorphically maps to $\mathbf{B}$. Now let $\mathbf{A} \in$ co-CSP($\mathbf{B}$) be arbitrary. By assumption, there is a caterpillar $\mathbf{C}$ such that $\mathbf{C} \to \mathbf{A}$ and $\mathbf{C} \not\to \mathbf{B}$. It follows from Lemma 20 that the program accepts $\mathbf{A}$. The proof of this direction for the jellyfish case is identical. ■

Corollary 22. *If* co-CSP($\mathbf{B}$) *is definable by some caterpillar (jellyfish) program then it is definable by the canonical one.*

Lemma 23. *(i) A structure* $\mathbf{A}$ *is not accepted by the canonical caterpillar program for* $\mathbf{B}$ *if and only if* $\mathbf{A} \to \mathbf{C}(\mathbf{B})$.

(ii) A structure $\mathbf{A}$ *is not accepted by the canonical jellyfish program for* $\mathbf{B}$ *if and only if* $\mathbf{A} \to \mathbf{J}(\mathbf{B})$.

Proof. Assume first that $\mathbf{A} \to \mathbf{C}(\mathbf{B})$ and show that $\mathbf{A}$ is not accepted by the canonical caterpillar program. Since the class of structures accepted by any Datalog program is closed under homomorphism, it suffices to show that $\mathbf{C}(\mathbf{B})$ is not accepted by the canonical program. We will show by induction on the length of derivation that whenever the fact $I_j(\mathcal{S})$ is derived by the program, we have $S_j \in \mathcal{S}$ where S_j is the subset of B corresponding to I_j.

Assume first that $I_j(\mathcal{S})$ is derived by an introductory rule (i.e., one whose body contains no IDB and $R \in \tau$ is the EDB in the rule), that is, we have $I_j(\mathcal{S}) : -R(\ldots, \mathcal{S}, \ldots)$ where $\mathcal{S}$ appears in the m-th component in the tuple on the right. Note that this tuple belongs to $R^{\mathbf{C}(\mathbf{B})}$. Then, by the definition of the canonical program, I_j corresponds to the subset $\mathrm{pr}_m(R^{\mathbf{B}})$ of B, which must be contained in $\mathcal{S}$ by the definition of $\mathbf{C}(\mathbf{B})$.

Assume now that $I_j(\mathcal{S})$ is derived by a rule $I_j(\mathcal{S}) : -R(\ldots, \mathcal{S}, \ldots, \mathcal{T}, \ldots), I_l(\mathcal{T})$. Assume without loss of generality that S_j is the smallest set such that I_j can be in the head of this rule. By the induction hypothesis, we have $S_l \in \mathcal{T}$. Assume that $\mathcal{S}$ appears in the m-th component and $\mathcal{T}$ in the k-th component in the EDB of the rule. Again, by the definition of the program, we have $S_j = \mathrm{pr}_m(R^{\mathbf{B}} \cap (B^{k-1} \times S_l \times B^{r-k}))$ where r is the arity of R. Then we must have $S_j \in \mathcal{S}$ by the definition of $\mathbf{C}(\mathbf{B})$.

Assume now that $\mathbf{C}(\mathbf{B})$ is accepted by the canonical caterpillar program. Then the program can derive $I_0(\mathcal{S})$ for some $\mathcal{S}$. Then, as we just proved, the empty set S_0 belongs to $\mathcal{S}$ which is impossible by the definition of $\mathbf{C}(\mathbf{B})$.

To complete the proof of this direction for part (ii), notice that if the canonical jellyfish program derives the goal predicate on $\mathbf{J}(\mathbf{B})$ via a non-linear rule then we can assume that this rule is of the form

$$G : -R(X_1, \ldots, X_r), I_{11}(X_1), , \ldots, I_{1s_1}(X_1), \ldots, I_{r1}(X_r), \ldots, I_{rs_r}(X_r).$$

By the definition of canonical jellyfish program we have that

$$\{(b_1, \ldots, b_r) \in R^{\mathbf{B}} \mid b_i \in \bigcap_{j=1}^{s_i} S_{ij} \text{ for each } i\} = \emptyset$$

where the sets S_{ij} correspond to the IDBs I_{ij} in the program. For every $i = 1, \ldots, r$ let $\mathcal{S}^i$ be the element of $\mathbf{J}(\mathbf{B})$ to which X_i is instantiated when applying the non-linear rule above. As for the caterpillar case (see above), we have that $S_{ij} \in \mathcal{S}^i$ for all $i = 1, \ldots, r$ and $j = 1, \ldots, s_i$. It follows that $R^{\mathbf{B}} \cap (\bigcap \mathcal{S}^1 \times \cdots \times \bigcap \mathcal{S}^r) = \emptyset$ which contradicts the definition of $\mathbf{J}(\mathbf{B})$.

Conversely, assume that $\mathbf{A}$ is not accepted by the canonical caterpillar program. Hence the program stabilizes without deriving the goal predicate. Recall that every IDB I_j in the canonical program corresponds to a subset S_j of B. For every element $a \in A$, consider the family $\mathcal{S}_a = \{S_j \mid I_j(a) \text{ is derived}\}$ of subsets of B. It is easy to see that the family is non-empty for any a that appears in a tuple in a relation in $\mathbf{A}$. Moreover, since the goal predicate is not derived, $I_0(a)$ is not derived either, and so each subset in a non-empty $\mathcal{S}_a$ is non-empty. It is straightforward to check that the mapping $h : A \to C(B)$ given by $h(a) = \mathcal{S}_a$ is a homomorphism from $\mathbf{A}$ to $\mathbf{C}(\mathbf{B})$. The proof for the jellyfish case is very similar, with the very same h. ■

Lemma 24. *For any structure* $\mathbf{B}$*,*

(i) co-CSP$(\mathbf{B})$ *is definable by a caterpillar program if and only if* $\mathbf{C}(\mathbf{B}) \to \mathbf{B}$.

(ii) co-CSP$(\mathbf{B})$ *is definable by a jellyfish program if and only if* $\mathbf{J}(\mathbf{B}) \to \mathbf{B}$.

Proof. We will prove (i), the proof of (ii) is very similar. Suppose that co-CSP($\mathbf{B}$) is definable by a caterpillar program. Then it is definable by the canonical one, by Corollary 22. By Lemma 23, $\mathbf{C}(\mathbf{B})$ is not accepted by the canonical program, and hence $\mathbf{C}(\mathbf{B}) \to \mathbf{B}$.

Conversely, suppose that $\mathbf{C}(\mathbf{B}) \to \mathbf{B}$. By Fact 6, each structure from CSP($\mathbf{B}$) is not accepted by the canonical caterpillar program for $\mathbf{B}$. On the other hand, if a structure $\mathbf{A}$ is not accepted by the program then we have $\mathbf{A} \to \mathbf{C}(\mathbf{B})$ by Lemma 23, and so $\mathbf{A} \to \mathbf{B}$. Hence, co-CSP($\mathbf{B}$) is definable by the canonical caterpillar program for $\mathbf{B}$. ■

Lemma 25. *(i) The relations in* $\mathbf{C}(\mathbf{B})$ *are invariant under the operations* $x \cap y$ *and* $x \cup y$.
(ii) The relations in $\mathbf{J}(\mathbf{B})$ *are invariant under the operation* $x \cap (y \cup z)$.

Proof. (i) Let $R^{\mathbf{C}(\mathbf{B})}$ be an r-ary relation in $\mathbf{C}(\mathbf{B})$ and take arbitrary tuples $(\mathcal{S}^1, \dots, \mathcal{S}^r)$ and $(\mathcal{T}^1, \dots, \mathcal{T}^r)$ in $R^{\mathbf{C}(\mathbf{B})}$.

It follows directly from the definition of $\mathbf{C}(\mathbf{B})$ that for all $j, m = 1, \dots, r$ we have

1. $\mathrm{pr}_m(R^{\mathbf{B}}) \in \mathcal{S}^m \cap \mathcal{T}^m$, and
2. $\mathrm{pr}_m(R^{\mathbf{B}} \cap (B^{j-1} \times S \times B^{r-j})) \in \mathcal{S}^m \cap \mathcal{T}^m$ for each $S \in \mathcal{S}^j \cap \mathcal{T}^j$.

It follows that $(\mathcal{S}^1 \cap \mathcal{T}^1, \dots, \mathcal{S}^r \cap \mathcal{T}^r) \in R^{\mathbf{C}(\mathbf{B})}$. The fact that $(\mathcal{S}^1 \cup \mathcal{T}^1, \dots, \mathcal{S}^r \cup \mathcal{T}^r) \in R^{\mathbf{C}(\mathbf{B})}$ can be verified equally easily.

(ii) By part (i), it is enough to show that if $(\mathcal{S}^1, \dots, \mathcal{S}^r), (\mathcal{T}^1, \dots, \mathcal{T}^r), (\mathcal{U}^1, \dots, \mathcal{U}^r) \in R^{\mathbf{J}(\mathbf{B})}$, and $\mathcal{V}^i = \mathcal{S}^i \cap (\mathcal{T}^i \cup \mathcal{U}^i)$ for $1 \le i \le r$ then we have $R^{\mathbf{B}} \cap (\bigcap \mathcal{V}^1 \times \ldots \times \bigcap \mathcal{V}^r) \neq \emptyset$. However, the last condition follows trivially from the corresponding condition for $(\mathcal{S}^1, \dots, \mathcal{S}^r)$ because $\mathcal{V}^i \subseteq \mathcal{S}^i$ (and hence $\bigcap \mathcal{V}^i \supseteq \bigcap \mathcal{S}^i$) for all i. ■

Lemma 26. *A structure* $\mathbf{B}$ *has a* kn*-ary* k*-ABS polymorphism for all* k, n *if and only if* $\mathbf{C}(\mathbf{B}) \to \mathbf{B}$.

Proof. Let $h : \mathbf{C}(\mathbf{B}) \to \mathbf{B}$ be a homomorphism. By Lemma 25, the structure $\mathbf{C}(\mathbf{B})$ has polymorphisms which are set-theoretic union and intersection operations. Since composition of polymorphisms is again a polymorphism, it follows that $\mathbf{C}(\mathbf{B})$ also has the k-ABS polymorphisms

$$f_{k,n}(X_1, \dots, X_n) = (\bigcap X_1) \cup (\bigcap X_2) \cup \ldots \cup (\bigcap X_n),$$

where $X_i = \{x_{i1}, \dots, x_{ik}\}$. By Lemma 23, there exists a homomorphism $g : \mathbf{B} \to \mathbf{C}(\mathbf{B})$. It is straightforward to check that the operations $h(f_{k,n}(g(x_{11}), \dots, g(x_{kn})))$ are k-ABS polymorphisms of $\mathbf{B}$.

Conversely, let f be a kn-ary k-ABS polymorphism of $\mathbf{B}$ with $k = \rho|B|$ and $n = \rho(2^{|B|}-1)$, where ρ is the maximum of the arities of the relations in $\mathbf{B}$. Define a map $h : C(B) \to B$ by the rule $h(\mathcal{S}) = f(\mathcal{S})$ for non-empty $\mathcal{S}$ and set $h(\emptyset)$ arbitrarily. Let us now show that h is a homomorphism. By the properties of f, h is clearly a well-defined function. Take an arbitrary (say, r-ary) relation $R \in \tau$ and fix $(\mathcal{S}^1, \dots, \mathcal{S}^r) \in R^{\mathbf{C}(\mathbf{B})}$. We need to show that $(h(\mathcal{S}^1), \dots, h(\mathcal{S}^r)) \in R^{\mathbf{B}}$.

Let $\hat{\mathcal{S}}^i = \{X \cap \mathrm{pr}_i R^{\mathbf{B}} \mid X \in \mathcal{S}^i\}$. It immediately follows from the definition of the structure $\mathbf{C}(\mathbf{B})$ that we have $\hat{\mathcal{S}}^i \subseteq \mathcal{S}^i$ for all $1 \le i \le r$, and also that $(\hat{\mathcal{S}}^1, \dots, \hat{\mathcal{S}}^r) \in R^{\mathbf{C}(\mathbf{B})}$. Since f is absorptive, we have $f(\hat{\mathcal{S}}^i) = f(\mathcal{S}^i)$. Therefore, we can without loss of generality assume that each $\mathcal{S}^i$ contains only subsets of $\mathrm{pr}_i\, R^{\mathbf{B}}$.

For a set $S \in \mathcal{S}^i$, construct a $(k \times r)$-matrix M^i_S whose entries are elements from B and such that

1. each row of M^i_S is an element of $R^{\mathbf{B}}$, and
2. for any $1 \leq m \leq r$, the set of entries in the m-th column is $\mathrm{pr}_m(R^{\mathbf{B}} \cap (B^{i-1} \times S \times B^{r-i}))$.

Let us show that this is possible. Recall that $S \subseteq \mathrm{pr}_i(R^{\mathbf{B}})$. Divide the matrix into r submatrices of $|B|$ consecutive rows. For $1 \leq m \leq r$, the rows of the m-th submatrix are tuples (from $R^{\mathbf{B}}$) whose i-th coordinate belongs to S and whose m-th coordinates cover all of $\mathrm{pr}_m(R^{\mathbf{B}} \cap (B^{i-1} \times S \times B^{r-i}))$. Note that this submatrix can contain repeated rows.

Now construct a matrix M with kn rows and r columns, as follows. It is divided into n layers of k consecutive rows, each layer is a matrix M^i_S for some $1 \leq i \leq r$ and some $S \in \mathcal{S}^i$, and each matrix of this form appears as a layer. By the choice of n, this is possible.

It remains to notice that the operation f applied to the i-th column of M gives the value $f(\mathcal{S}^i)$, and, since f is a polymorphism of $\mathbf{B}$ and every row of M is in $R^{\mathbf{B}}$, we have $(f(\mathcal{S}^1), \ldots, f(\mathcal{S}^r)) \in R^{\mathbf{B}}$, as required. Thus $(h(\mathcal{S}^1), h(\mathcal{S}^2), \ldots, h(\mathcal{S}^r)) \in R^{\mathbf{B}}$. We conclude that $h : \mathbf{C}(\mathbf{B}) \to \mathbf{B}$. ■

Lemma 27. *A structure* $\mathbf{B}$ *has a* $(kn+1)$*-ary extended* k*-ABS polymorphism for all* k, n *if and only if* $\mathbf{J}(\mathbf{B}) \to \mathbf{B}$.

Proof. Let $h : \mathbf{J}(\mathbf{B}) \to \mathbf{B}$. By Lemma 25, the structure $\mathbf{J}(\mathbf{B})$ has polymorphism $q_2(x, y, z) = x \cap (y \cup z)$. Since composition of polymorphisms is again a polymorphism, we obtain that $q_3(x, y, z, z') = q_2(x, q_2(x, y, z), z') = x \cap (y \cup z \cup z')$ is also a polymorphism of $\mathbf{J}(\mathbf{B})$. Continuing in the same way, we obtain that, for any $n \geq 2$, the operation $q_n(x_0, x_1, \ldots, x_n) = x_0 \cap (\bigcup_{i=1}^n x_i)$ is also a polymorphism. The operation $r_2(x, y) = q_2(x, y, y) = x \cap y$ is a polymorphism of $\mathbf{J}(\mathbf{B})$. Composing r_2 with itself, one can get the operation $r_k(x_1, \ldots, x_k) = \bigcap_{i=1}^k x_i$ for any $k \geq 2$. It follows that, for any n, k, the structure $\mathbf{J}(\mathbf{B})$ also has the polymorphism

$$f_{n,k}(x_0, X_1, \ldots, X_n) = q_n(x_0, r_k(X_1), \ldots, r_k(X_n)) = x_0 \cap ((\bigcap X_1) \cup \ldots \cup (\bigcap X_n))$$

where $X_i = \{x_{i1}, \ldots, x_{ik}\}$ (the case when $k = 1$ or $n = 1$ can be easily obtained by identifying variables). Notice that $f_{n,k}$ is of the form given in Example 17, for the lattice $(J(B), \cap, \cup)$. By Lemma 23, there exists a homomorphism $g : \mathbf{B} \to \mathbf{J}(\mathbf{B})$. It is straightforward to check that the operations $h(f_{n,k}(g(x_0), g(x_{11}), \ldots, g(x_{kn})))$ are extended k-ABS polymorphisms of $\mathbf{B}$.

Conversely, let f be a $(kn+1)$-ary extended k-ABS polymorphism of $\mathbf{B}$ with $k = \rho|B|$ and $n = \rho(2^{|B|} - 1)$, where ρ is the maximum of the arities of the relations in $\mathbf{B}$. Define a map $h : J(B) \to B$ by the rule $h(\mathcal{S}) = f(x, \mathcal{S})$ for each $\mathcal{S}$ such that $x \in \bigcap \mathcal{S}$ and, if $\bigcap \mathcal{S} = \emptyset$, set $h(\mathcal{S})$ arbitrarily. Note that, by the replacement property of f, $f(x, \mathcal{S})$ does not depend on the choice of x as long as $x \in \bigcap \mathcal{S} \neq \emptyset$, so h is a well-defined function. Let us now show that h is a homomorphism. Take an arbitrary (say, r-ary) relation $R \in \tau$ and fix $(\mathcal{S}^1, \ldots, \mathcal{S}^r) \in R^{\mathbf{J}(\mathbf{B})}$. We need to show that $(h(\mathcal{S}^1), \ldots, h(\mathcal{S}^r)) \in R^{\mathbf{B}}$.

By the definition of $\mathbf{J}(\mathbf{B})$, we have that each $\mathcal{S}^i$ has a non-empty intersection, and moreover, there exists a tuple $(a_1, \ldots, a_r) \in R^{\mathbf{B}} \cap (\bigcap \mathcal{S}^1 \times \ldots \times \bigcap \mathcal{S}^r)$. We can now assume that, for each i, $h(\mathcal{S}^i) = f(a_i, \mathcal{S}^i)$. This is where the replacement property is important in this proof. As in the previous proof, it follows from the properties of f and of $\mathbf{J}(\mathbf{B})$ that we can without loss of generality assume that each $\mathcal{S}^i$ contains only subsets of $\mathrm{pr}_i(R^{\mathbf{B}})$. Finally, we construct the matrix M exactly as in the previous proof, and then we add one new row $a_1, \ldots, a_r$ at the very top of the matrix. Now, as in the previous proof, the operation f applied to the i-th column of the new matrix gives the value $f(a_i, \mathcal{S}^i)$, and, since f is a polymorphism of $\mathbf{B}$ and every row of the new matrix is in $R^{\mathbf{B}}$, we conclude that $(h(\mathcal{S}^1), \ldots, h(\mathcal{S}^r)) \in R^{\mathbf{B}}$, as required, and thus $h : \mathbf{J}(\mathbf{B}) \to \mathbf{B}$. ■

Remark 28. If a structure $\mathbf{B}$ has a kn-ary k-ABS polymorphism (or $(kn+1)$-ary extended k-ABS polymorphism) for $k = \rho|B|$ and $n = \rho(2^{|B|}-1)$, where ρ is the maximum of the arities of the relations in $\mathbf{B}$, then, for any k, $\mathbf{B}$ has k-ABS polymorphisms of all arities divisible by k (respectively, extended k-ABS polymorphisms of all possible arities) .

Proof. (of Theorem 16).

(1) ⇔ (2) follows from Lemma 21.

(2) ⇔ (3) follows from Lemma 24.

(3) ⇔ (4) follows from Lemma 26.

(4) ⇒ (5) Let $\mathbf{B}' = \mathrm{core}(\mathbf{B})$ and let $r : \mathbf{B} \to \mathbf{B}'$ be the corresponding retraction. It is well known and easy to see that, for each polymorphism f of $\mathbf{B}$, the operation $r \circ f$ (restricted to $\mathbf{B}'$) is a polymorphism of $\mathbf{B}'$. In particular, it follows that $\mathbf{B}'$ also has a kn-ary k-ABS polymorphism for all k, n. By Lemmas 23 and 26, $\mathbf{B}'$ is homomorphically equivalent to $\mathbf{C}(\mathbf{B}')$. Since $\mathbf{B}$ is (obviously) homomorphically equivalent to $\mathbf{B}'$, we can take the structure $\mathbf{C}(\mathbf{B}')$ as the required $\mathbf{A}$, it has the necessary polymorphisms by Lemma 25.

(5) ⇒ (6) Trivial.

(6) ⇒ (1) It is follows from Example 15(ii) that the structure $\mathbf{A}'$ has kn-ary k-ABS polymorphisms for all k, n. We already showed that (1) ⇔ (4), so it follows that $\mathbf{A}'$ has caterpillar duality. It is clear that homomorphically equivalent structures have this property simultaneously.

■

Proof. (of Theorem 18). The proof of the implications (1) ⇔ (2) ⇔ (3) ⇔ (4) ⇒ (5) is very similar to the previous proof. The proof that (5) ⇒ (1) is also similar to that of (6) ⇒ (1) above, but we need to show how to obtain, by using composition, extended k-ABS polymorphisms of all arities from the operation $x \sqcap (y \sqcup z)$ of a distributive lattice. In fact, we already showed this in the (beginning of the) proof of Lemma 27 in the case when the elements are subsets and the operations are set-theoretic intersection and union. However, as we mentioned before, any distributive lattice can be represented by a family of subsets, with operations acting as set-theoretic intersection and union, so we are done. ■

Remark 29. Note that, in contrast with Theorem 16, there is no item (6) in Theorem 18. This is because it is unclear whether the operation $x \sqcap (y \sqcup z)$ on a *non-distributive* lattice can generate (via composition) extended k-ABS operations of all necessary arities.

3.3 Some remarks about the new dualities

We will start with the following useful remark.

Remark 30. *Any core structure with caterpillar duality has a majority polymorphism.*

Indeed, if f is an idempotent 6-ary 2-ABS operation then it is easy to check that $f(x, y, z, x, y, z)$ is a majority operation. Therefore, by Theorem 16, any structure with caterpillar duality has a majority polymorphism (compare with Theorem 14). However there exist structures with jellyfish duality, but without majority polymorphisms. For instance take any structure $\mathbf{B}_{ij}$ from Example 12 with $i + 1 \neq j$ and $(i, j) \neq (1, n)$. The obstructions given in that Example are minimal (in the sense that any proper substructure belongs to $\mathrm{CSP}(\mathbf{B}_{ij})$) and have at least three coloured elements. By Theorem 1.17 of [34], such a structure cannot have a majority polymorphism. However, any core structure with jellyfish duality has an NU polymorphism of some (possibly large) arity.

Proposition 31. *Any core structure with jellyfish duality has an NU polymorphism.*

Proof. Let $\mathbf{B}$ be a core structure with jellyfish duality. It suffices to show that $\mathbf{J}(\mathbf{B})$ has an NU polymorphism because it can be carried over to $\mathbf{B}$ as in the proof of Lemma 27. Let ρ be the maximum arity of a relation in $\mathbf{B}$ and define $f(x_1, \ldots, x_m) = \bigcap_{i<j} x_i \cup x_j$ where $m = \rho(2^{|B|} - 1) + 1$. In other words, if $\mathcal{S}^1, \ldots, \mathcal{S}^m$ are elements of $\mathbf{J}(\mathbf{B})$ then $f(\mathcal{S}^1, \ldots, \mathcal{S}^m)$ is defined to contain exactly all those non-empty subsets S of B that appear in at least $m - 1$ of the arguments. Operation f is clearly an NU operation. It remains to show that it is a polymorphism of every relation $R^{\mathbf{J}(\mathbf{B})}$ of $\mathbf{J}(\mathbf{B})$. Indeed, let $(\mathcal{T}_1, \ldots, \mathcal{T}_r)$ be the result of applying f component-wise to tuples $\overline{t}^i = (\mathcal{S}^i_1, \ldots, \mathcal{S}^i_r)$, $i = 1, \ldots, m$, of $R^{\mathbf{J}(\mathbf{B})}$. The definition of f easily implies that $(\mathcal{T}_1, \ldots, \mathcal{T}_r)$ belongs to $R^{\mathbf{C}(\mathbf{B})}$. To see that it also belongs to $R^{\mathbf{J}(\mathbf{B})}$ it is only necessary to observe that for every $j = 1, \ldots, r$ and every $S \in \mathcal{T}_j$ there is at most one tuple in $\{\overline{t}^1, \ldots, \overline{t}^m\}$ such that its jth component does *not* contain S. Hence, by the pigeon-hole principle, there exists a tuple $\overline{t}^i = (\mathcal{S}^i_1, \ldots, \mathcal{S}^i_r)$ such that for every $j = 1, \ldots, r$, $\mathcal{T}_j \subseteq \mathcal{S}^i_j$. It follows that

$$R^{\mathbf{B}} \cap (\bigcap \mathcal{T}_1 \times \cdots \bigcap \mathcal{T}_r) \supseteq R^{\mathbf{B}} \cap (\bigcap \mathcal{S}^i_1 \times \cdots \bigcap \mathcal{S}^i_r) \neq \emptyset.$$

■

We will now discuss the problem of recognising structures with a given duality.

Lemma 32. *For any class $\mathcal{C}$ of oriented trees that includes all directed paths, the problem of deciding whether a given structure has some subclass of $\mathcal{C}$ as an obstruction set is* **NP***-hard.*

Proof. We prove the lemma by reduction from 3-Sat. Let $\mathbf{T}_n$ be the transitive tournament on n vertices. It is shown in the proof of Theorem 6.1 of [28] that, given an instance $\mathcal{I}$ of 3-Sat, one can construct in polynomial time a digraph $\mathbf{H}$ such that (i) $\mathbf{T}_n$ is the core of $\mathbf{H}$ if and only if $\mathcal{I}$ is satisfiable, and (ii) either $\mathbf{T}_n$ is the core of $\mathbf{H}$ or else $\mathbf{H}$ does not have tree duality. It remains to say that the directed path on $n+1$ vertices forms an obstruction set for $\mathbf{T}_n$ [20], so $\mathbf{T}_n$ has caterpillar duality. ■

Theorem 33. *The problem of checking whether a given structure has caterpillar (jellyfish) duality is decidable, but* **NP***-hard.*

Proof. Decidability of the problem immediately follows from condition (3) of Theorems 16 and 18, respectively, while **NP**-hardness follows from Lemma 32. ■

Remark 34. It is unknown whether Lemma 32 and the hardness part of Theorem 33 remain valid if input structures in the problem are required to be cores. It is known that, for the case of finite duality, the complexity of the problem changes (from **NP**-hardness for arbitrary structures to **PTIME** for cores) [28].

One can define τ*-paths* as τ-caterpillars with at most two pendant blocks. Say, if τ is the signature of digraphs then τ-paths are oriented paths (i.e., digraphs obtained from paths by orienting each edge in some way). One can also define path duality in a natural way, and obtain a characterisation similar to conditions (1)-(3) of Theorem 16. However, the fragment of Datalog arising from this connection is not very natural and it does not seem to have a natural equivalent algebraic condition such as conditions (4)–(6) of Theorem 16. Since paths and caterpillars are very close structurally, it is natural to ask whether caterpillar duality and path duality are equivalent properties. It is easy to answer this (and, in fact, a more general) question by using digraphs with finite duality.

Proposition 35. *Let* $\mathbf{T}$ *be an oriented tree that is a core digraph and let* $\mathcal{O}$ *be any class of digraphs such that* $\mathbf{T}$ *is not a core of any digraph in* $\mathcal{O}$*. Then there is a digraph* $\mathbf{B}$ *such that* $\{\mathbf{T}\}$ *is an obstruction set for* $\mathbf{B}$*, but* $\mathcal{O}$ *is not.*

Proof. By results of [32], there exists a structure $\mathbf{B}$ such that $\{\mathbf{T}\}$ is the obstruction set for $\mathbf{B}$, that is, for any digraph $\mathbf{G}$, we have either $\mathbf{T}\rightarrow\mathbf{G}$ or else $\mathbf{G}\rightarrow\mathbf{B}$. We claim that $\mathcal{O}$ is not an obstruction set for $\mathbf{B}$. Suppose, for a contradiction, that it is. Then, since $\mathbf{T}\not\rightarrow\mathbf{B}$, there is a digraph $\mathbf{P}\in\mathcal{O}$ such that $\mathbf{P}\rightarrow\mathbf{T}$ and $\mathbf{P}\not\rightarrow\mathbf{B}$. The latter property implies that $\mathbf{T}\rightarrow\mathbf{P}$, in which case $\mathbf{T}$ must be the core of $\mathbf{P}$, a contradiction. ■

For example, using the above proposition with $\mathbf{T}$ being a core caterpillar and $\mathcal{O}$ any class of oriented paths, we get a structure $\mathbf{B}$ that has caterpillar duality, but not path duality. Similarly, one can get a structure with jellyfish duality, but not caterpillar duality.

4 Applications to list H-colouring

In the *list* $\mathbf{H}$*-colouring* problem for a fixed (di)graph $\mathbf{H}$, one is given a (di)graph $\mathbf{G}$, and, for each vertex v of $\mathbf{G}$, a list L_v of possible target vertices in $\mathbf{H}$. The question is whether there is a homomorphism $h:\mathbf{G}\rightarrow\mathbf{H}$ such that $h(v)\in L_v$ for each vertex v of $\mathbf{G}$. It is well known (and easy to see) that this problem is exactly CSP($\mathbf{H}_u$) where $\mathbf{H}_u$ is the structure obtained by expanding the (di)graph $\mathbf{H}$ with unary relations U where U runs through all non-empty subsets of H. It is easy to see that the polymorphisms of $\mathbf{H}_u$ are exactly the conservative polymorphisms of $\mathbf{H}$.

Recall that a (di)graph is called *reflexive* if it contains all self-loops, and *irreflexive* if it contains no self-loop.

4.1 List H-colouring for undirected graphs

All graphs in this subsection are undirected. It was shown in [14] that, for a reflexive graph $\mathbf{H}$, the list $\mathbf{H}$-colouring problem is solvable in polynomial time if $\mathbf{H}$ is an interval graph and **NP**-complete otherwise. Recall that a (reflexive) graph is called an *interval graph* if its vertices can be represented by intervals (on the real line) in such a way that two vertices are adjacent if and only if the corresponding intervals intersect.

Assume now that $\mathbf{H} = (H, E)$ is a reflexive interval graph. By modifying the proof in [14], it is possible to show directly that the structure $\mathbf{H}_u$ (as above) has caterpillar duality. We give a short proof of this fact using Theorem 16.

Theorem 36. *For every k and n, the graph $\mathbf{H}$ has a conservative k-ABS polymorphism of arity kn.*

Proof. Fix an interval representation of $\mathbf{H}$. We can without loss of generality assume that the endpoints of the intervals representing vertices of $\mathbf{H}$ are pairwise distinct [14]. Given an interval $u \in V$, we denote by $l(u)$ and $r(u)$ the left and right endpoints of u, respectively.

Let $k, n \geq 1$ be arbitrary. Define two functions on H, Min_l and Max_r, as follows:

$$\mathrm{Min}_l(u_1, \ldots, u_n) = u_i \text{ such that } l(u_i) = \min_j l(u_j),$$

$$\mathrm{Max}_r(u_1, \ldots, u_k) = u_i \text{ such that } r(u_i) = \max_j r(u_j).$$

Note that the functions are well defined because the intervals in H cannot have the same endpoints.

Let $S_1, S_2, \ldots, S_n$ be sets of vertices of $\mathbf{H}$ (i.e., sets of intervals) with at most k elements each. We obtain from them a new sequence of sets, as follows: for each $j = 1, \ldots, n$ such that S_j properly contains some set S_i, choose S_i so that S_i is minimal with this property and replace S_j by S_i. Break ties arbitrarily. Call the obtained sets $S'_1, S'_2, \ldots, S'_n$.

Define an operation $h : H^{nk} \to H$ as follows:

$$h(x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk}) = \mathrm{Min}_l(\mathrm{Max}_r(S'_1), \ldots, \mathrm{Max}_r(S'_n))$$

where, for $1 \leq i \leq n$, $S_i = \{x_{i1}, \ldots, x_{ik}\}$, and S'_i is obtained as described above. Note that the set $\{S'_1, \ldots, S'_n\}$ depends only on $\{S_1, \ldots, S_n\}$. This and the (obvious) fact that the operations Max_r and Min_l are totally symmetric implies that the operation h is well defined and also that it is a k-block symmetric operation.

Let us show that h is absorptive. We now can use the notation $h(S_1, S_2, \ldots, S_n)$ since h is k-block symmetric. Assume that $S_2 \subset S_1$. Then $S'_1 = S_i$ for some $i > 1$. Note that, by construction, we have $S'_i = S_i$. Assume without loss of generality that $i = 3$. Then we have

$$h(S_1, S_2, \ldots, S_n) = \mathrm{Min}_l(\mathrm{Max}_r(S'_3), \mathrm{Max}_r(S'_2), \ldots, \mathrm{Max}_r(S'_n)),$$

and

$$h(S_2, S_2, S_3, \ldots, S_n) = \mathrm{Min}_l(\mathrm{Max}_r(S'_2), \mathrm{Max}_r(S'_2), \ldots, \mathrm{Max}_r(S'_n)).$$

The right-hand sides of the above equations are the same (since Min_l is totally symmetric), so the left-hand sides are equal as well, as required.

It is obvious that h is conservative. It remains to show that it is a polymorphism of $\mathbf{H}$. For all $1 \leq i \leq n$ and $1 \leq l \leq k$, let s_{il} and t_{il} be intervals in V that intersect (i.e., adjacent in $\mathbf{H}$). Also let $S_i = \{s_{i1}, \dots, s_{ik}\}$ and $T_i = \{t_{i1}, \dots, t_{ik}\}$ for $1 \leq i \leq n$. We need to show that the intervals $s = h(S_1, \dots, S_n)$ and $t = h(T_1, \dots, T_n)$ also intersect.

We have $s = \mathrm{Min}_l(\mathrm{Max}_r(S'_1), \dots, \mathrm{Max}_r(S'_n))$ and $t = \mathrm{Min}_l(\mathrm{Max}_r(T'_1), \dots, \mathrm{Max}_r(T'_n))$. Hence, $s = \mathrm{Max}_r(S'_i)$ for some i and $t = \mathrm{Max}_r(T'_j)$ for some j. It is easy to see that i and j can be chosen so that $S_i = S'_i$ and $T_j = T'_j$. Since $S_i = S'_i$, and $T'_i \subseteq T_i$ we know that every interval in T'_i intersects some interval in S'_i. Similarly, every interval in S'_j intersects some interval in T'_j.

Suppose, for a contradiction, that $s \cap t = \emptyset$. Assume first that t precedes s, i.e. $r(t) < l(s)$. Since $s = \mathrm{Min}_l(\mathrm{Max}_r(S'_1), \dots, \mathrm{Max}_r(S'_n))$, we have $l(s) \leq l(\mathrm{Max}_r(S'_j))$. Since $\mathrm{Max}_r(S'_j) \in S'_j$, it intersects some interval $t_j \in T'_j$. In particular, we have $l(\mathrm{Max}_r(S'_j)) < r(t_j)$. By combining the three above inequalities, we obtain $r(t) < l(s) \leq l(\mathrm{Max}_r(S'_j)) < r(t_j)$, which contradicts the fact $t = \mathrm{Max}_r(T'_j)$. If $r(s) < l(t)$ then the argument is symmetric.

Thus h is a polymorphism and the theorem is proved. ■

Corollary 37. *For a reflexive graph* $\mathbf{H}$*, either* $\mathbf{H}_u$ *has caterpillar duality or* $\mathrm{CSP}(\mathbf{H}_u)$ *is* **NP***-complete.*

Remark 38. If $\mathbf{H}$ is the reflexive claw (i.e., the complete bipartite graph $\mathbf{K}_{1,3}$ with loops) then it is easy to check that $\mathbf{H}_u$ does not have lattice polymorphisms, even though it is the core of a structure with such polymorphisms (by Theorem 16).

Remark 39. By using results from [28], one can show that, for a reflexive graph $\mathbf{H}$, $\mathbf{H}_u$ does not have finite duality unless the graph $\mathbf{H}$ is complete.

4.2 List H-colouring for directed graphs

All graphs in this subsection are directed. We will start by considering irreflexive digraphs. It was shown in [23] that every oriented path has path duality (that is it has an obstruction set consisting of oriented paths). Since every oriented path is a caterpillar, every oriented path has caterpillar duality. We will show how to generalise this result to a much wider class of digraphs which, in particular, includes all oriented caterpillars. A directed acyclic graph (DAG) $\mathbf{G}$ is called *layered* (or *balanced*) if each vertex u of $\mathbf{G}$ can be assigned a positive integer $l(u)$, the *level* of u, so that every arc (u, v) of $\mathbf{G}$ satisfies $l(u) + 1 = l(v)$. Every layered DAG can be embedded into the plane in such a way that each vertex u lies on the horizontal line $y = l(u)$, and the arcs are straight lines. If, in addition, the embedding can be arranged in such a way that the arcs never cross each other then the graph is called a *planar layered* DAG. It is easy to see that every oriented caterpillar is a planar layered DAG.

Theorem 40. *If* $\mathbf{H}$ *is a planar layered DAG then* $\mathbf{H}_u$ *has caterpillar duality.*

Proof. Fix a planar layered embedding of $\mathbf{H}$ into the plane such that the vertices lie on horizontal lines (as described above) and consider the following total order on H: $u < v$ if and only if either (i) $l(u) < l(v)$ or else (ii) $l(u) = l(v)$ and u is to the left of v.

Let min and max be the lattice operations with respect to the above order. We now show that they are polymorphisms of $\mathbf{H}$. Let (a_1, b_1) and (a_2, b_2) be arcs in $\mathbf{H}$. We need to show that $(\min(a_1, a_2), \min(b_1, b_2))$ and $(\max(a_1, a_2), \max(b_1, b_2))$ are also arcs in $\mathbf{H}$. We consider the former case, the latter is similar. Assume without loss of generality that $\min(a_1, a_2) = a_1$. If $l(a_1) < l(a_2)$ then $l(b_1) < l(b_2)$ so $\min(b_1, b_2) = b_1$ and we are done. If $l(a_1) = l(a_2)$ then $l(b_1) = l(b_2)$ and we again have $\min(b_1, b_2) = b_1$ because otherwise the arcs (a_1, b_1) and (a_2, b_2) would cross. By Example 15 and Theorem 16, we are done. ■

Reflexive digraphs that admit polymorphisms min and max with respect to some linear ordering of the vertices were characterised in [19]. Obviously, if $\mathbf{H}$ is such a digraph then $\mathbf{H}_u$ has caterpillar duality.

In the rest of this section, all digraphs are assumed to be reflexive. We will now consider reflexive digraphs that are oriented (reflexive) trees. Recall that a caterpillar (graph) is a tree which becomes a path after removing all its leaves. If we extend this path with one (arbitrary) leaf adjacent to the first node on the path and one (arbitrary) leaf adjacent to the last node on this path, then we will call this extended path the *main path* of the caterpillar. Oriented caterpillars will play an important role in the next result, so we will fix notation for them. Let $0, \ldots, p$ denote the elements of the main path, in any (fixed) of the two possible directions. For each $0 < i < p$, let L_i denote the set of leaves of the tree (except those on the main path) adjacent to i. Also, let $L'_i = \{i\} \cup L_i$. We fix an arbitrary total order on each (non-empty) set L_i. We will consider the operations min and max of taking minimum and maximum, respectively, elements on the main path or in some set L_i, it will always be clear from the context where these operations are considered.

It was shown in [15] that, for a (reflexive) oriented tree $\mathbf{H}$, CSP($\mathbf{H}_u$) is solvable in polynomial time if $\mathbf{H}$ is a good caterpillar, and it is **NP**-complete in all other cases. A good caterpillar is an oriented caterpillar where the ordering (one of the two) of the main path is chosen so that, for each element i on the main path, all arcs between i and L_i have the same direction (i.e., to i or from i), and, for $0 < i < p$, it is the direction of the arc i and $i + 1$. We will say that an oriented caterpillar $\mathbf{H}$ is *special* if it is good with respect to both orderings of the main path. In other words, an oriented caterpillar is special if every element i on the main path that is adjacent to at least three other nodes is either a sink or a source (meaning that the loop is the only arc leaving or coming into i, respectively).

Theorem 41. *Let* $\mathbf{H}$ *be an oriented reflexive tree. The* $\mathbf{H}_u$ *has caterpillar duality if and only if* $\mathbf{H}$ *is a special caterpillar.*

Proof. We prove necessity first. Note that if $\mathbf{H}$ is not a special caterpillar then it has an induced subdigraph isomorphic or anti-isomorphic to one of the digraphs shown in Fig. 4. (Note that the loops are not shown and the directions of the dashed arcs in the second digraph can be arbitrary). We will show that $\mathbf{H}$ does not have a conservative majority polymorphism, which, by Remark 30, is sufficient to prove that $\mathbf{H}_u$ does not have caterpillar duality. By conservativity, the restriction of such a polymorphism to any induced subdigraph $\mathbf{G}$ would be a conservative majority polymorphism for $\mathbf{G}$. Let $\mathbf{G}$ be one of these two digraphs from Fig. 4, and show that it cannot have such a polymorphism. Assume, for contradiction, that m is such a polymorphism. For the first digraph, the element $m(a, b_2, b_3)$ must have arcs going to both $b_2 = m(b_2, b_2, b_3)$ and $b_3 = m(b_3, b_2, b_3)$, so we must have $m(a, b_2, b_3) = a$. Furthermore, there is an arc from $m(b_1, b_2, b_3)$ to $m(a, b_2, b_3) = a$, and, by conservativity, we have $m(b_1, b_2, b_3) = b_1$. Finally, there is an arc from $a = m(b_1, a, a)$ to $m(b_1, b_2, b_3) = b_1$, a contradiction. For the second digraph, $m(b_1, b_2, b_3) \in \{b_1, b_2, b_3\}$, by conservativity. Assume, without loss of generality, that $m(b_1, b_2, b_3) = b_1$. Then $m(c_1, b_2, b_3)$ is adjacent to $m(b_1, b_2, b_3) = b_1$ in some direction, so it must be c_1, by conservativity. Finally, there must be an arc from $m(c_1, a, a) = a$ to $m(c_1, b_2, b_3) = c_1$, a contradiction again.

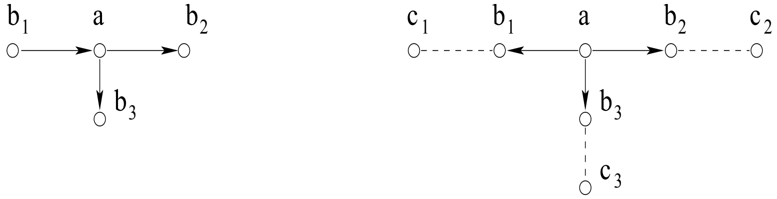


Figure 4: Oriented reflexive trees without a conservative majority polymorphism.

Assume now that $\mathbf{H}$ is a special caterpillar. It is enough, by Theorem 16, to show that $\mathbf{H}$ has a conservative nk-ary k-ABS polymorphism for all n, k. Note that if $\mathbf{H}'$ is the main path of $\mathbf{H}$ then (as is easy to check) the operations min and max are polymorphisms of $\mathbf{H}'$, and so the conservative nk-ary k-ABS operation

$$f(x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk}) = \min(\max(x_{11}, \ldots, x_{1k}), \ldots, \max(x_{n1}, \ldots, x_{nk}))$$

from Example 15(i) is also a polymorphism of $\mathbf{H}'$. Consider the unary operation r on $\mathbf{H}$ such that, for every node i on the main path and every leaf $x \in L_i$, we have $r(i) = r(x) = i$. Clearly, r is a polymorphism of $\mathbf{H}$. Hence, the operation $f(r(x_{11}), \ldots, r(x_{nk}))$ is also a polymorphism of $\mathbf{H}$ (though not necessarily conservative).

We will now define the required operation $g(x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk})$ on $\mathbf{H}$. In a sense, we will combine the above operation f with the operation h from the proof of Theorem 36. Take an arbitrary nk-tuple $(a_{11}, \ldots, a_{1k}, \ldots, a_{n1}, \ldots, a_{nk})$ of elements from H, and, for $1 \leq i \leq n$, let $S_i = \{a_{i1}, \ldots, a_{ik}\}$. Let $j = f(r(a_{11}), \ldots, r(a_{nk}))$ and define $S'_i = S_i \cap L'_j$ if $\max\{r(x) \mid x \in S_i\} = j$ and $S'_i = \emptyset$ if $\max\{r(x) \mid x \in S_i\} > j$. Next, for each non-empty set S'_i that properly contains some other non-empty set $S'_{i'}$, choose $S'_{i'}$ to be minimal with this property and replace S'_i by $S'_{i'}$. Break ties arbitrarily. Call the obtained sets $S''_1, \ldots, S''_n$. It is easy to see that the set $\{S''_1, \ldots, S''_n\}$ depends only on $\{S_1, \ldots, S_n\}$. Finally, we let

$$g(a_{11}, \ldots, a_{1k}, \ldots, a_{n1}, \ldots, a_{nk}) = \begin{cases} j & \text{if } j \in S''_1 \cup \ldots \cup S''_n \\ \min_{S''_i \neq \emptyset} \max(S''_i) & \text{otherwise} \end{cases}$$

Note that min and max in the last line of the above formula are computed in L_j.

It is straightforward to verify that the operation g defined above is a conservative k-ABS operation. It remains to prove that it is a polymorphism of $\mathbf{H}$. For all $1 \leq i \leq n$ and $1 \leq l \leq k$, let s_{il} and t_{il} be nodes in $\mathbf{H}$ such that (s_{il}, t_{il}) is an arc in $\mathbf{H}$. Also let $S_i = \{s_{i1}, \ldots, s_{ik}\}$ and $T_i = \{t_{i1}, \ldots, t_{ik}\}$ for $1 \leq i \leq n$. We need to show that (s, t) is also an arc where $s = g(S_1, \ldots, S_n)$ and $t = g(T_1, \ldots, T_n)$.

It is not hard to see that $r(s) = f(r(s_{11}), \ldots, r(s_{nk}))$ and $r(t) = f(r(t_{11}), \ldots, r(t_{nk}))$. Hence, $(r(s), r(t))$ is an arc because $f(r(x_{11}), \ldots, r(x_{nk}))$ is a polymorphism of $\mathbf{H}$. In particular, if s and t are both on the main path then $s = r(s)$ and $t = r(t)$, so we are done. Assume now that s is not, i.e., $s \in L_{r(s)}$. It follows that $r(s)$ is either a source or a sink and that all sets in $\{S''_1, \ldots, S''_n\}$ are subsets of $L_{r(s)}$.

Claim 1. If $r(s) = r(t)$ and $t \in L_{r(s)}$ then $s = t$.
Note that, since $t \in L_{r(s)}$, all sets in $\{T''_1, \dots, T''_n\}$ are subsets of $L_{r(s)}$. We will now prove that the non-empty sets in $\{S''_1, \dots, S''_n\}$ and $\{T''_1, \dots, T''_n\}$ are the same, which implies $s = t$. Assume first that $r(s)$ is a sink. Notice that, for every non-empty set S'_i that does not include $r(s)$, we have either $T'_i = S'_i$ or $\{r(s)\} \subseteq T'_i \subseteq \{r(s)\} \cup S'_i$. We know that, in the latter case, there exists a non-empty set $T''_{i'} = T'_{i'}$ which is included in T'_i and does not contain $r(s)$. In this case $S'_{i'} = T'_{i'}$ and it must be contained in S'_i. Therefore, each (inclusion-wise) minimal non-empty set in $\{S'_1, \dots, S'_n\}$ belongs to $\{T'_1, \dots, T'_n\}$. Take an arbitrary set T''_i such that $T''_i = T'_i$. All nodes in this set are sources, so $T'_i = S'_i$. It follows that each minimal non-empty set in $\{T'_1, \dots, T'_n\}$ belongs to $\{S'_1, \dots, S'_n\}$. It immediately follows that the only possible difference between $\{S''_1, \dots, S''_n\}$ and $\{T''_1, \dots, T''_n\}$ is that one of them contains the empty set and the other does not. If $r(s)$ is a source then the same argument, but reversing the role of the T's and S's, works. Claim 1 is proved.

If $r(s)$ is a sink then it immediately follows that $r(t) = r(s)$. In this case, if $t = r(t)$ then are done because there is an arc from s to $r(s)$; otherwise, we use Claim 1. Assume now that $r(s)$ is a source. It follows that each node in $L_{r(s)}$ is a sink and that $r(t)$ is one of $r(s) - 1, r(s), r(s) + 1$. We consider the three cases separately.

Case 1. $r(t) = r(s) - 1$. Take an arbitrary non-empty set $T''_i = T'_i$. Since $\max\{r(x) \mid x \in T_i\} = r(s) - 1$, the set S_i cannot contain elements from $L_{r(s)}$ or from $L'_{i'}$ with $i' > r(s)$. Furthermore, $\max\{r(x) \mid x \in S_i\} \geq r(s)$, so we have $S_i \cap L'_{r(s)} = \{r(s)\}$. It is now easy to see that $S''_i = \{r(s)\}$, which implies $s = r(s)$, contradicting our assumption that $s \in L_{r(s)}$.

Case 2. $r(t) = r(s)$. By Claim 1, we can assume that $t = r(t)$. Then one can choose an i such that $t \in T'_i = T''_i \subseteq L'_t$. Since t is a source, we have $t \in S'_i \subseteq T'_i$. Since $s \in L_{r(s)}$, it follows that there is an i' such that $S''_{i'} = S'_{i'} \subseteq S'_i \setminus \{t\} \subseteq L_t$. Then $T'_{i'} = S'_{i'}$ is non-empty and satisfies $T'_{i'} \subset T''_i$, which is impossible by minimality of T''_i.

Case 3. $r(t) = r(s) + 1$. This case is similar to the previous cases. Take an arbitrary non-empty set $S''_i = S'_i$. We have $\max\{r(x) \mid x \in S_i\} = r(s)$ and $\max\{r(x) \mid x \in T_i\} \geq r(s) + 1$. This implies that $r(s) + 1 \in T_i$ which, in turn, implies that $r(s) \in S'_i$. Since $s \in L_{r(s)}$, it follows that there is an i' such that $S''_{i'} = S'_{i'} \subseteq S'_i \cap L_{r(s)}$. Then $T'_{i'} = S'_{i'}$ is non-empty and satisfies $T'_{i'} \subset T''_i$, which is impossible by minimality of T''_i. ■

Now let $\mathbf{H}$ be a reflexive digraph, and let $\mathbf{H}_c$ denote the structure obtained from $\mathbf{H}$ by adding all unary relations of the form $\{a\}$, $a \in H$. The problem CSP($\mathbf{H}_c$) is known in graph theory as *one-or-all* list $\mathbf{H}$-homomorphism problem, and it is equivalent to the so-called $\mathbf{H}$-retraction problem [14, 20]. It is easy to see that the polymorphisms of $\mathbf{H}_c$ are the idempotent polymorphisms of $\mathbf{H}$. Note that if τ is the signature of $\mathbf{H}_c$ then a τ-path is an oriented path each of whose ends may belong to a unary relation.

Theorem 42. *For any reflexive digraph H, the following are equivalent:*

1. *$\mathbf{H}_c$ has caterpillar duality;*

2. *$\mathbf{H}_c$ has tree duality and a majority polymorphism;*

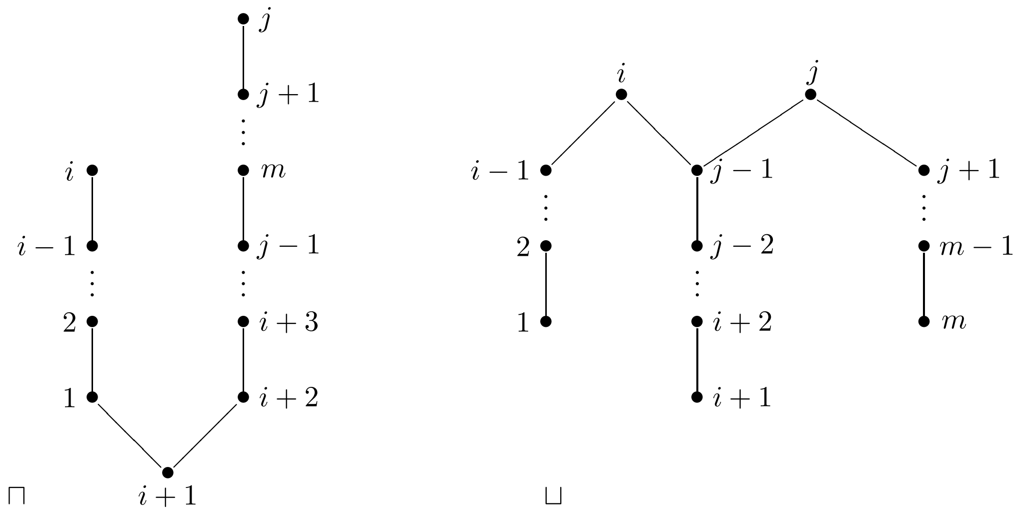
3. *$\mathbf{H}_c$ has path duality.*

Proof. Clearly, (3) implies (1). Caterpillar duality trivially implies tree duality, and, as we argued in the beginning of Section 3.3, it also implies the presence of a majority polymorphism, so (1) implies (2). Finally, let us show that (2) implies (3). Let τ be the signature of $\mathbf{H}_c$ (i.e., one binary and $|H|$ unary relations). Let $\mathbf{A}$ be a τ-structure such that $\mathbf{A} \not\to \mathbf{H}_c$. By the tree duality of $\mathbf{H}_c$, there exists a τ-tree $\mathbf{T}$ that is homomorphic to $\mathbf{A}$, but not to $\mathbf{H}_c$. Take $\mathbf{T}$ to be minimal, that is, any proper substructure of $\mathbf{T}$ is homomorphic to $\mathbf{H}_c$. Then, since $\mathbf{H}_c$ has a majority polymorphism, Theorem 1.17 of [34] implies that at most two elements of $\mathbf{T}$ are in unary relations in $\mathbf{T}$. This, the fact that $\mathbf{H}$ is reflexive, and the minimality of $\mathbf{T}$ imply that $\mathbf{T}$ is in fact a path. ■

Appendix: Polymorphisms for the structures from Example 12

In Example 12, we gave concrete non-trivial examples of structures with jellyfish duality. By Theorem 18, these structures $\mathbf{B}_{ij}$ should have extended k-ABS polymorphisms of all appropriate arities. In this Appendix we describe these polymorphisms.

Let $\sqcap$ be the (semilattice) operation, on B_{ij} of taking the greatest common lower bound with respect to the poset shown on the diagram below (left) and let $\sqcup$ be the partial operation of taking the least common upper bound with respect to the poset shown on the diagram below (right). Note that $\sqcup$ is also a semilattice operation when restricted to $\{1, \dots, j-1\}$ or to $\{i+1, \dots, m\}$.



Let $k, n \geq 1$ be arbitrary. Define two (partial) functions on $\mathbf{B}_{ij}$, Min and Max as follows:

$$\mathrm{Min}(u_1, \dots, u_k, u_{k+1}) = u_1 \sqcap u_2 \sqcap \dots \sqcap u_{k+1}, \qquad \mathrm{Max}(u_1, \dots, u_n) = u_1 \sqcup u_2 \sqcup \dots \sqcup u_n.$$

Obviously, Min is a totally symmetric operation, while Max is also totally symmetric when restricted to $\{1, \dots, j-1\}$ or to $\{i+1, \dots, m\}$.

Let $S_1, S_2, \dots, S_n$ be sets of vertices of $\mathbf{B}_{ij}$ with at most k elements each. We obtain from them a new sequence of sets, as follows. For each $l = 1, \dots, n$, if $S_l \cap \{1, \dots, i\} \neq \emptyset$, $S_l \cap \{i+1, \dots, m\} \neq \emptyset$ and there exists $S_p \subseteq \{1, \dots, i\}$ then let $\hat{S}_l := S_l \cap \{1, \dots, i\}$, otherwise let $\hat{S}_l := S_l$; if $\hat{S}_l \cap \{1, \dots, j-1\} \neq \emptyset$ and $\hat{S}_l \cap \{j, \dots, m\} \neq \emptyset$ and there exists $\hat{S}_p \subseteq \{j, \dots, m\}$ then let $S'_l := \hat{S}_l \cap \{j, \dots, m\}$, otherwise let $S'_l := \hat{S}_l$.

Define an operation $h : H^{nk+1} \to H$ as follows:

$$h(x, x_{11}, \dots, x_{1k}, \dots, x_{n1}, \dots, x_{nk}) = x \sqcap \mathrm{Max}(\mathrm{Min}(S'_1, x), \dots, \mathrm{Min}(S'_n, x))$$

where, for $1 \leq i \leq n$, $S_i = \{x_{i1}, \ldots, x_{ik}\}$, and S'_i is obtained as described above. Note that the set $\{S'_1, \ldots, S'_n\}$ depends only on $\{S_1, \ldots, S_n\}$. It is easy to check that h is a well-defined total operation, and that fixing any value for the first variable in h gives a k-block symmetric operation.

Claim 1. *The operation h is a polymorphism of $\mathbf{B}_{ij}$.*

Proof. For all $1 \leq p \leq n$ and $1 \leq q \leq k$, let s_{pq} and t_{pq} be vertices of $\mathbf{B}_{ij}$ such that there is an arc from s_{pq} to t_{pq}. Also let $S_p = \{s_{p1}, \ldots, s_{pk}\}$ and $T_p = \{t_{p1}, \ldots, t_{pk}\}$ for $1 \leq p \leq n$. Let x and y be vertices of $\mathbf{B}_{ij}$ such that (x, y) is an arc of $\mathbf{B}_{ij}$. We will start by showing that the operation $f : H^{nk+1} \longrightarrow H$

$$f(x, x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk}) = x \sqcap \mathrm{Max}(\mathrm{Min}(S_1, x), \ldots, \mathrm{Min}(S_n, x))$$

is a polymorphism of $\mathbf{B}_{ij}$, i.e., that there is an arc from $s = f(x, S_1, \ldots, S_n)$ to $t = f(y, T_1, \ldots, T_n)$.

It is easy to check that $\sqcap$ is a polymorphism of $\mathbf{B}_{ij}$ and that $\sqcup$ is a polymorphism of the induced subgraphs of $\mathbf{B}_{ij}$ with vertices $\{1, \ldots, i+1\}$ and $\{i+1, \ldots, m\}$. It is a direct consequence of the definition of $\sqcap$ that if (x, y) is an arc of $\mathbf{B}_{ij}$ then $(x \sqcap i, y \sqcap j)$ is also an arc of $\mathbf{B}_{ij}$, and that we cannot have $s = i$ and $t = j$. Hence we just need to show that $s \leq t$.

Let $\mathrm{Min}(S_l) = s_l$ and $\mathrm{Min}(T_l) = t_l$ for all $l = 1, \ldots, n$. We can then rewrite $f(x, S_1, \ldots, S_n)$ as $x \sqcap \mathrm{Max}(s_1 \sqcap x, \ldots, s_n \sqcap x)$. We have either $s_l \sqcap x \in \{1, \ldots, i+1\}$ for all $l = 1, \ldots, n$, or $s_l \sqcap x \in \{i+1, \ldots, m\}$ for all $l = 1, \ldots, n$. A similar statement holds for $t_l \sqcap y$. If $s_l \sqcap x \in \{i+1, \ldots, m\}$ then we have $t_l \sqcap y \in \{i+1, \ldots, m\}$ for all l. Since $\sqcup$ (and consequently Max) is a polymorphism of $\{i+1, \ldots, m\}$ and $\sqcap$ is a polymorphism of $\mathbf{B}_{ij}$, it follows that there is an arc from s to t. Assume now that $s_l \sqcap x \in \{1, \ldots, i+1\}$ for all $l = 1, \ldots, m$. If $t_l \sqcap y \in \{1, \ldots, i+1\}$ for all l then, as above, we can show that there is an arc from s to t. If $t_l \sqcap y \in \{i+1, \ldots, m\}$ for all l then it follows, by the definition of $\sqcup$, that $\mathrm{Max}(s_1 \sqcap x, \ldots, s_n \sqcap x) \leq \mathrm{Max}(t_1 \sqcap y, \ldots, t_n \sqcap y)$, and so (s, t) is an arc of $\mathbf{B}_{ij}$.

We now just need to show that there is an arc from $\mathrm{Min}(S'_l, x)$ to $\mathrm{Min}(T'_l, x)$ for all $l = 1, \ldots, m$. Suppose that $T'_1 \neq T_1$. If $T'_1 = T_1 \cap \{1, \ldots, i\}$ then there exists $T_2 \subseteq \{1, \ldots, i\}$. Let $T'_1 = \{t_1, \ldots, t_q\}$, we then know that there exist $s_1, \ldots, s_q \in S'_1$ such that $s_1 \leq t_1, \ldots, s_q \leq t_q$. Since $S_2 \subseteq \{1, \ldots, i\}$ we have $S'_1 \subseteq \{1, \ldots, i\}$, which implies that $\mathrm{Min}(S'_1) \leq \mathrm{Min}(s_1, \ldots, s_q)$. It follows that $\mathrm{Min}(S'_1) \leq \mathrm{Min}(s_1, \ldots, s_q) \leq \mathrm{Min}(T'_1)$. Assume now that $T'_1 = T_1 \cap \{j, \ldots, m\}$, we know that there exists $T'_2 = T_2 \subseteq \{j, \ldots, m\}$. If $S'_1 = S_1$ or $S'_1 = S_1 \cap \{1, \ldots, i\}$ then clearly $\mathrm{Min}(S'_1) \leq \mathrm{Min}(S_1) \leq \mathrm{Min}(T_1) \leq \mathrm{Min}(T'_1)$. Suppose now that $S'_1 = S_1 \cap \{j, \ldots, m\}$. We have $\mathrm{Min}(S'_1) = \max\{s : s \in S'_1\}$, and there is an element in T'_1, say t_1, such that $(\mathrm{Min}(S'_1), t_1)$ is an arc of B_{ij}. It is clear that $t_1 \leq \mathrm{Min}(T'_1)$. Thus we can conclude that in all cases $(\mathrm{Min}(S'_1, x), \mathrm{Min}(T'_1, y))$ is an arc of B_{ij}.

Suppose now that $T_l = T'_l$ for all $l = 1, \ldots, m$. Assume that $S_1 \neq S'_1$. If $S'_1 = S_1 \cap \{1, \ldots, i\}$ then $\mathrm{Min}(S'_1) = \mathrm{Min}(S_1 \cap \{1, \ldots, i\}) < \mathrm{Min}(S_1) = i+1$. It follows that $s' = h(x, S_1, \ldots, S_n) \leq s$. Since $T_l = T'_l$ for all $l = 1, \ldots, m$ we know that $t' = h(y, T_1, \ldots, T_n) = t$ then, since there is an arc from s to t', there is an arc from s' to t'. Assume now that $S'_1 = S_1 \cap \{j, \ldots, m\}$. There exists $S_2 \subseteq \{j, \ldots, m\}$, which implies that $T_2 \subseteq \{j, \ldots, m\}$, and consequently $T_1 = T'_1 \subseteq \{j, \ldots, m\}$. As above, we have $\mathrm{Min}(S'_1) = \max\{s : s \in S'_1\} \leq \max\{t : t \in T'_1\} = \mathrm{Min}(T'_1)$, and so there is an arc from $\mathrm{Min}(S'_1, x)$ to $\mathrm{Min}(T'_1, y)$. It is then easy to check that h is a polymorphism of $\mathbf{B}_{ij}$. ■

Claim 2. *The operation h is an extended k-ABS operation.*

Proof. We just need to show that it is absorptive when fixing the first component and that it satisfies the replacement property. Let $S_1, \ldots, S_n$ be sets of at most k elements each of $\mathbf{B}_{ij}$, and assume that $S_2 \subset S_1$. Given x a vertex of $\mathbf{B}_{ij}$, we will show that $h(x, S_1, \ldots, S_n) = h(x, S_2, S_2, S_3, \ldots, S_n)$.

If $\mathrm{Min}(S_1') \in \{1, \ldots, i\}$, $\mathrm{Min}(S_2', x) \in \{i+1, \ldots, m\}$ and $x \in \{1, \ldots, i\}$, then there exists $S_3 = S_3' \subseteq \{1, \ldots, i\}$, and we have

$$\begin{aligned}\mathrm{Min}(S_1', x) \sqcup \mathrm{Min}(S_2', x) \sqcup \mathrm{Min}(S_3, x) &= \mathrm{Min}(S_1', x) \sqcup i+1 \sqcup \mathrm{Min}(S_3, x)\\ &= i = i+1 \sqcup \mathrm{Min}(S_3, x)\\ &= \mathrm{Min}(S_2', x) \sqcup \mathrm{Min}(S_2', x) \sqcup \mathrm{Min}(S_3, x)\end{aligned}$$

since $\mathrm{Min}(S_3, x) \in \{1, \ldots, i\}$.

In all other cases (regarding the subsets of B_{ij} that $\mathrm{Min}(S_2', x)$ and $\mathrm{Min}(S_1', x)$ belong to) we can easily see, using the definitions of $\sqcup$ and $\sqcap$ that $\mathrm{Min}(S_1', x) \sqcup \mathrm{Min}(S_2', x) = \mathrm{Min}(S_2', x)$.

This proves that $h(x, S_1, \ldots, S_n) = h(x, S_2, S_2, S_3, \ldots, S_n)$.

Let us now check that it satisfies the replacement property. Let $y \in S_1 \cap S_2 \cap \ldots \cap S_n$ be arbitrary. Since this intersection is non-empty it is clear that $y \in S_1' \cap S_2' \cap \ldots \cap S_n'$, and that either all sets intersect $\{i+1, \ldots, m\}$ or are contained in $\{1, \ldots, i\}$. By considering these two cases and the set that x belongs to, $\{1, \ldots, i\}$ or $\{i+1, \ldots, m\}$, we can show that either $h(x, S_1, \ldots, S_n) = i+1 = h(y, S_1 \cup \{x\}, \ldots, S_n \cup \{x\})$, or $\mathrm{Min}(S_l, x) = \mathrm{Min}(S_l, x, y) \leq x, y$ for all l, in which case

$$\begin{aligned}h(x, S_1, \ldots, S_n) &= x \sqcap \mathrm{Max}(\mathrm{Min}(S_1', x), \ldots, \mathrm{Min}(S_n', x))\\ &= \mathrm{Max}(\mathrm{Min}(S_1', x), \ldots, \mathrm{Min}(S_n', x))\\ &= \mathrm{Max}(\mathrm{Min}(S_1', x, y), \ldots, \mathrm{Min}(S_n', x, y))\\ &= y \sqcap \mathrm{Max}(\mathrm{Min}(S_1', x, y), \ldots, \mathrm{Min}(S_n', x, y))\\ &= h(y, S_1 \cup \{x\}, \ldots, S_n \cup \{x\}).\end{aligned}$$

Thus h satisfies the replacement property.

■

Acknowledgments

The authors would like to thank the anonymous referee for valuable comments. The first author is supported by FCT grant SFRH/BPD/26216/2006 and also by FCT and FEDER within the project ISFL-1-143 of the Centre for Algebra of the University of Lisbon. The second author is supported by grants TIN2006-15387-C03-03 and TIN2007-68005-C04 funded by the MCyT. The third author is supported by the UK EPSRC grants EP/G011001/1 and EP/C543831/1.

References

[1] A. Atserias. On digraph coloring problems and treewidth duality. *European Journal of Combinatorics*, 29(4):796–820, 2008.

[2] L. Barto and M. Kozik. Constraint satisfaction problems of bounded width. In *FOCS'09*, pages 595–603, 2009.

[3] L. Barto, M. Kozik, and T. Niven. Graphs, polymorphisms and the complexity of homomorphism problems. In *STOC'08*, pages 789–796, 2008.

[4] A. Bulatov. H-coloring dichotomy revisited. *Theoretical Computer Science*, 349(1):31–39, 2005.

[5] A. Bulatov. Bounded relational width, 2009. manuscript.

[6] A. Bulatov, A. Krokhin, and B. Larose. Dualities for constraint satisfaction problems. In *Complexity of Constraints*, volume 5250 of *LNCS*, pages 93–124. 2008.

[7] D. Cohen and P. Jeavons. The complexity of constraint languages. In F. Rossi, P. van Beek, and T. Walsh, editors, *Handbook of Constraint Programming*, chapter 8. Elsevier, 2006.

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

[9] V. Dalmau. Linear Datalog and bounded path duality for relational structures. *Logical Methods in Computer Science*, 1(1), 2005. (electronic).

[10] V. Dalmau and A. Krokhin. Majority constraints have bounded pathwidth duality. *European Journal of Combinatorics*, 29(4):821–837, 2008.

[11] V. Dalmau and J. Pearson. Set functions and width 1 problems. In *CP'99*, volume 1713 of *LNCS*, pages 159–173, 1999.

[12] H-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1999.

[13] L. Egri, B. Larose, and P. Tesson. Symmetric Datalog and constraint satisfaction problems in Logspace. In *LICS'07*, pages 193–202, 2007.

[14] T. Feder and P. Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory, Ser.B*, 72:236–250, 1998.

[15] T. Feder, P. Hell, and J. Huang. List homomorphisms and retractions to reflexive digraphs. manuscript, 2008.

[16] T. Feder and M.Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28:57–104, 1998.

[17] G. Grätzer. *General Lattice Theory*. Birkhäuser, 2nd edition, 2002.

[18] M. Grohe. The structure of tractable constraint satisfaction problems. In *MFCS'06*, volume 4162 of *LNCS*, pages 58–72, 2006.

[19] A. Gupta, P. Hell, M. Karimi, and A. Rafiey. Minimum cost homomorphisms to reflexive digraphs. In *LATIN'08*, volume 4957 of *LNCS*, pages 182–193, 2008.

[20] P. Hell and J. Nešetřil. *Graphs and Homomorphisms.* Oxford University Press, 2004.

[21] P. Hell, J. Nešetřil, and X. Zhu. Duality of graph homomorphisms. In *Combinatorics, Paul Erdös is Eighty (Vol.2)*, volume 2 of *Bolyai Soc. Math. Stud.*, pages 271–282. János Bolyai Math. Soc., 1996.

[22] P. Hell, H. Zhou, and X. Zhu. On homomorphisms to acyclic local tournaments. *Journal of Graph Theory*, 20(4):467–471, 1995.

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

[24] P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. Tractability and learnability arising from algebras with few subpowers. In *LICS'07*, pages 213–222, 2007.

[25] Ph.G. Kolaitis. On the expressive power of logics on finite models. In *Finite Model Theory and its Applications*, EATCS Series: Texts in Theoretical Computer Science, pages 27–124. Springer, 2007.

[26] Ph.G. Kolaitis and M.Y. Vardi. A logical approach to constraint satisfaction. In *Finite Model Theory and its Applications*, EATCS Series: Texts in Theoretical Computer Science, pages 339–370. Springer, 2007.

[27] A. Krokhin, A. Bulatov, and P. Jeavons. The complexity of constraint satisfaction: an algebraic approach. In *Structural Theory of Automata, Semigroups, and Universal Algebra*, volume 207 of *NATO Science Series II: Math., Phys., Chem.*, pages 181–213. Springer Verlag, 2005.

[28] B. Larose, C. Loten, and C. Tardif. A characterisation of first-order constraint satisfaction problems. *Logical Methods in Computer Science*, 3(4), 2007. (electronic).

[29] B. Larose and P. Tesson. Universal algebra and hardness results for constraint satisfaction problems. In *ICALP'07*, volume 4596 of *LNCS*, pages 267–278, 2007.

[30] C. Loten and C. Tardif. Majority functions on structures with finite duality. *European Journal of Combinatorics*, 29(4):979–986, 2008.

[31] J. Nešetřil and V. Rödl. Chromatically optimal rigid graphs. *Journal of Combinatorial Theory, Ser.B*, 46:133–141, 1989.

[32] J. Nešetřil and C. Tardif. Duality theorems for finite structures (characterising gaps and good characterisations). *Journal of Combinatorial Theory, Ser.B*, 80:80–97, 2000.

[33] J. Nešetřil and C. Tardif. Short answers to exponentially long questions: extremal aspects of homomorphism duality. *SIAM Journal on Discrete Mathematics*, 19(4):914–920, 2005.

[34] L. Zádori. Relational sets and categorical equivalence for algebras. *International Journal of Algebra and Computation*, 7(5):561–576, 1997.