

Almost Symmetric Linear Arc Monadic Datalog and Transitive Tournaments

Sebastian Meyer¹, Florian Starke¹

¹TU Dresden

Abstract

We introduce *n*-almost symmetric Datalog and study *n*-almost symmetric linear arc monadic Datalog. We characterize the finite relational structures whose constraint satisfaction problem is solved by this Datalog fragment as those that can be primitive positively constructed from the transitive tournament on $n + 2$ vertices. We also give characterizations in terms of a certain homomorphism duality, which we call *n*-fixed unfolded caterpillar duality, and in universal-algebraic terms, namely the existence of *k*-absorptive operations and operations forming an *elevator chain of length* $n + 1$. This article generalizes the results from Bodirsky and Starke about symmetric linear arc monadic Datalog.

Keywords

Datalog fragments, constraint satisfaction problems, Hagemann-Mitschke chains, caterpillar duality

1. Introduction

Datalog fragments have been used a lot to study constraint satisfaction problems (CSPs) that are solvable in polynomial time. Well known fragments are *linear Datalog*, *symmetric linear Datalog*, and *arc monadic Datalog*, better known as *arc consistency*. In 2011 Carvalho, Dalmau, and Krokhin combined two of those restrictions and characterized the CSPs solved by linear arc monadic (lam) Datalog [1]. Recently, Bodirsky and Starke used these results to study the CSPs solved by symmetric linear arc monadic (slam) Datalog [2]. One of the characterizations of these CSPs uses *primitive positive (pp-)constructions*. These constructions give rise to a quasi order on finite relational structures which respects the computational complexity of the corresponding CSPs. The characterizations state that the CSP of a finite relational structure is solved by lam Datalog if and only if the structure can be pp-constructed by st-Con and it is solved by slam Datalog if and only if the structure is pp-constructed by $\mathbb{T}_2$, the transitive tournament on two elements. It is well known that the transitive tournaments ordered by pp-constructability form the following chain

$$\text{st-Con} < \dots < \mathbb{T}_4 < \mathbb{T}_3 < \mathbb{T}_2 < \mathbb{T}_1.$$

In this article we introduce *n*-almost symmetric Datalog, a Datalog fragment where non-symmetric rules are allowed but every derivation can use them at most n times. With the help of higher arity IDBs we can show that *n*-almost symmetric Datalog programs can solve the same problems as symmetric Datalog programs. Hence, *n*-almost symmetric Datalog is actually already well studied. When restricting to unary IDBs “hiding” the non-symmetric rules does no longer work and in fact we do obtain new interesting Datalog fragments. In this article we study *n*-almost symmetric linear arc monadic (s_n lam) Datalog and show that the CSP of a finite relational structure is solved by s_n lam Datalog if and only if the structure can be pp-constructed by $\mathbb{T}_{n+2}$, generalizing the results from Bodirsky and Starke about slam Datalog. We also give characterizations in terms of a certain homomorphism duality, which we call *n*-fixed unfolded caterpillar duality, and in universal-algebraic terms, namely the existence of *k*-absorptive operations and operations forming an *elevator chain of length* $n + 1$.

Datalog 2.0 2026, September 07, 2026, Klagenfurt, Austria

✉ sebastian.meyer2@tu-dresden.de (S. Meyer); florian.starke@tu-dresden.de (F. Starke)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Preliminaries

We write $[n]$ for the set $\{1, \dots, n\}$ and $[m, n]$ for the set $\{m, m+1, \dots, n\}$. We say that a tuple $t \in A^k$, for $k \in \mathbb{N}$, is *injective* if t is injective when viewed as a function from $[k]$ to A .

2.1. Structures, Homomorphisms, and CSPs

We assume familiarity with the concepts of relational structures and first-order formulas from mathematical logic, as introduced for instance in [3]. All structures we consider will have a finite domain and a relational signature. Let τ be a relational signature and $\mathbb{B}$ be a τ -structure. We say that $\mathbb{B}$ is *injective* if for all relation symbols R of τ all tuples in $R^{\mathbb{B}}$ are injective. The structure $\mathbb{B}$ is *connected* if for all $b, b' \in B$ there are relation symbols $R_1, \dots, R_n \in \tau$ and tuples $t_1 \in R_1^{\mathbb{B}}, \dots, t_n \in R_n^{\mathbb{B}}$ such that b is an element of t_1 , b' is an element of t_n , and t_i and t_{i+1} have at least one element in common for all $i \in [n-1]$. Let $\mathbb{A}$ be another τ -structure. A *homomorphism* from $\mathbb{A} \rightarrow \mathbb{B}$ is a map $h: A \rightarrow B$ such that for all $R \in \tau$ we have $h(R^{\mathbb{A}}) \subseteq R^{\mathbb{B}}$. An *endomorphism* is a homomorphism from a structure to itself. We write $\mathbb{A} \rightarrow \mathbb{B}$ if there exists a homomorphism from $\mathbb{A}$ to $\mathbb{B}$ and $\mathbb{A} \not\rightarrow \mathbb{B}$ if there does not exist one. For $n \geq 1$ a *polymorphism of $\mathbb{B}$ of arity n* is a homomorphism from $\mathbb{B}^n$ to $\mathbb{B}$. We denote the set of all polymorphisms of $\mathbb{B}$ by $\text{Pol}(\mathbb{B})$. The structures $\mathbb{A}$ and $\mathbb{B}$ are called *homomorphically equivalent* if $\mathbb{A} \rightarrow \mathbb{B}$ and $\mathbb{B} \rightarrow \mathbb{A}$. The *constraint satisfaction problem of $\mathbb{B}$* , denoted $\text{CSP}(\mathbb{B})$, is the set $\{\mathbb{A} \mid \mathbb{A} \text{ is a finite } \tau\text{-structure and } \mathbb{A} \rightarrow \mathbb{B}\}$. For example let st-Con be the $\{\text{source}, \text{target}, E\}$ -structure with domain $\{0, 1\}$ and relations $\text{source}^{\text{st-Con}} := \{1\}$, $\text{target}^{\text{st-Con}} := \{0\}$, and $E^{\text{st-Con}} := \{(0, 0), (0, 1), (1, 1)\} = \leq$. Then $\text{CSP}(\text{st-Con})$ consists of all finite directed graphs that have some vertices labeled with target and some vertices labeled with source such that there is no directed path from a source vertex to a target vertex. Hence, $\text{CSP}(\text{st-Con})$ encodes the complement of the directed reachability problem.

2.2. Primitive Positive Formulas

A τ -formula is called *primitive positive (pp)* if it only uses atomic formulas ($x = y$ is allowed), conjunction, and existential quantification. There is an easy way to translate between finite τ -structures and pp- τ -formulas. Let $\mathbb{B}$ be a τ -structure, the *canonical conjunctive query of $\mathbb{B}$* is the pp-formula with variables B that is the conjunct of all formulas $R(b_1, \dots, b_m)$ for $R \in \tau$ and $(b_1, \dots, b_m) \in R^{\mathbb{B}}$. For a pp- τ -formula ϕ let ϕ' be an equivalent pp- τ -formula that is of the form $\exists x_1, \dots, x_n \bigwedge \text{atomic formula}$. Construct a τ -structure in the following way. Take all variables (free and quantified) of ϕ' as the domain, a tuple $(y_1, \dots, y_m)$ is in relation R if ϕ' contains the conjunct $R(y_1, \dots, y_m)$. Finally, identify all elements x, y for which ϕ' contains the conjunct $x = y$ or $y = x$. The resulting structure is called the *canonical database of ϕ* and unique up to isomorphism.

2.3. Minions and Minor Conditions

Let $\mathbb{B}$ be τ -structure. A *clone* is a set of operations, i.e., maps from B^n to B , that contains all projections and is closed under composition. Note that $\text{Pol}(\mathbb{B})$ is a clone. Let $\mathbb{A}$ be another τ -structure. A *polymorphism of $(\mathbb{A}, \mathbb{B})$ of arity n* is a homomorphism from $\mathbb{A}^n$ to $\mathbb{B}$. By $\text{Pol}(\mathbb{A}, \mathbb{B})$ we denote the set of all polymorphisms of $(\mathbb{A}, \mathbb{B})$. Note that $\text{Pol}(\mathbb{B}, \mathbb{B}) = \text{Pol}(\mathbb{B})$. Let $\sigma: [n] \rightarrow [m]$ and $t = (a_1, \dots, a_m) \in A^m$, then $t_\sigma := (a_{\sigma(1)}, \dots, a_{\sigma(n)}) \in A^n$. For a map $f: A^n \rightarrow B$ the *minor of f given by σ* is the map $f_\sigma: A^m \rightarrow B, t \mapsto f(t_\sigma)$. A *Minion* is a set of functions from A^n to B that is closed under taking minors. Note that $\text{Pol}(\mathbb{A}, \mathbb{B})$ is a minion. Let $\mathcal{M}$ and $\mathcal{N}$ be minions. A map $\xi: \mathcal{M} \rightarrow \mathcal{N}$ is a *minion homomorphism* if ξ preserves the arity of the operations and it preserves minors, i.e., for all $f \in \mathcal{M}$ of arity n and all $\sigma: [n] \rightarrow [m]$ we have $\xi(f_\sigma) = (\xi(f))_\sigma$. In this article we will be only concerned with minions of the form $\text{Pol}(\mathbb{B})$. However, seeing that these clones are special minions makes it more plausible that we care about minion homomorphism instead of clone homomorphisms. For example the n -ary minimum and maximum operations on $\{0, 1\}$ are polymorphisms of st-Con .

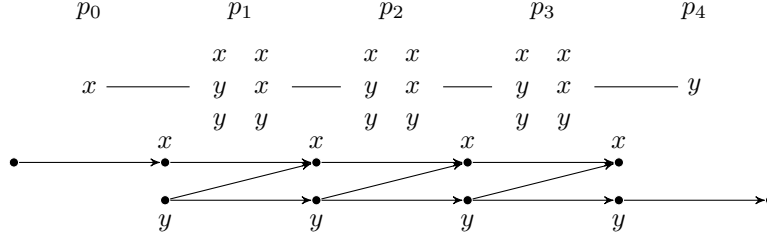


Figure 1: A visualization of HM(3) (top) and a 3-braid (bottom), which is another way to visualize HM(3).

Observe that the clone $\text{Pol}(\text{st-Con})$ consists of all operations $f: \{0, 1\}^n \rightarrow \{0, 1\}$ with $f(0, \dots, 0) = 0$, $f(1, \dots, 1) = 1$, and $s \leq t$ implies $f(s) \leq f(t)$ for all $s, t \in \{0, 1\}^n$.

Let τ be a set of function symbols each equipped with an arity. A *minor condition* is a finite set of *height-one identities*, i.e., expressions of the form

$$f(x_1, \dots, x_n) \approx g(y_1, \dots, y_m)$$

where $f, g \in \tau$ of arities n and m , respectively, and $x_1, \dots, x_n, y_1, \dots, y_m$ are not necessarily distinct variables. A minion $\mathcal{M}$ *satisfies* a minor condition Σ , denoted $\mathcal{M} \models \Sigma$, if there is an assignment $\tau \rightarrow \mathcal{M}$, $f \mapsto f^{\mathcal{M}}$ that preserves the arity and for each height-one identity $f(x_1, \dots, x_n) \approx g(y_1, \dots, y_m)$ in Σ and for each assignment $a: \{x_1, \dots, x_n, y_1, \dots, y_m\} \rightarrow B$ we have

$$f^{\mathcal{M}}(a(x_1), \dots, a(x_n)) = g^{\mathcal{M}}(a(y_1), \dots, a(y_m)).$$

Let Σ and Σ' be minor conditions. We say that Σ *implies (for finite clones)* Σ' if every clone over a finite domain that satisfies Σ also satisfies Σ' . Similarly we define implication for minions. It is easy to see that minion homomorphisms compose to minion homomorphisms, hence we can quasi-order finite structures via their polymorphism clones, i.e., $\mathbb{A} \leq \mathbb{B}$ if there is a minion homomorphism from $\text{Pol}(\mathbb{A})$ to $\text{Pol}(\mathbb{B})$. This poset has been extensively studied [4, 5, 6, 7]. There is the following well known connection between minion homomorphisms, minor conditions, and primitive positive constructions (we will not introduce those here, for a definition see for example [8]).

Theorem 2.1. [Theorem 4.12 in [9]] *Let $\mathbb{A}$ and $\mathbb{B}$ be finite τ -structures. Then the following are equivalent.*

1. *There is a minion homomorphism from $\text{Pol}(\mathbb{B})$ to $\text{Pol}(\mathbb{A})$.*
2. *Every minor condition that holds in $\text{Pol}(\mathbb{B})$ also holds in $\text{Pol}(\mathbb{A})$.*
3. *There is a pp-construction of $\mathbb{A}$ in $\mathbb{B}$.*

The following family of minor conditions is central to the article.

Definition 2.2. For $n \geq 1$, the *quasi Hagemann-Mitschke condition of length n* , called HM(n), consists of the identities

$$\begin{aligned} p_0(x) &\approx p_1(x, y, y), \\ p_i(x, x, y) &\approx p_{i+1}(x, y, y) && \text{for all } i \in \{1, \dots, n-1\}, \text{ and} \\ p_n(x, x, y) &\approx p_{n+1}(y). \end{aligned}$$

For $n = 0$ we define HM(0) to be the condition consisting of the identity $p_0(x) \approx p_1(y)$.

Note that the polymorphism clone of a structure $\mathbb{B}$ satisfies HM(0) if and only if $\mathbb{B}$ has a constant polymorphism. In Figure 1 there is a visualization of HM(3) that makes it clear why $p_0(x) \approx p_1(y)$ is a natural choice for HM(0).

2.4. Caterpillars

The notion of caterpillars was generalized from graphs to relational structures by Carvalho, Dalmau, and Krokhin in [1]. We will use an equivalent definition here that makes it easier to transform caterpillars.

Definition 2.3. An injective τ -structure $\mathbb{C}$ is a *caterpillar* if there is a tuple $(a_0, \dots, a_n)$ of distinct elements of $\mathbb{C}$ and a pp-formula $(\phi \wedge \psi)(a_0, \dots, a_n)$ such that

- $\mathbb{C}$ is isomorphic to the canonical database of $\phi \wedge \psi$,
- the formula ϕ is of the form

$$\phi_0(a_0, a_1) \wedge \dots \wedge \phi_{n-1}(a_{n-1}, a_n),$$

where each $\phi_i(a_i, a_{i+1})$ is obtained from an atomic formula, where a_i and a_{i+1} occur, by existentially quantifying all variables except for a_i and a_{i+1} ,

- the formula ψ is of the form

$$\psi_0(a_0) \wedge \dots \wedge \psi_n(a_n),$$

where each $\psi_i(a_i)$ is a (possibly empty) conjunct of formulas obtained from an atomic formula, where a_i occurs, by existentially quantifying all variables except for a_i , and

- if $\mathbb{C}$ is not an isolated point, then neither ψ_0 nor ψ_n are the empty conjunct.

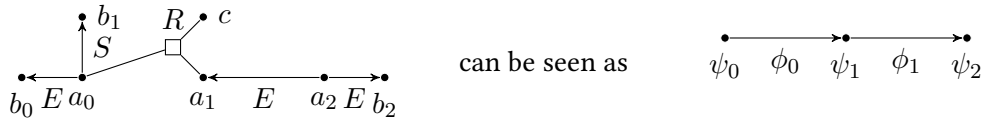
The elements $a_0, \dots, a_n$ are called *joints* of $\mathbb{C}$, the tuples (a_k, a_{k+1}) are called *vertebra*, and the tuple $(a_0, \dots, a_n)$ is called the *spine* of $\mathbb{C}$. A vertebra (a_k, a_{k+1}) lies *between* a_i and a_j if $i \leq k$ and $k+1 \leq j$. The conjuncts of $\psi_i(a_i)$ are called *hairs* of a_i . If ψ_i is the empty conjunct we say that a_i has *no hairs*. The formula $\phi \wedge \psi$ is the *conjunctive query representation* of $\mathbb{C}$.

Note that for every caterpillar with at least one vertebra the spine and the conjunctive query representation are unique up to orientation, hence using the definite article is still justified.

Example 2.4. The structure $\mathbb{C}$ with domain $\{a_0, a_1, a_2, b_0, b_1, b_2, c\}$ and relations $R^{\mathbb{C}} = \{(a_0, a_1, c)\}$, $E^{\mathbb{C}} = \{(a_2, a_1), (a_0, b_0), (b_2, a_2)\}$, $S^{\mathbb{C}} = \{(a_0, b_1)\}$ is a caterpillar with spine (a_0, a_1, a_2) . Its conjunctive query representation consists of the formulas

$$\begin{aligned} \phi_0(a_0, a_1) &:= \exists x R(a_0, a_1, x), & \phi_1(a_1, a_2) &:= E(a_2, a_1), \\ \psi_0(a_0) &:= \exists x E(a_0, x) \wedge \exists x S(a_0, x), & \psi_2(a_2) &:= \exists x E(x, a_2), \text{ and} \end{aligned}$$

$\psi_1(a_1)$ is the empty conjunct. The caterpillar $\mathbb{C}$



Note that (a_2, a_1, a_0) is also the spine of $\mathbb{C}$. △

Definition 2.5. Let $\mathbb{C}$ be a caterpillar with spine $(a_0, \dots, a_n)$ and conjunctive query representation $(\phi \wedge \psi)(a_0, \dots, a_n)$. Let $i, j \in [0, n]$ with $i < j$ and define

$$\begin{aligned} \phi_{\text{head}}(a_0, \dots, a_i) &:= \phi_0(a_0, a_1) \wedge \dots \wedge \phi_{i-1}(a_{i-1}, a_i), \\ \phi_{\text{body}}(a_i, \dots, a_j) &:= \phi_i(a_i, a_{i+1}) \wedge \dots \wedge \phi_{j-1}(a_{j-1}, a_j), \\ \phi_{\text{tail}}(a_j, \dots, a_n) &:= \phi_j(a_j, a_{j+1}) \wedge \dots \wedge \phi_{n-1}(a_{n-1}, a_n), \\ \psi_{\text{head}}(a_0, \dots, a_i) &:= \psi_0(a_0) \wedge \dots \wedge \psi_i(a_i), \end{aligned}$$

$$\begin{aligned}\psi_{\text{body}}(a_{i+1}, \dots, a_{j-1}) &:= \psi_{i+1}(a_{i+1}) \wedge \dots \wedge \psi_{j-1}(a_{j-1}), \text{ and} \\ \psi_{\text{tail}}(a_j, \dots, a_n) &:= \psi_j(a_j) \wedge \dots \wedge \psi_n(a_n).\end{aligned}$$

The (a_i, a_j) -*unfolding* of $\mathbb{C}$ is the canonical database $\mathbb{C}'$ of the pp-formula $\phi' \wedge \psi'$ with

$$\begin{aligned}\phi' &:= \phi_{\text{head}}(a_0, \dots, a_i) \wedge \phi_{\text{body}}(a_i, a'_{i+1}, \dots, a'_{j-1}, a'_j) \wedge \\ &\quad \phi_{\text{body}}(a''_i, a_{i+1}, \dots, a_{j-1}, a'_j) \wedge \\ &\quad \phi_{\text{body}}(a''_i, a''_{i+1}, \dots, a''_{j-1}, a_j) \wedge \phi_{\text{tail}}(a_j, \dots, a_n)\end{aligned}$$

and

$$\begin{aligned}\psi' &:= \psi_{\text{head}}(a_0, \dots, a_i) \wedge \psi_{\text{body}}(a'_{i+1}, \dots, a'_{j-1}) \wedge \\ &\quad \psi_{\text{body}}(a_{i+1}, \dots, a_{j-1}) \wedge \\ &\quad \psi_{\text{body}}(a''_{i+1}, \dots, a''_{j-1}) \wedge \psi_{\text{tail}}(a_j, \dots, a_n).\end{aligned}$$

It is easy to see that $\mathbb{C}'$ is a caterpillar and it has the spine

$$(a_0, \dots, a_i, a'_{i+1}, \dots, a'_{j-1}, a'_j, a_{j-1}, \dots, a_{i+1}, a''_i, a''_{i+1}, \dots, a''_{j-1}, a_j, \dots, a_n).$$

Observe that a'_j and a''_i have no hairs and that for $k \in \{0, \dots, i-1, j, \dots, n-1\}$ the caterpillar $\mathbb{C}'$ also has the vertebra (a_k, a_{k+1}) , the (a_i, a_j) -unfolding of $\mathbb{C}$ is said to *fix* the vertebra (a_k, a_{k+1}) . We will assume that for an element c that is in the head or the tail of $\mathbb{C}$, the corresponding element in $\mathbb{C}'$ is also called c (the elements a_i, a_j , and the elements in their hairs also belong to the body and tail, respectively). For any element c in the body of $\mathbb{C}$ there are three corresponding elements in $\mathbb{C}'$, our convention will be that the first one is called c' , the second c , and the third c'' . Note that this convention extends the naming convention we used for the joints. This way we obtain a surjective homomorphism $h: \mathbb{C}' \rightarrow \mathbb{C}$ that is the identity on the tail and the head and sends the elements c, c' and c'' from the body to c .

An *unfolding* $\mathbb{C}'$ of $\mathbb{C}$ that fixes $v_1, \dots, v_m$ is a structure that is obtained from $\mathbb{C}$ by repeatedly applying (a, b) -unfoldings that fix $v_1, \dots, v_m$. This is well defined, since whenever an (a, b) -unfolding fixes some vertebra v_i , then the unfolded caterpillar still has the vertebra v_i . Note that, by composing the previously constructed homomorphisms, we obtain a surjective homomorphism $h: \mathbb{C}' \rightarrow \mathbb{C}$ that is the identity on all joints in $v_1, \dots, v_m$.

2.5. Datalog Fragments

For a detailed introduction, see, e.g., [10]. Let τ and ρ be finite relational signatures such that $\tau \subseteq \rho$. A *Datalog program* Π is a finite set of rules of the form $\phi_0 := \phi_1 \wedge \dots \wedge \phi_n$, where each ϕ_i is an atomic ρ -formula. The formula ϕ_0 is called the *head* of the rule, and $\phi_1 \wedge \dots \wedge \phi_n$ is called the *body* of the rule. The symbols in τ are called *EDBs* (*extensional database predicates*) and the other symbols from $\rho \setminus \tau$ are called *IDBs* (*intensional database predicates*). In the rule heads, only IDBs are allowed. There is one special IDB G of arity 0, which is called the *goal predicate*. IDBs might also appear in the rule bodies. A *derivation* of Π on a finite τ -structure $\mathbb{A}$ is a sequence $\mathbb{A}_0 \vdash_{R_0} \dots \vdash_{R_{n-1}} \mathbb{A}_n$ of ρ -structures $\mathbb{A}_0, \dots, \mathbb{A}_n$ and rules $R_0, \dots, R_{n-1}$ of Π such that

- the τ -reduct of $\mathbb{A}_0$ is $\mathbb{A}$ and $P^{\mathbb{A}_0} = \emptyset$ for all $P \in \rho \setminus \tau$ and
- for all $i \in [0, n-1]$ the structure $\mathbb{A}_{i+1}$ is obtained from $\mathbb{A}_i$ by adding a tuple $(s(x_1), \dots, s(x_k))$ to $P^{\mathbb{A}_i}$, where R_i is $P(x_1, \dots, x_k) := \phi$ and s is a satisfying assignment of the variables of $Q(y_1, \dots, y_\ell) \wedge \phi$ to elements of $\mathbb{A}_i$.

We say that this derivation *derives the IDB P on the tuple t* if $\mathbb{A}_n$ is obtained from $\mathbb{A}_{n-1}$ by adding t to $P^{\mathbb{A}_{n-1}}$. We say that this derivation *derives the goal predicate (on $\mathbb{A}$)* if it derives G on the empty tuple $()$,

i.e., if $G^{A_n} \neq \emptyset$. If $\mathbb{B}$ is a τ -structure, then we say that $\text{CSP}(\mathbb{B})$ is *solved* by Π if the following holds: the goal predicate can be derived by Π on a finite τ -structure $\mathbb{A}$ if and only if there is *no* homomorphism from $\mathbb{A}$ to $\mathbb{B}$. A Datalog program is called

- *linear* if in each rule, at most one IDB appears in the body (we then assume without loss of generality that in every rule whose body contains an IDB, the IDB is listed first).
- *arc* if each rule involves at most one EDB.
- *symmetric* if it is linear and for every rule $\phi_0 :- \phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_n$ where ϕ_0 is not the goal predicate and ϕ_1 is an atomic formula whose relation symbol is an IDB, the Datalog program also contains the *reversed rule* $\phi_1 :- \phi_0 \wedge \phi_2 \wedge \dots \wedge \phi_n$.¹
- *n-almost symmetric* if it is linear and there is a quasi-order $\leq$ on the IDBs without the goal predicate such that
 - every strict chain $P_1 < \dots < P_m$ of IDBs has length $m \leq n + 1$ and
 - for every rule $\phi_0 :- \phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_n$ that does not contain the goal predicate, where ϕ_0 and ϕ_1 are atomic formulas whose relation symbols are the IDBs P_0 and P_1 , respectively, we have $P_0 \leq P_1$, if $P_1 \leq P_0$, then the Datalog program also contains the reversed rule, and if the Datalog program does not contain the reversed rule, then $P_0 < P_1$.

Note that any derivation of an n -almost symmetric Datalog program can use non-symmetric rules at most n times. In particular, a 0-almost symmetric Datalog program is symmetric.

We say that a Datalog program has *width* (ℓ, k) if all IDBs have arity at most ℓ , and if every rule has at most k variables. For given (ℓ, k) and a structure $\mathbb{B}$, there exists a Datalog program Π of width (ℓ, k) with the remarkable property that if some Datalog program of width (ℓ, k) solves $\text{CSP}(\mathbb{B})$, then Π solves $\text{CSP}(\mathbb{B})$. This Datalog program is referred to as the *canonical Datalog program for $\mathbb{B}$ of width (ℓ, k)* , and is constructed as follows [11]: For every relation R over B of arity at most ℓ , we introduce a new IDB. The empty relation of arity 0 plays the role of the goal predicate. Then Π contains all rules $\phi_0 :- \phi_1 \wedge \dots \wedge \phi_n$ with at most k variables such that the formula $\forall \bar{x} (\phi_1 \wedge \dots \wedge \phi_n \Rightarrow \phi_0)$ holds in the expansion of $\mathbb{B}$ by all IDBs. Note that if the canonical Datalog program Π for $\mathbb{B}$ does not solve $\text{CSP}(\mathbb{B})$, then it is still true that if Π derives the goal predicate on a finite structure $\mathbb{A}$, then there is no homomorphism from $\mathbb{A}$ to $\mathbb{B}$, (see, e.g., [10]).

If k is the maximal arity of the EDBs, we may restrict the canonical Datalog program of width $(1, k)$ to those rules with only unary IDBs and at most one EDB; in this case, we obtain the canonical arc monadic Datalog program, which is also known as the *arc consistency procedure*. Analogously, we may define the canonical *linear*, and the canonical *symmetric* Datalog program. We may also combine these restrictions, and in particular obtain a definition the *canonical slam Datalog program*, i.e., the canonical symmetric linear arc monadic Datalog program.

Definition 2.6. Let $\mathbb{B}$ be a finite structure with a finite relational signature, the *canonical s_n lam Datalog program for $\mathbb{B}$* is obtained by taking $n + 1$ many copies $\Pi_0, \dots, \Pi_n$ of the canonical slam Datalog program and adding for each rule $\phi_0 :- \phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_n$ of the canonical lam Datalog program, where ϕ_0 and ϕ_1 are atomic formulas whose relation symbols are the IDBs P_0 and P_1 , respectively, the rules $\phi_0^i :- \phi_1^{i+1} \wedge \phi_2 \wedge \dots \wedge \phi_n$ for $i \in [0, n - 1]$, where ϕ_0^i is obtained from ϕ_0 by replacing P_0 by its copy from Π_i and ϕ_1^{i+1} is obtained from ϕ_1 by replacing P_1 by its copy from Π_{i+1} . Clearly, the resulting program is n -almost symmetric, linear, arc, and monadic.

Note that the canonical s_0 lam Datalog program is the canonical slam Datalog program. The following lemma can be shown analogously to the well-known fact for unrestricted canonical Datalog programs of width (ℓ, k) (see, e.g., [10]).

¹Note that excluding the case that ϕ_0 is the goal predicate is not essential, because we may always add its symmetric version without changing the set of structures on which the goal predicate is derived. The advantage of this version is that symmetric Datalog programs do not need to have rules with the goal predicate in their body.

Lemma 2.7. *Let $\mathbb{B}$ be a finite structure with a finite relational signature, $n \in \mathbb{N}$, and let Π be the canonical s_n lam Datalog program for $\mathbb{B}$. If $\mathbb{A}$ is a finite structure with a homomorphism to $\mathbb{B}$, then Π does not derive the goal predicate on $\mathbb{A}$.*

2.6. Dualities

Let $\mathbb{B}$ be a τ -structure and $\mathcal{F}$ be a set of τ -structures. The tuple $(\mathcal{F}, \mathbb{B})$ is a *duality pair* if for all structures $\mathbb{A}$ with $\mathbb{A} \not\rightarrow \mathbb{B}$ there is a $\mathbb{F} \in \mathcal{F}$ with $\mathbb{F} \rightarrow \mathbb{A}$. The set $\mathcal{F}$ is called an *obstruction set* of $\mathbb{B}$ and its elements *obstructions*. We can now describe structures by properties of their obstruction sets. The most commonly known are *finite duality*, i.e. there is a finite obstruction set, and *tree duality*, i.e., there is an obstruction set consisting of trees. In 2011 Carvalho, Dalmau, and Krokhn introduced *caterpillar duality*, i.e., there is an obstruction set consisting of caterpillars, and characterized when a structure does have caterpillar duality. This characterization was later extended by Bodirsky and Starke by the items (5)-(7) involving the structure st-Con introduced in Section 2.1.

Theorem 2.8. *[Theorem 16 in [1] and Remark 5.1 in [2]] Let $\mathbb{B}$ be a finite relational τ -structure. Then the following are equivalent.*

1. $\mathbb{B}$ has caterpillar duality.
2. $\text{CSP}(\mathbb{B})$ can be solved by a linear arc monadic Datalog program.
3. $\text{Pol}(\mathbb{B})$ contains for every $k, n \geq 1$ an k -absorbing operation of arity kn .
4. $\mathbb{B}$ is homomorphically equivalent to a structure $\mathbb{B}'$ with binary polymorphisms $\sqcup$ and $\sqcap$ such that $(\mathbb{B}', \sqcup, \sqcap)$ is a (distributive) lattice.
5. Every minor condition that holds in $\text{Pol}(\text{st-Con})$ also holds in $\text{Pol}(\mathbb{B})$.
6. There is a minion homomorphism from $\text{Pol}(\text{st-Con})$ to $\text{Pol}(\mathbb{B})$.
7. There is a primitive positive construction of $\mathbb{B}$ in st-Con .

The k -absorbing operations of arity kn from item (3) are introduced under the name k -ABS operations in Section 3.1 in [1]. For us it is only important that whether an operation is k -absorbing can be described by a minor condition. Bodirsky and Starke introduced *unfolded caterpillar duality*, i.e., there is an obstruction set consisting of caterpillars that is closed under unfoldings, and used the result about caterpillar duality to obtain similar characterizations for structures with unfolded caterpillar duality [2]. In this article we will generalize their result.

Finally, one general observation about duality pairs that motivates why we defined caterpillars only for injective structures. Using the sparse incomparability theorem it is easy to show that we can assume without loss of generality that obstruction sets only contain injective structures, for a proof see for example Lemma 3.7 in [2].

Lemma 2.9. *Let $\mathbb{B}$ be a finite structure and let $\mathcal{F}$ be a class of finite structures such that $(\mathcal{F}, \mathbb{B})$ is a duality pair. Define*

$$\mathcal{F}' := \{\mathbb{F} \in \mathcal{F} \mid \mathbb{F} \text{ is injective}\}.$$

Then $(\mathcal{F}', \mathbb{B})$ is a duality pair.

2.7. Transitive tournaments and directed reachability

For $n \geq 1$ define the *transitive tournament on n vertices* to be the $\{E\}$ -structure

$$\mathbb{T}_n := ([0, n-1], \{(i, j) \mid 0 \leq i < j < n\}).$$

These structures will be central to this article. We now give a selection of well known facts about these tournaments for more information see for example Section 5.1 in [8]. It is true that there is a minion homomorphism from $\text{Pol}(\text{st-Con})$ to $\text{Pol}(\mathbb{T}_n)$ for all $n \geq 1$ and that there is a minion homomorphism from $\text{Pol}(\mathbb{T}_n)$ to $\text{Pol}(\mathbb{T}_m)$ if and only if $n \geq m$. Hence, we can order these structures in the following way

$$\text{st-Con} < \cdots < \mathbb{T}_4 < \mathbb{T}_3 < \mathbb{T}_2 < \mathbb{T}_1.$$

Furthermore, st-Con is the limit of this chain of $\mathbb{T}_n$'s in the sense that for any structure $\mathbb{B}$ with $\mathbb{T}_n \leq \mathbb{B}$ for all $n \geq 1$ we also have $\text{st-Con} \leq \mathbb{B}$. This chain lies at the top of the poset of all finite structures ordered by minion homomorphisms. The structure $\mathbb{T}_1$ is the unique maximal element and $\mathbb{T}_2$ is its unique lower cover. Meyer and Starke showed in 2024 that $\mathbb{T}_2$ has infinitely many lower covers: one for each finite simple group and $\mathbb{T}_3$ [12]. Unfortunately, $\mathbb{T}_4$ is not a lower cover of $\mathbb{T}_3$ as we will show in Example 5.1. It is well known that transitive tournaments are closely related to Hagemann-Mitschke chains in the following way. For a proof see Theorem 5.4 in [8].

Theorem 2.10. *Let $\mathbb{B}$ be a finite structure and $n \geq 0$. Then $\mathbb{B} \leq \mathbb{T}_{n+2}$ if and only if $\text{Pol}(\mathbb{B}) \not\models \text{HM}(n)$.*

In particular, $\mathbb{T}_{n+2} \not\models \text{HM}(n)$ and $\mathbb{T}_{n+2} \models \text{HM}(n+1)$. For $n \geq 0$ define the *directed path of length n* to be the $\{E\}$ -structure $\mathbb{P}_n := ([0, n], \{(i-1, i) \mid i \in [n]\})$. It is easy to see that $(\{\mathbb{P}_n\}, \mathbb{T}_n)$ is a duality pair for all $n \geq 1$. Hence, $\mathbb{T}_n$ has finite duality and, since $\mathbb{P}_n$ is a caterpillar, $\mathbb{T}_n$ also has caterpillar duality. We can now give a typical example of a $s_n\text{lam}$ Datalog program.

Example 2.11. Let $\mathbb{T}'_{n+2}$ be the $\{E, R\}$ -expansion of $\mathbb{T}_{n+2}$ where $R^{\mathbb{T}'_{n+2}}$ is the equality relation. Observe that the Datalog program consisting of the rules

$$\begin{aligned} P_1(y) &:- E(x, y) & P_{i+1}(y) &:- P_i(x) \wedge E(x, y) & i \in [n] \\ P_i(y) &:- P_i(x) \wedge R(x, y) & P_i(x) &:- P_i(y) \wedge R(x, y) & i \in [n+1] \\ G &:- P_{n+1}(x) \wedge E(x, y) \end{aligned}$$

solves $\text{CSP}(\mathbb{T}'_{n+2})$. Note that this program is n -almost symmetric, linear, arc, and monadic. $\triangle$

3. Caterpillars and Elevators

We generalize the notion of unfolded caterpillar duality. For $n \geq 0$ a set $\mathcal{F}$ of finite structures is *closed under n -fixed unfoldings (of caterpillars)* if for every caterpillar $\mathbb{C}$ in $\mathcal{F}$, there are vertebrae $v_1, \dots, v_m$ with $m \leq n$ such that any unfolding of $\mathbb{C}$ that fixes $v_1, \dots, v_m$ is contained in $\mathcal{F}$. The set $\mathcal{F}$ is closed under *unfoldings (of caterpillars)* if it is closed under 0-fixed unfoldings. A structure $\mathbb{B}$ has *n -fixed unfolded caterpillar duality* if there exists a set $\mathcal{F}$ of caterpillars such that

- $(\mathcal{F}, \mathbb{B})$ is a duality pair and
- $\mathcal{F}$ is closed under n -fixed unfoldings of caterpillars.

Note that unfolded caterpillar duality is 0-fixed unfolded caterpillar duality.

Observation 3.1. For all $n \geq 0$ the path $\mathbb{P}_{n+2}$ is a caterpillar with spine $(1, \dots, n+1)$. Hence, $\mathbb{P}_{n+2}$ has n vertebrae. Since $(\{\mathbb{P}_{n+2}\}, \mathbb{T}_{n+2})$ is a duality pair, $\mathbb{T}_{n+2}$ has n -fixed unfolded caterpillar duality, as we can fix all vertebra in $\mathbb{P}_{n+2}$.

Clearly, we can generalize this observation to the following lemma.

Lemma 3.2. *If $\mathbb{B}$ is a finite structure and $\mathcal{F}$ is a finite set of caterpillars such that $(\mathcal{F}, \mathbb{B})$ is a duality pair, then $\mathbb{B}$ has n -fixed unfolded caterpillar duality, where n is the maximal number of vertebrae in a caterpillar in $\mathcal{F}$.*

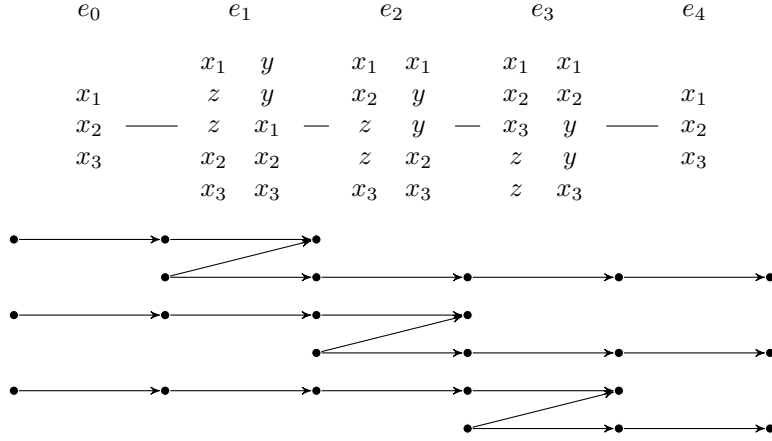


Figure 2: A visualization of $\text{Ele}(3)$ (top) and a 3-staircase (bottom), which is another way to visualize $\text{Ele}(3)$.

Let $n \in \mathbb{N}$, $\mathbb{C}$ be a caterpillar with spine $(a_0, \dots, a_m)$, and $0 \leq i_1 < j_1 \leq \dots \leq i_n < j_n \leq m$. For $k \in [n]$ let $\mathbb{C}_k$ denote the (a_{i_k}, a_{j_k}) -unfolding of $\mathbb{C}$. Then $\mathbb{C}$ is an n -folding of $(\mathbb{C}_1, \dots, \mathbb{C}_n)$. A set of structures $\mathcal{B}$ is *closed under n -foldings (of caterpillars)* if for all caterpillars $\mathbb{C}_1, \dots, \mathbb{C}_n \in \mathcal{B}$ we have that all n -foldings of $(\mathbb{C}_1, \dots, \mathbb{C}_n)$ are in $\mathcal{B}$.

In his dissertation Starke introduced the elevator condition [8]. Here we give an updated version of this definition. The identities are different but they capture the same idea.

Definition 3.3. For $n \geq 1$, the *elevator condition of length n* , called $\text{Ele}(n)$, consists of the identities

$$\begin{aligned}
 e_0(x_1, \dots, x_n) &\approx e_1(x_1, z, z, x_2, \dots, x_n), \\
 e_i(x_1, \dots, x_{i-1}, y, y, x_i, x_{i+1}, x_{i+2}, \dots, x_n) \\
 &\approx e_{i+1}(x_1, \dots, x_{i-1}, x_i, x_{i+1}, z, z, x_{i+2}, \dots, x_n) \quad \text{for all } i \in [n-1], \text{ and} \\
 e_n(x_1, \dots, x_{n-1}, y, y, x_n) &\approx e_{n+1}(x_1, \dots, x_n).
 \end{aligned}$$

See Figure 2 for a visualization of $\text{Ele}(3)$. A set of operations that satisfies $\text{Ele}(n)$ is called an *elevator chain of length n* . Note that $\text{Ele}(1)$ is equivalent to $\text{HM}(1)$. In general, we have the following connection.

Lemma 3.4. Let $n \geq 1$. Then $\text{Ele}(n)$ implies (for minions) $\text{HM}(n)$.

Proof. Let $\mathcal{M}$ be a minion and let $e_0, \dots, e_{n+1} \in \mathcal{M}$ form an elevator chain of length n . Define functions $p_0, \dots, p_{n+1}$ in the following way

$$\begin{aligned}
 p_0(a) &:= e_0(a, \dots, a) \\
 p_i(a, b, c) &:= e_i(c, \dots, c, a, b, c, a, \dots, a) \text{ with } b \text{ at position } i+1 \\
 p_{n+1}(c) &:= e_{n+1}(c, \dots, c)
 \end{aligned}$$

for all a, b, c . Note that $p_0, \dots, p_{n+1}$ are minors of $e_0, \dots, e_{n+1}$, respectively. Therefore, they are elements of $\mathcal{M}$. Furthermore, they form a Hagemann-Mitschke chain. Hence, $\mathcal{M} \models \text{HM}(n)$. $\square$

Observe that $\text{Ele}(n)$ implies

$$\begin{aligned}
 e_0(x_1, \dots, x_n) &\approx e_1(x_1, x_1, x_1, x_2, \dots, x_n) \approx \dots \\
 &\dots \approx e_n(x_1, \dots, x_{n-1}, x_n, x_n, x_n) \approx e_{n+1}(x_1, x_2, \dots, x_n).
 \end{aligned}$$

Hence, $e_0 = e_{n+1}$ in every elevator chain of length n .

4. Main Result

Now we can formulate our main result.

Theorem 4.1. *Let $n \geq 0$ and let $\mathbb{B}$ be a structure with a finite domain and a finite relational signature. Then the following are equivalent.*

1. $\text{Pol}(\mathbb{B})$ contains operations that form an elevator chain of length $n + 1$ and k -absorptive operations of arity ℓk , for all $\ell, k \geq 1$.
2. $\mathbb{B}$ has caterpillar duality and $\text{CSP}(\mathbb{B})$ is closed under $(n + 1)$ -foldings of caterpillars.
3. The canonical s_n lam Datalog program for $\mathbb{B}$ solves $\text{CSP}(\mathbb{B})$.
4. Some s_n lam Datalog program solves $\text{CSP}(\mathbb{B})$.
5. $\mathbb{B}$ has caterpillar duality and $\overline{\text{CSP}(\mathbb{B})}$ is closed under n -fixed unfoldings of caterpillars.
6. $\mathbb{B}$ has n -fixed unfolded caterpillar duality.
7. If $\text{Pol}(\mathbb{B})$ does not satisfy a minor condition Σ , then Σ implies (for finite clones) $\text{HM}(n)$.
8. Every minor condition that holds in $\text{Pol}(\mathbb{T}_{n+2})$ also holds in $\text{Pol}(\mathbb{B})$.
9. There is a minion homomorphism from $\text{Pol}(\mathbb{T}_{n+2})$ to $\text{Pol}(\mathbb{B})$.
10. There is a primitive positive construction of $\mathbb{B}$ in $\mathbb{T}_{n+2}$.
11. $\mathbb{B}$ is homomorphically equivalent to a structure $\mathbb{B}'$ such that $\text{Pol}(\mathbb{B}')$ contains operations that form an elevator chain of length $n + 1$ and operations $\sqcup$ and $\sqcap$ such that $(\mathbb{B}', \sqcup, \sqcap)$ forms a (distributive) lattice.

For $n = 0$ this is Theorem 3.1 from [2]. The proof of this theorem can be found in the appendix. We prove $(1) \Rightarrow (2) \Rightarrow (5) \Rightarrow (6) \Rightarrow (7) \Rightarrow (8) \Rightarrow (1)$ and $(5) \Rightarrow (3) \Rightarrow (4) \Rightarrow (5)$. The equivalence of $(8) - (10)$ follows from general results. Finally, we show $(1) \Leftrightarrow (11)$. Combining Lemma 3.2 and Theorem 4.1 yields the following corollary.

Corollary 4.2. *If $\mathbb{B}$ is a finite structure and $\mathcal{F}$ is a finite set of caterpillars such that $(\mathcal{F}, \mathbb{B})$ is a duality pair, then items (1) – (11) hold for $\mathbb{B}$, where n is the maximal number of vertebrae in a caterpillar in $\mathcal{F}$.*

5. Examples

Here we present some examples of interesting structures that can be pp-constructed from st-Con.

Example 5.1. Let $\mathbb{P}_{313}$ be the digraph with vertices $\{0, 1, 2, 3, 2', 3', 4', 5'\}$ and edges $\{(0, 1), (1, 2), (2, 3), (2', 3), (2', 3'), (3', 4'), (4', 5')\}$. Let $\mathcal{F}$ be the set of all unfoldings of $\mathbb{P}_{323}$ that fix the vertebrae indicated by the bold edges in Figure 3. Observe that $(\mathcal{F}, \mathbb{P}_{313})$ is a duality pair. Hence, 2 is the smallest number n such that $\mathbb{P}_{313}$ has n -fixed unfolded caterpillar duality and, by Theorem 4.1, $\mathbb{P}_{313}$ is pp-constructable from $\mathbb{T}_4$ but not from $\mathbb{T}_3$. Furthermore, $\mathbb{P}_{313} \models \text{HM}(2)$ and $\mathbb{P}_{313} \not\models \text{HM}(1)$ (verified by computer). Hence, $\mathbb{P}_{313}$ can not pp-construct $\mathbb{T}_4$ and it can pp-construct $\mathbb{T}_3$. Therefore, $\mathbb{P}_{313}$ lies strictly between $\mathbb{T}_4$ and $\mathbb{T}_3$. $\triangle$

Example 5.2. Let $\mathbb{P}_{414}$ be the digraph with vertices $\{0, 1, 2, 3, 4, 3', 4', 5', 6', 7'\}$ and edges $\{(0, 1), (1, 2), (2, 3), (3, 4), (3', 4), (3', 4'), (4', 5'), (5', 6'), (6', 7')\}$. Let $\mathcal{F}$ be the set of all unfoldings of $\mathbb{P}_{423}$ and $\mathbb{P}_{324}$ that fix the vertebrae indicated by the bold edges in Figure 4. Observe that $(\mathcal{F}, \mathbb{P}_{414})$ is a duality pair. Hence, 3 is the smallest number n such that $\mathbb{P}_{414}$ has n -fixed unfolded caterpillar duality and, by Theorem 4.1, $\mathbb{P}_{414}$ is pp-constructable from $\mathbb{T}_5$ but not from $\mathbb{T}_4$. Furthermore, $\mathbb{P}_{414} \models \text{HM}(2)$ (verified by computer). Hence, $\mathbb{P}_{414}$ can not pp-construct $\mathbb{T}_4$. Therefore, $\mathbb{P}_{414}$ can be pp-constructed from $\mathbb{T}_5$ and is (pp-)incomparable with $\mathbb{T}_4$. $\triangle$

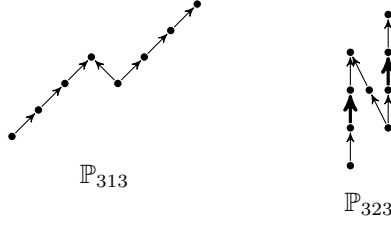


Figure 3: An example of a graph (left) that is strictly between $\mathbb{T}_4$ and $\mathbb{T}_3$ and one of its obstructions (right).

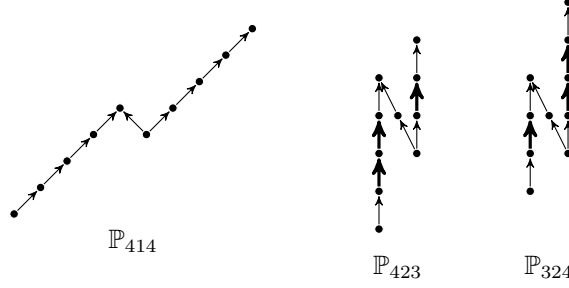


Figure 4: An example of a graph (left) that is strictly above $\mathbb{T}_5$ and is incomparable with $\mathbb{T}_4$ and some of its obstructions (right).

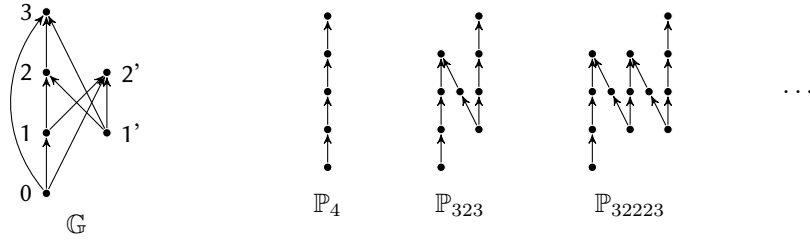


Figure 5: An example of a graph (left) that is strictly above st-Con but can not be pp-constructed from any $\mathbb{T}_n$ and its obstructions (right).

Example 5.3. Let $\mathbb{G}$ be the digraph with vertices $\{0, 1, 1', 2, 2', 3\}$ and edges

$$\{(0, 1), (1, 2), (2, 3), (0, 3), (1', 2), (1', 3), (0, 2'), (1, 2'), (1', 2')\}$$

and $\mathbb{P}_4, \mathbb{P}_{323}, \mathbb{P}_{32223}, \dots$ be oriented paths as depicted in Figure 5. Observe that the tuple $(\{\mathbb{P}_4, \mathbb{P}_{323}, \mathbb{P}_{32223}, \dots\}, \mathbb{G})$ is a duality pair. By Theorem 2.8 $\mathbb{G}$ is pp-constructable from st-Con. Furthermore, $\mathbb{G} \models \text{HM}(3)$ (verified by computer). Hence $\mathbb{G}$ is strictly above st-Con. Note that $\text{CSP}(\mathbb{G})$ can be solved by a symmetric linear monadic Datalog Program but not by a $s_n\text{lam}$ Datalog program. Hence, by Theorem 4.1, there is no n such that $\mathbb{T}_n$ pp-constructs $\mathbb{G}$. $\triangle$

6. Conclusions

We characterized the structures whose CSP is solvable by $s_n\text{lam}$ Datalog as those that are pp-constructable from $\mathbb{T}_{n+2}$. Recall that pp-constructability orders the transitive tournaments in the following way $\text{st-Con} < \dots < \mathbb{T}_4 < \mathbb{T}_3 < \mathbb{T}_2 < \mathbb{T}_1$. In Section 5 we have seen that not all structures pp-constructable from st-Con are pp-constructable from some $\mathbb{T}_n$. We suggest introducing (n, m) -almost symmetric Datalog, which allows, besides the usual n non-symmetric rule applications, derivations $P_0(a_0) \vdash_{R_1} \dots \vdash_{R_k} P_k(a_k)$ with at most m non-symmetric rules if there are rules to derive

$$P_k(a_k) \vdash_{\tilde{R}_k} \cdots \vdash_{\tilde{R}_1} P_0(a_0).$$

Conjecture 6.1. *A structure $\mathbb{B}$ lies strictly above st-Con if and only if there are $n, m \in \mathbb{N}$ such that $\text{CSP}(\mathbb{B})$ is solved by an (n, m) -almost symmetric linear arc monadic Datalog program. This is exactly the case if $\mathbb{B}$ has caterpillar duality and $\text{CSP}(\mathbb{B})$ is solvable by symmetric linear Datalog.*

Another avenue of future research is starting again with slam Datalog and replacing linearity by *n-almost linearity*. We expect this to yield Datalog fragments for CSPs of the structures pp-constructed from

$$\mathbb{B}_n = (\{0, 1\}, \{1\}, \{0, 1\}^n \setminus \{(1, \dots, 1)\}).$$

It is well known that these structures form a chain $\mathbb{B}_\infty < \cdots < \mathbb{B}_3 < \mathbb{B}_2$ and that $\mathbb{B}_2$ is pp-interconstructable with $\mathbb{T}_3$. This would be a next step in characterizing all structures that are pp-constructable from a structure with finite duality.

Acknowledgements. The authors would like to thank Andrew Moorhead for many helpful and motivating discussions. Both authors have received funding from the European Research Council (Project POCOCOP, ERC Synergy Grant 101071674). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Declaration on Generative AI. During the preparation of this work, the authors used ChatGPT in order to search for typos. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] C. Carvalho, V. Dalmau, A. A. Krokhin, Two new homomorphism dualities and lattice operations, *J. Log. Comput.* 21 (2011) 1065–1092. doi:10.1093/LOGCOM/EXQ030.
- [2] M. Bodirsky, F. Starke, Symmetric linear arc monadic datalog and gadget reductions, *Theory Comput. Syst.* 70 (2026) Paper No. 18, 34. doi:10.1007/s00224-026-10261-2.
- [3] W. Hodges, *A shorter model theory*, Cambridge University Press, Cambridge, 1997.
- [4] M. Bodirsky, F. Starke, Maximal digraphs with respect to primitive positive constructability, *Combinatorica* 42 (2022) 997–1010.
- [5] M. Bodirsky, A. Vucaj, Two-element structures modulo primitive positive constructability, *Algebra Universalis* 81 (2020). Preprint available at ArXiv:1905.12333.
- [6] L. Barto, J. Opršal, M. Pinsker, The wonderland of reflections, *Israel Journal of Mathematics* 223 (2018) 363–398.
- [7] D. N. Zhuk, A proof of CSP dichotomy conjecture, in: 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15–17, 2017, pp. 331–342.
- [8] F. Starke, Digraphs modulo primitive positive constructability, Ph.D. thesis, Dresden University of Technology, Germany, 2024. URL: <https://arxiv.org/abs/2402.16729>.
- [9] J. Bulín, A. A. Krokhin, J. Opršal, Algebraic approach to promise constraint satisfaction, in: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23–26, 2019, pp. 602–613.
- [10] M. Bodirsky, V. Dalmau, Datalog and constraint satisfaction with infinite templates, *Journal on Computer and System Sciences* 79 (2013) 79–100. A preliminary version appeared in the proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS’05).
- [11] T. Feder, 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 (1999) 57–104.
- [12] S. Meyer, F. Starke, Finite simple groups in the primitive positive constructability poset, 2025. URL: <https://arxiv.org/abs/2409.06487>.

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

A. Almost Symmetric Datalog

In this section we will show that n -almost symmetric Datalog programs can solve the same problems as symmetric Datalog programs. Let Π be an n -almost symmetric Datalog program. Note that the rule of that derives the goal predicate in a derivation of Π is by definition not symmetric. The idea is to use high ary IDBs to apply all non-symmetric rules in parallel with the initial rule and then verify (using symmetric rules) that these applications were valid.

Example A.1. In Example 2.11 we have seen that $\text{CSP}(\mathbb{T}'_{n+2})$ is solved by a s_n lam Datalog program. This CSP is also solved by the symmetric arc Datalog program consisting of the rules

$$\begin{aligned} P(x'_1, x_1, x'_2, x_2, \dots, x'_{n+1}, x_{n+1}) &:- E(x, x'_1) \wedge E(x_1, x'_2) \wedge \dots \wedge E(x_n, x'_{n+1}) \wedge E(x_{n+1}, y) \\ P(x'_1, x_1, \dots, x'_i, x_i, \dots, x'_{n+1}, x_{n+1}) &:- P(x'_1, x_1, \dots, x'_i, x_i, \dots, x'_{n+1}, x_{n+1}) \wedge R(x'_i, x'_i) \\ P(x'_1, x_1, \dots, x'_i, x_i, \dots, x'_{n+1}, x_{n+1}) &:- P(x'_1, x_1, \dots, x'_i, x_i, \dots, x'_{n+1}, x_{n+1}) \wedge R(x''_i, x'_i) \end{aligned}$$

for $i \in [n+1]$ and the rule $G :- P(x_1, x_1, \dots, x_{n+1}, x_{n+1})$. $\triangle$

Generalizing this construction we obtain the following lemma.

Lemma A.2. Let $n \in \mathbb{N}$ and let Π be an n -almost symmetric Datalog program. Then there is a symmetric Datalog program Π' such that Π' can derive the goal predicate on a structure $\mathbb{A}$ if and only if Π can derive the goal predicate on $\mathbb{A}$.

Proof. Let $\sim$ denote the equivalence relation induced by the quasi-order $\leq$ on the IDBs of Π . We construct the symmetric Datalog program Π' . For all IDBs $P_1 \sim Q_1 < P_2 \sim Q_2 < \dots < P_k \sim Q_k$ of Π we add the IDB $P_{P_1, Q_1, P_2, Q_2, \dots, P_k, Q_k}$ to Π' , the arity of this new IDB is the sum of the arities of $P_1, Q_1, P_2, Q_2, \dots, P_k, Q_k$. Note that since Π is n -almost symmetric we have that $k \leq n+1$. Hence, we added only finitely many IDBs to Π' . For each list $R_0, \dots, R_k$ of rules of Π such that

- R_0 is of the form $P_1(\bar{x}_1) :- \phi_0(\bar{y}_0)$,
- for $i \in [k-1]$ the rule R_i is of the form $P_{i+1}(\bar{x}'_{i+1}) :- Q_i(\bar{x}_i) \wedge \phi_i(\bar{y}_i)$, where Q_i is the unique IDB in the body of R_i ,
- R_k is of the form $G :- Q_k(\bar{x}_k) \wedge \phi_k(\bar{y}_k)$, and
- $P_1 \sim Q_1 < P_2 \sim Q_2 < \dots < P_k \sim Q_k$

we add the rule

$$P_{P_1, Q_1, \dots, P_k, Q_k}(\bar{x}'_1, \bar{x}_1, \dots, \bar{x}'_k, \bar{x}_k) :- \phi_0(\bar{y}_0) \wedge \phi_1(\bar{y}_1) \wedge \dots \wedge \phi_k(\bar{y}_k)$$

to Π' . We assume here that the sets of variables used in each rule are distinct. For each IDB $P_{P_1, Q_1, \dots, P_k, Q_k}$ of Π' and each rule $P'_i(\bar{x}''_i) :- P_i(\bar{x}'_i) \wedge \phi(\bar{y}'_i)$ of Π , with $P'_i \sim P_i$, we add to Π' the rule with the head $P_{P_1, Q_1, \dots, P'_i, Q_i, \dots, P_k, Q_k}(\bar{x}'_1, \bar{x}_1, \dots, \bar{x}''_i, \bar{x}_i, \dots, \bar{x}'_k, \bar{x}_k)$ and the body

$$P_{P_1, Q_1, \dots, P_i, Q_i, \dots, P_k, Q_k}(\bar{x}'_1, \bar{x}_1, \dots, \bar{x}'_i, \bar{x}_i, \dots, \bar{x}'_k, \bar{x}_k) \wedge \phi(\bar{y}'_i)$$

where $\bar{x}'_1, \bar{x}_1, \dots, \bar{x}_{i-1}, \bar{x}_i, \bar{x}'_{i+1}, \dots, \bar{x}'_k, \bar{x}_k$ are tuples of new distinct variables. Note that since $P'_i \sim P_i$ we have that $P_i(\bar{x}'_i) :- P'_i(\bar{x}''_i) \wedge \phi(\bar{y}'_i)$ is also a rule Π . Hence, Π' is, after adding all of these rules, symmetric. Finally, we add for each IDB $P_{Q_1, Q_1, \dots, Q_k, Q_k}$ of Π' the rule

$$G :- P_{Q_1, Q_1, \dots, Q_k, Q_k}(\bar{x}_1, \bar{x}_1, \dots, \bar{x}_k, \bar{x}_k)$$

where $\bar{x}_1, \dots, \bar{x}_k$ are tuples of new distinct variables. Let $\mathbb{A}$ be a structure. It is easy to see that Π' can derive the goal predicate on $\mathbb{A}$ with a derivation that derives $P_{Q_1, Q_1, \dots, Q_k, Q_k}$ on the tuple $(\bar{a}_1, \bar{a}_1, \dots, \bar{a}_k, \bar{a}_k)$ if and only if Π can derive the goal predicate on $\mathbb{A}$ with a derivation that derives Q_1 on $\bar{a}_1, \dots$, and Q_k on $\bar{a}_k$. Hence, Π and Π' can derive the goal predicate on the same structures. $\square$

B. Folding and Unfolding Caterpillars

First we show a connection between elevator chains and caterpillars.

Lemma B.1. *Let $n \in \mathbb{N}$ and $\mathbb{B}$ be a finite structure such that $\text{Pol}(\mathbb{B})$ contains operations that form an elevator chain of length n . Then $\text{CSP}(\mathbb{B})$ is closed under n -foldings of caterpillars.*

Proof. Let $\mathbb{C}$ be a caterpillar with spine $(a_0, \dots, a_m)$ and $0 \leq i_1 < j_1 \leq \dots \leq i_n < j_n \leq m$ such that for each $k \in [n]$ the (a_{i_k}, a_{j_k}) -unfolding of $\mathbb{C}$ has a homomorphism h_k to $\mathbb{B}$. Let $e_0, \dots, e_{n+1} \in \text{Pol}(\mathbb{B})$ be operations that form an elevator chain of length n . Let $e = e_0$ and recall that $e_{n+1} = e$ and that $e_k(c_1, \dots, c_{k-1}, c_k, c_k, c_k, c_{k+1}, \dots, c_n) = e(c_1, \dots, c_n)$ for all $k \in [n]$ and all $c_1, \dots, c_n \in \mathbb{C}$. Define a map $h: \{a_0, \dots, a_m\} \rightarrow \mathbb{B}$ where $h(a_\ell)$ is

- $e_k(h_1(a_\ell), \dots, h_{k-1}(a_\ell), h_k(a'_\ell), h_k(a_\ell), h_k(a''_\ell), h_{k+1}(a_\ell), \dots, h_n(a_\ell))$ if $i_k < \ell < j_k$, where a'_ℓ and a''_ℓ are the copies of a_ℓ in the (a_{i_k}, a_{j_k}) -unfolding of $\mathbb{C}$ according to the naming convention from the definition of unfoldings and
- $e(h_1(a_\ell), \dots, h_n(a_\ell))$ if ℓ is in none of the intervals $[i_k + 1, j_k - 1]$.

By looking at the definition of $\text{Ele}(n)$ it is easy to see, that h can be extended to a homomorphism from $\mathbb{C}$ to $\mathbb{B}$. $\square$

The remainder of this section is dedicated to proving the implication $(2) \Rightarrow (5)$, by studying the connection between $(n + 1)$ -foldings and n -fixed unfoldings.

Example B.2. Recall that the duality pair $(\{\mathbb{P}_4\}, \mathbb{T}_4)$ witnesses that $\mathbb{T}_4$ has 2-fixed unfolded caterpillar duality and that $\mathbb{P}_4$ is a caterpillar with spine $(1, 2, 3)$. Let $\mathbb{P}_{213}$ denote the $(1, 2)$ -unfolding of $\mathbb{P}_4$. The name $\mathbb{P}_{213}$ comes from the fact that $\mathbb{P}_{231}$ is an oriented path consisting of 2 forward edges, followed by 1 backward edge, followed by 3 forward edges. Using this naming convention it is clear that $\mathbb{P}_{312}$ is the $(2, 3)$ -unfolding of $\mathbb{P}_4$. Our main theorem predicts that $\text{CSP}(\mathbb{T}_4)$ is not closed under 1-fixed unfoldings and that $\text{CSP}(\mathbb{T}_4)$ is not closed under 2-foldings. The former holds, since $\mathbb{P}_{213} \rightarrow \mathbb{T}_4$ shows that not all unfoldings of $\mathbb{P}_4$ that fix the vertebra $(2, 3)$ are obstructions of $\mathbb{T}_4$ and analogously $\mathbb{P}_{312} \rightarrow \mathbb{T}_4$ shows that fixing the vertebra $(1, 2)$ is also not sufficient. These structures also witness the latter, since $\mathbb{P}_4$ is a 2-folding of $(\mathbb{P}_{213}, \mathbb{P}_{312})$, $\mathbb{P}_{213} \rightarrow \mathbb{T}_4$, $\mathbb{P}_{312} \rightarrow \mathbb{T}_4$, and $\mathbb{P}_4 \not\rightarrow \mathbb{T}_4$.

In this case the connection from 2-foldings to 1-fixed unfoldings is easy. Now consider the paths $\mathbb{P}_{21113}$ and $\mathbb{P}_{31112}$. They still witness that $\text{CSP}(\mathbb{T}_4)$ is not closed under 1-fixed unfoldings. However, for 2-foldings we need to do more work. Note that $\mathbb{P}_{21212}$ is a 2-folding of $(\mathbb{P}_{21113}, \mathbb{P}_{31112})$ and that $\mathbb{P}_{21212} \rightarrow \mathbb{T}_4$. We can still salvage these witnesses, since $\mathbb{P}_{213}$ is a 2-folding of $(\mathbb{P}_{21113}, \mathbb{P}_{21212})$ and $\mathbb{P}_{312}$ is a 2-folding of $(\mathbb{P}_{21212}, \mathbb{P}_{31112})$, yielding our original witnesses. It is easy to see that this approach still works when the two initial witnesses are even more unfolded. $\triangle$

In general this can become much more complicated since the number of steps we might have to take increases significantly in n and witnesses might not be as simple as unfolding a single vertebra multiple times. We will deal with the first complication in Lemma B.4 and with the second one in Lemma B.6. Note that in the example we essentially only cared about the number of times a vertebra was unfolded. Hence, we introduce a version of foldings for numbers and translate our findings to caterpillars afterwards.

Let $n \in \mathbb{N}$ and let $r_1, \dots, r_n \in \mathbb{N}^n$. The n -folding of $(r_1, \dots, r_n)$ is the tuple $r \in \mathbb{N}^n$, where the i -th entry of r is

$$\max((r_1)_i, \dots, (r_{i-1})_i, (r_i)_i - 1, (r_{i+1})_i, \dots, (r_n)_i).$$

Let $M \subseteq \mathbb{N}^n$. The set M is *closed under n -foldings* if for all $r_1, \dots, r_n \in M$ the n -folding of $(r_1, \dots, r_n)$ is contained in M .

Lemma B.3. *Let $n, m \in \mathbb{N}$ and let $M \subseteq \mathbb{N}^n$. If M is closed under n -foldings and $(m, 0, \dots, 0), (0, m, \dots, 0), \dots, (0, 0, \dots, m) \in M$, then $(a_1, \dots, a_n) \in M$ for all $a_1, \dots, a_n \in [0, m-1]$. In particular $(0, \dots, 0) \in M$.*

It is a nice exercise to show this lemma from hand for the case $n = m = 3$.

Proof. We order the set $[0, m-1]^n$ lexicographically and show by induction that all its elements are contained in M . For the base case consider the following diagram

$$\begin{array}{rcccc} e_1 = & (m & 0 & \dots & 0) \\ e_2 = & (0 & m & \dots & 0) \\ \vdots & \vdots & \vdots & & \vdots \\ e_n = & (0 & 0 & \dots & m) \\ \hline e = & (m-1 & m-1 & \dots & m-1) \end{array}$$

it shows that e is the n -folding of $(e_1, \dots, e_n)$. Hence $e \in M$. For the induction step let $i \in [n]$ and $a_i, \dots, a_n \in [1, m-1]$. Consider the diagram

$$\begin{array}{rccccccc} r_1 = & (a_1 + 1 & \dots & a_{i-1} & a_i & 0 & \dots & 0) \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ r_{i-1} = & (a_1 & \dots & a_{i-1} + 1 & a_i & 0 & \dots & 0) \\ r_i = & (a_1 & \dots & a_{i-1} & a_i & 0 & \dots & 0) \\ r_{i+1} = & (0 & \dots & 0 & 0 & m & \dots & 0) \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ r_n = & (0 & \dots & 0 & 0 & 0 & \dots & m) \\ \hline r = & (a_1 & \dots & a_{i-1} & a_i - 1 & m-1 & \dots & m-1) \end{array}$$

it shows that r is the n -folding of $(r_1, \dots, r_n)$. Note that r_i is the (lexicographic) successor of r . We want to show $r \in M$. For $j \in [n]$ define r'_j to be

- r_j if $j \leq i$ and $a_j + 1 < m$ and
- e_j otherwise.

Note that r is the n -folding of $(r'_1, \dots, r'_n)$. Furthermore for every $j \in [n]$ we have that either

- r'_j is e_j , which is in M or
- r'_j (lexicographically) indirectly succeeds r and $r'_j \in [0, m-1]^n$, hence $r'_j \in M$ by induction hypothesis.

Since M is closed under n -foldings we have that $r \in M$. □

Now we translate the previous result about n -foldings of tuples to n -foldings of caterpillars.

Lemma B.4. *Let $\mathbb{B}$ be a finite structure whose CSP is closed under n -foldings of caterpillars, $\mathbb{C}$ be a caterpillar with spine $(a_0, \dots, a_m)$ and vertebrae $v_1 = (a_0, a_1), \dots, v_m = (a_{m-1}, a_m)$, and $0 \leq i_1 \leq j_1 \leq \dots \leq i_n \leq j_n \leq m$. Let $\mathbb{C}_1, \dots, \mathbb{C}_n$ be caterpillars such that for each $\ell \in [n]$ we have that $\mathbb{C}_\ell$ is an unfolding of $\mathbb{C}$ that fixes all vertebrae of $\mathbb{C}$ that are not in $\mathbb{C}[i_\ell, j_\ell]$. If $\mathbb{C}_1, \dots, \mathbb{C}_n \rightarrow \mathbb{B}$, then $\mathbb{C} \rightarrow \mathbb{B}$.*

Proof. If there is an $\ell \in [n]$ with $i_\ell = j_\ell$, then $\mathbb{C} = \mathbb{C}_\ell \rightarrow \mathbb{B}$. Hence, we can assume that $i_\ell < j_\ell$ for all ℓ . For each $\ell \in [n]$ let $(a_1^\ell, b_1^\ell), \dots, (a_{k(\ell)}^\ell, b_{k(\ell)}^\ell)$ be such that the sequence

$$\mathbb{C} = \mathbb{C}_{(0, \dots, 0, \dots, 0)}, \mathbb{C}_{(0, \dots, 1, \dots, 0)}, \dots, \mathbb{C}_{(0, \dots, k(\ell), \dots, 0)} = \mathbb{C}_\ell$$

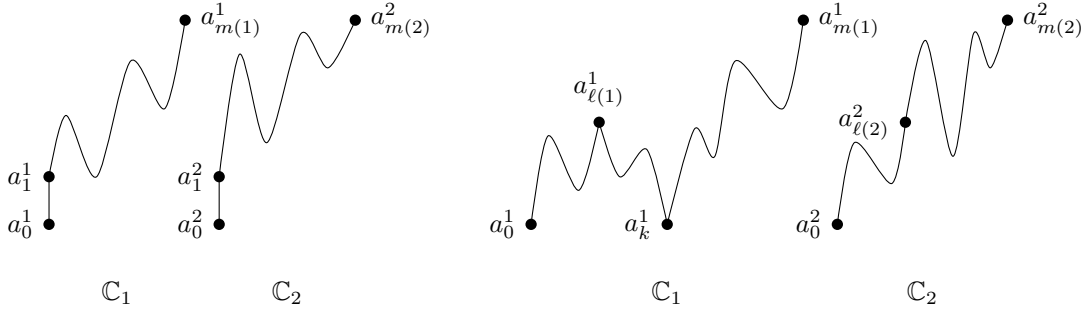


Figure 6: The two cases in the proof of Lemma B.5. The first case is on the left side and second is on the right.

is well defined, where $\mathbb{C}_{(0,\dots,i+1,\dots,0)}$ is the $(a_{i+1}^\ell, b_{i+1}^\ell)$ -unfolding of $\mathbb{C}_{(0,\dots,i,\dots,0)}$. Note that since $i_\ell < j_\ell$ for all ℓ we could replace each $\mathbb{C}_\ell$ by a further unfolded version of itself. Hence we can assume without loss of generality that all $k(\ell)$ are equal to $k := \max(k(1), \dots, k(n))$. We can extend the definition of $\mathbb{C}_{(0,\dots,i,\dots,0)}$ to tuples with more than one non-zero entry: For $(k_1, \dots, k_n) \in [0, k]^n$ and $\ell \in [n]$ with $k_\ell < k$ define $\mathbb{C}_{(k_1,\dots,k_\ell+1,\dots,k_n)}$ as the $(a_{k_\ell+1}^\ell, b_{k_\ell+1}^\ell)$ -unfolding of $\mathbb{C}_{(k_1,\dots,k_\ell,\dots,k_n)}$.

Let M be the set of all tuples $t \in [0, k]^n$ such that $\mathbb{C}_t \rightarrow \mathbb{B}$. By definition

$$(k, 0, \dots, 0), (0, k, \dots, 0), \dots, (0, 0, \dots, k) \in M.$$

We will show that M is closed under n -foldings. Let $r_1, \dots, r_n \in M$ and let r be the n -folding of $(r_1, \dots, r_n)$. For $\ell \in [n]$ let r'_ℓ be the tuple obtained from r by increasing the ℓ -th entry by 1. Note that $r_\ell \leq r'_\ell$ componentwise. Hence, $\mathbb{C}_{r'_\ell} \rightarrow \mathbb{C}_{r_\ell} \rightarrow \mathbb{B}$. Observe that $\mathbb{C}_r$ is the n -folding of $(\mathbb{C}_{r'_1}, \dots, \mathbb{C}_{r'_n})$. Therefore, $\mathbb{C}_r \rightarrow \mathbb{B}$ and $r \in M$. By Lemma B.3 $(0, \dots, 0) \in M$. Hence, $\mathbb{C} = \mathbb{C}_{(0,\dots,0)} \rightarrow \mathbb{B}$, as desired. $\square$

Next we introduce some convenient notation for manipulating caterpillars. Let $\mathbb{C}$ be a caterpillar with conjunctive query representation $(\phi \wedge \psi)(a_0, \dots, a_m)$ and let $0 \leq i \leq j \leq m$. The $[i, j]$ -interval of $\mathbb{C}$, denoted $\mathbb{C}[i, j]$, is the canonical database of

$$(\phi_i(a_i, a_{i+1}) \wedge \dots \wedge \phi_{j-1}(a_{j-1}, a_j)) \wedge (\psi_i(a_i) \wedge \dots \wedge \psi_j(a_j)).$$

Note that $\mathbb{C}[i, j]$ is a caterpillar. The spine of $\mathbb{C}[i, j]$ is $(a_i, \dots, a_j)$. Recall that $(a_j, \dots, a_i)$ is also the spine of $\mathbb{C}[i, j]$. We denote $\mathbb{C}[i, j]$ with this *reversed spine* by $\mathbb{C}[j, i]$. We define the (i, j) -interval of $\mathbb{C}$, denoted $\mathbb{C}(i, j)$, as the canonical database of

$$(\phi_i(a_i, a_{i+1}) \wedge \dots \wedge \phi_{j-1}(a_{j-1}, a_j)) \wedge (\psi_{i+1}(a_{i+1}) \wedge \dots \wedge \psi_{j-1}(a_{j-1})).$$

Analogously we define $\mathbb{C}(i, j]$ and $\mathbb{C}[i, j)$. Using this notation we can show that any unfoldings of a caterpillar have a common unfolding, similar to a common multiple for numbers.

Lemma B.5. *Let $\mathbb{C}$ be a caterpillar and let $\mathbb{C}_1$ and $\mathbb{C}_2$ be unfoldings of $\mathbb{C}$ fixing $v_1, \dots, v_n$. Then there exists a caterpillar $\mathbb{C}'$ that is an unfolding of $\mathbb{C}_1$ and of $\mathbb{C}_2$ fixing $v_1, \dots, v_n$.*

Proof. Let $\mathbb{C}$ be a caterpillar with spine $(a_0, \dots, a_m)$ and vertebrae $v_1 = (a_0, a_1), \dots, v_m = (a_{m-1}, a_m)$, let $0 \leq i(1) < \dots < i(n) \leq m$, and let $\mathbb{C}_1$ and $\mathbb{C}_2$ be unfoldings of $\mathbb{C}$ fixing $v_{i(1)}, \dots, v_{i(n)}$. Let $h_1: \mathbb{C}_1 \rightarrow \mathbb{C}$ and $h_2: \mathbb{C}_2 \rightarrow \mathbb{C}$ be the homomorphisms from the definition of unfoldings and let $(a_0^1, \dots, a_{m(1)}^1)$ and $(a_0^2, \dots, a_{m(2)}^2)$ be the spines of $\mathbb{C}_1$ and $\mathbb{C}_2$, respectively. We proceed with an induction on $m(1) + m(2)$. If $m(1) + m(2) = 0$, then $\mathbb{C}_1 = \mathbb{C}_2 = \mathbb{C}$ and $\mathbb{C}$ is a common unfolding of both. For $m(1) + m(2) > 0$ there are two cases to consider, see Figure 6 for a visualization. In the first case $h_1^{-1}(a_0) = \{a_0^1\}$ and $h_2^{-1}(a_0) = \{a_0^2\}$. Then $h_i(a_k^i) \in \{a_1, \dots, a_m\}$ for all $i \in \{1, 2\}$ and all $k \in [m_i]$. Therefore $\mathbb{C}_1[1, m(1)]$ and $\mathbb{C}_2[1, m(2)]$ are unfoldings of $\mathbb{C}[1, m]$ fixing $v_{i(1)}, \dots, v_{i(n)}$.

By induction there is a common unfolding $\mathbb{C}'$ of $\mathbb{C}_1[1, m(1)]$ and $\mathbb{C}_2[1, m(2)]$ that fixes $v_{i(1)}, \dots, v_{i(n)}$. Replacing the interval $\mathbb{C}[1, m]$ in $\mathbb{C}$ by $\mathbb{C}'$ yields the desired common unfolding of $\mathbb{C}_1$ and $\mathbb{C}_2$. The second case is that either $|h_1^{-1}(a_0)| \geq 2$ or $|h_2^{-1}(a_0)| \geq 2$. Assume without loss of generality that $h_1^{-1}(a_0) = \{a_0^1, a_k^1, \dots\}$. Let ℓ be the largest number in $[m]$ such that there exists an $\ell(1) \in [0, k]$ with $h_1(a_{\ell(1)}^1) = a_\ell$. Let $\ell(2)$ be the smallest number in $[0, m(2)]$ with $h_2(a_{\ell(2)}^2) = a_\ell$. Note that by definition $v_{i(1)}, \dots, v_{i(n)}$ must be in the interval $\mathbb{C}[\ell, m]$ and that

- $\mathbb{C}_1[0, \ell(1)]$ and $\mathbb{C}_2[0, \ell(2)]$ are unfoldings of $\mathbb{C}[0, \ell]$,
- $\mathbb{C}_1[\ell(1), k]$ and $\mathbb{C}_2[\ell(2), 0]$ are unfoldings of $\mathbb{C}[\ell, 0]$, and
- $\mathbb{C}_1[\ell(1), m(1)]$ and $\mathbb{C}_2[0, m(2)]$ are unfoldings of $\mathbb{C}[0, m]$ fixing $v_{i(1)}, \dots, v_{i(n)}$.

By induction there are common unfoldings $\mathbb{C}'_1, \mathbb{C}'_2, \mathbb{C}'_3$. By taking $\mathbb{C}_1$ and replacing

$$\mathbb{C}_1[0, \ell(1)] \text{ by } \mathbb{C}'_1, \quad \mathbb{C}_1[\ell(1), k] \text{ by } \mathbb{C}'_2, \quad \text{and} \quad \mathbb{C}_1[k, m(1)] \text{ by } \mathbb{C}'_3$$

we obtain a caterpillar $\mathbb{C}'$. By construction $\mathbb{C}'$ is the desired common unfolding of $\mathbb{C}_1$ and $\mathbb{C}_2$ fixing $v_{i(1)}, \dots, v_{i(n)}$. Finally, observe that if $v_{i(1)} = v_1$, then $h_1^{-1}(a_0) = \{a_0^1\}$ and $h_2^{-1}(a_0) = \{a_0^2\}$. Hence, we are in the first case and the constructed common unfolding fixes v_1 . $\square$

Lemma B.6. *Let $n \in \mathbb{N}$ and $\mathbb{B}$ be a finite structure whose CSP is closed under $(n+1)$ -foldings of caterpillars. Then $\overline{\text{CSP}(\mathbb{B})}$ is closed under n -fixed unfoldings of caterpillars.*

Proof. Let $\mathbb{C}$ be a caterpillar with spine $(a_0, \dots, a_m)$ and vertebra $v_1 = (a_0, a_1), \dots, v_m = (a_{m-1}, a_m)$ such that for all $n' \leq n$ and all $1 \leq i_1 < \dots < i_{n'} \leq m$ there is an unfolding $\mathbb{C}_{\{v_1, \dots, v_{n'}\}}$ of $\mathbb{C}$ fixing $v_{i_1}, \dots, v_{i_{n'}}$ such that $\mathbb{C}_{\{v_1, \dots, v_{n'}\}} \rightarrow \mathbb{B}$. As discussed in Example B.2, it suffices to show $\mathbb{C} \rightarrow \mathbb{B}$ to prove the lemma. Let $V := \{v_1, \dots, v_m\}$ and $\mathcal{I}$ be the set of all subsets I of V such that there is an unfolding $\mathbb{C}_I$ of $\mathbb{C}$ that fixes all vertebrae in I and $\mathbb{C}_I \rightarrow \mathbb{B}$. Note that $\mathcal{I}$ is closed under taking subsets. For each $I \subseteq V$ the *characteristic tuple* $t_I \in \{0, 1\}^m$ is the tuple that has for all $i \in [m]$ a 1 in the i -th entry if and only if $v_i \in I$. We order the subsets of V by comparing their characteristic tuples lexicographically. We will show by induction that $\mathcal{I}$ contains all subsets of V .

Let $n' \in [0, m]$ and $1 \leq i(1) < \dots < i(n') \leq m$. Define $I := \{v_{i(1)}, \dots, v_{i(n')}\}$. If $n' \leq n$, then $I \in \mathcal{I}$ by choice of $\mathbb{C}$. If $n' \geq n+1$, then define

$$I_1 := I \setminus \{v_{i(1)}\} \cup \{v_{i(1)+1}, \dots, v_m\}, \dots, I_{n+1} := I \setminus \{v_{i(n+1)}\} \cup \{v_{i(n+1)+1}, \dots, v_m\}.$$

By definition $t_{I_1}, \dots, t_{I_{n+1}} <_{\text{lex}} t_I$. Hence, by induction assumption, there are $\mathbb{C}_{I_1}, \dots, \mathbb{C}_{I_{n+1}}$ witnessing that $I_1, \dots, I_{n+1} \in \mathcal{I}$, respectively. Let

$$(a_0^1, \dots, a_{m(1)}^1), \dots, (a_0^{n+1}, \dots, a_{m(n+1)}^{n+1})$$

be the spines of $\mathbb{C}_{I_1}, \dots, \mathbb{C}_{I_{n+1}}$, respectively. For $k \in [2, n+1]$ let $i(k, 1)$ be the smallest number in $[m(k)]$ such that $a_{i(k, 1)}^k = a_{i(1)}$. Observe that $\mathbb{C}_{I_k}[0, i(k, 1)]$ is an unfolding of $\mathbb{C}[0, i(1)]$ that fixes $v_{i(1)}$. By Lemma B.5 there is a caterpillar that is a common unfolding of $\mathbb{C}_{I_2}[0, i(2, 1)], \dots, \mathbb{C}_{I_{n+1}}[0, i(n+1, 1)]$ fixing $v_{i(1)}$. We denote this caterpillar by $\mathbb{C}'[0, i'(1)]$, where $i'(1)$ is the number of vertebrae of $\mathbb{C}'[0, i'(1)]$. Note that $\mathbb{C}'[0, i'(1)]$ is in particular an unfolding of $\mathbb{C}[0, i(1)]$ that fixes $v_{i(1)}$.

Let $i(1, 1)$ be the biggest number in $[m(1)]$ such that $a_{i(1, 1)}^1 = a_{i(1)}$. Since $\mathbb{C}_{I_1}$ is an unfolding that fixes $v_{i(1)+1}$ we have that $a_{i(1)+1}$ does not occur in $\mathbb{C}_{I_1}[0, i(1, 1)]$. Hence, $\mathbb{C}_{I_1}[0, i(1, 1)]$ is an unfolding of $\mathbb{C}[0, i(1)]$ (that does not necessarily fix $v_{i(1)}$). By Lemma B.5 there is a caterpillar $\mathbb{C}''[0, i''(1)]$ that is a common unfolding of $\mathbb{C}_{I_1}[0, i(1, 1)]$ and $\mathbb{C}'[0, i'(1)]$, where $i''(1)$ is the number of vertebrae of $\mathbb{C}''[0, i''(1)]$. We continue in this fashion to define caterpillars $\mathbb{C}'[i'(1), i'(2)], \dots, \mathbb{C}'[i'(n+1), m]$ and $\mathbb{C}''[i''(1), i''(2)], \dots, \mathbb{C}''[i''(n+1), m]$, where analogous to the case $\ell = 1$ we have that $i(k, \ell)$ is the smallest number in $[m(k)]$ such that $a_{i(k, \ell)}^k = a_{i(\ell)}$ if $\ell \neq k$ and it is the biggest such number if $k = \ell$.

Observe that for $k \neq \ell$ the fact that $\mathbb{C}_{I_k}$ fixes $v_{i(\ell-1)}$ and $v_{i(\ell)}$ implies that $\mathbb{C}'[i'(k, \ell-1), i'(k, \ell)]$ is an unfolding of $\mathbb{C}_{I_k}[i(\ell-1), i(\ell)]$ that fixes $v_{i(\ell)}$. Similarly, the fact that $\mathbb{C}_{I_k}$ fixes $v_{i(k-1)}$ and $v_{i(k)+1}$ implies that $\mathbb{C}''[i''(k, k-1), i''(k, k)]$ is an unfolding of $\mathbb{C}_{I_k}[i(k-1), i(k)]$.

Note that $\mathbb{C}'[0, i'(1)], \dots, \mathbb{C}'[i'(n+1), m]$ uniquely defines a caterpillar $\mathbb{C}'$. Analogously, $\mathbb{C}''[0, i''(1)], \dots, \mathbb{C}''[i''(n+1), m]$ defines $\mathbb{C}''$. The caterpillar $\mathbb{C}'$ is an unfolding of $\mathbb{C}$ that fixes all vertebrae in I . For $k \in [n+1]$ the caterpillar $\mathbb{C}'_k$ is obtained from $\mathbb{C}'$ by replacing the interval $\mathbb{C}'[i'(k, k-1), i'(k, k)]$ by $\mathbb{C}''[i''(k, k-1), i''(k, k)]$. Observe that $\mathbb{C}'_k$ is an unfolding of $\mathbb{C}_{I_k}$ (that fixes $I \setminus \{v_k\}$). Hence $\mathbb{C}'_k \rightarrow \mathbb{C}_{I_k} \rightarrow \mathbb{B}$. By definition $\mathbb{C}'_k$ is an unfolding of $\mathbb{C}'$ that fixes all vertebrae of $\mathbb{C}'$ that are not in the interval $\mathbb{C}'[i'(k, k-1), i'(k, k)]$. Hence, Lemma B.4 implies $\mathbb{C}' \rightarrow \mathbb{B}$. Therefore, $I \in \mathcal{I}$.

This induction proof shows in particular that there is an unfolding $\mathbb{C}_V$ of $\mathbb{C}$ fixing all vertebrae in V witnessing that $V \in \mathcal{I}$. By definition of $\mathcal{I}$ there is a homomorphism from $\mathbb{C}_V$ to $\mathbb{B}$. Note that the only unfolding of $\mathbb{C}$ that fixes all vertebrae of $\mathbb{C}$ is $\mathbb{C}$ itself. Hence, $\mathbb{C}_V = \mathbb{C}$ and $\mathbb{C} \rightarrow \mathbb{B}$, as desired. $\square$

C. Monadic Datalog

Let $\mathbb{B}$ be a structure and Π be a monadic Datalog program that solves $\text{CSP}(\mathbb{B})$. It seems intuitively clear that for every structure $\mathbb{A}$ with $\mathbb{A} \not\rightarrow \mathbb{B}$ there is a derivation of Π that derives the goal predicate and is in some sense connected. In this section we will formalize this intuition. A rule $\phi_0 :- \phi$ of a Datalog program is called *injective* if the τ -reduct of the canonical database of ϕ is injective, where τ is the set of EDBs. A rule $\phi_0 :- \phi$ is called *connected* if

- the canonical databases of ϕ is connected,
- every variable in ϕ_0 occurs in ϕ ,
- there is no 0-ary IDB occurring in ϕ , and
- if ϕ_0 is 0-ary, then ϕ_0 is the goal predicate.

Let Π be a Datalog program and $\mathbb{A}$ be a finite structure. A derivation $\mathbb{A}_0 \vdash_{R_0} \dots \vdash_{R_{n-1}} \mathbb{A}_n$ of Π on $\mathbb{A}$ that derives the IDB P on the tuple t is *inefficient* if there is an $m < n$, $0 \leq i_1 < \dots < i_{m-1} < n$ and a derivation $\mathbb{A}'_0 \vdash_{R_{i_0}} \dots \vdash_{R_{i_{m-1}}} \mathbb{A}'_m$ of Π on $\mathbb{A}$ that also derives P on t . A derivation is called *efficient* if it is not inefficient. Note that for every derivation of Π on $\mathbb{A}$ deriving P on t there is an efficient derivation of Π on $\mathbb{A}$ deriving P on t . It is clear from these definitions that the following is true.

Lemma C.1. *Let Π be a Datalog program and $\mathbb{A}$ be a finite structure. If $\mathbb{A}_0 \vdash_{R_0} \dots \vdash_{R_{n-1}} \mathbb{A}_n$ is an efficient derivation of Π on $\mathbb{A}$ with connected rules $R_0, \dots, R_{n-1}$, then there is a set $S \subseteq A$ such that*

- $\mathbb{A}[S]$ is connected and
- $\mathbb{A}_0[S] \vdash_{R_0} \dots \vdash_{R_{n-1}} \mathbb{A}_n[S]$ is a derivation of Π on $\mathbb{A}[S]$,

where $\mathbb{A}[S]$ is the substructure of $\mathbb{A}$ induced by S .

We show that a monadic Datalog program that solves a CSP of a finite structure can basically always restrict to injective and connected rules, when deriving the goal predicate.

Lemma C.2. *Let $\mathbb{B}$ be a finite structure and Π be a Datalog program that solves $\text{CSP}(\mathbb{B})$. Let Π' be the Datalog program obtained from Π by removing all non-injective rules. Then Π' solves $\text{CSP}(\mathbb{B})$.*

Proof. Let $\mathbb{A}$ be an instance of $\text{CSP}(\mathbb{B})$. First, suppose $\mathbb{A} \rightarrow \mathbb{B}$. Since Π solves $\text{CSP}(\mathbb{B})$, it can not derive the goal predicate on $\mathbb{A}$. As Π' has fewer rules than Π , it also can not derive the goal predicate on $\mathbb{A}$. Now consider the case $\mathbb{A} \not\rightarrow \mathbb{B}$. Clearly, the set of all finite structures $\mathbb{A}'$ with $\mathbb{A}' \not\rightarrow \mathbb{B}$ is an obstruction set for $\mathbb{B}$. Hence, by Lemma 2.9, there exists an injective structure $\mathbb{A}'$ with $\mathbb{A}' \rightarrow \mathbb{A}$ and $\mathbb{A}' \not\rightarrow \mathbb{B}$. Since Π solves $\text{CSP}(\mathbb{B})$, it can derive the goal predicate on $\mathbb{A}'$. Note that, since $\mathbb{A}'$ is injective, all rules of this derivation are injective. Hence, this derivation is also a derivation of Π' that derives the goal predicate. Applying the homomorphism from $\mathbb{A}'$ to $\mathbb{A}$ to this derivation yields a derivation of Π' on $\mathbb{A}$ that derives the goal predicate. $\square$

Lemma C.3. *Let $\mathbb{B}$ be a (not necessarily finite) structure and Π be a monadic Datalog program that solves $\text{CSP}(\mathbb{B})$. Then there is a monadic Datalog program Π' such that*

- Π' solves $\text{CSP}(\mathbb{B})$,
- Π' is obtained from Π replacing each disconnected rule $\phi_0 :- \phi_1 \wedge \dots \wedge \phi_n$ with a suitable set of connected rules of the forms

$$\phi_0 :- \phi_{i_1} \wedge \dots \wedge \phi_{i_m} \quad \text{and} \quad G :- \phi_{i_1} \wedge \dots \wedge \phi_{i_m}.$$

If Π is symmetric, then the suitable sets can be chosen in such a way that Π' is also symmetric.

Proof. We iteratively construct Datalog programs Π_i that solves $\text{CSP}(\mathbb{B})$. Let $\Pi_0 = \Pi$. For $i \geq 0$ we construct Π_{i+1} as follows. Define $\Pi_{i+1} := \Pi_i$ if there is no finite structure $\mathbb{A}$ with $\mathbb{A} \not\rightarrow \mathbb{B}$ such that every derivation of Π_i on $\mathbb{A}$ that derives the goal predicate uses a disconnected rule. Otherwise, let $\mathbb{A}$ be such a structure and let $\mathbb{A}_0 \vdash_{R_0} \dots \vdash_{R_n} \mathbb{A}_{n+1}$ be a derivation of Π_i that

- derives the goal predicate on $\mathbb{A}$,
- uses a disconnected rule R_k , and
- among all derivations of Π_i on $\mathbb{A}$ that derive the goal predicate, minimizes the number of applications of disconnected rules.

Let R_k be of the form $\phi_0 :- \phi$. Let $\mathbb{R}$ denote the canonical database of ϕ . Let ψ be the empty conjunction if $\phi_0 = G$, otherwise ϕ_0 is of the form $P(x)$ and we define ψ to be the canonical conjunctive query of the connected component of $\mathbb{R}$ that contains x . Let $\psi_1, \dots, \psi_m$ be the canonical conjunctive queries of the remaining connected components of $\mathbb{R}$. The formulas $\psi, \psi_1, \dots, \psi_m$ partition the conjuncts of ϕ . We distinguish two cases.

First, assume that for every $j \in [m]$, there exists a finite structure $\mathbb{D}_j$ such that $\mathbb{D}_j \rightarrow \mathbb{B}$ and there is a derivation of Π_i on $\mathbb{D}_j$ that results in a structure $\mathbb{C}_j$ to which the canonical database of ψ_j admits a homomorphism. Then Π_{i+1} is obtained from Π_i by adding the connected rule $\phi_0 :- \psi$, which we denote by $\tilde{R}_k$. This rule was not already a rule of Π_i , since otherwise

$$\mathbb{A}_0 \vdash_{R_0} \dots \vdash_{R_{k-1}} \mathbb{A}_k \vdash_{\tilde{R}_k} \mathbb{A}_{k+1} \vdash_{R_{k+1}} \dots \vdash_{R_n} \mathbb{A}_{n+1}$$

would be a derivation of Π_i that derives the goal predicate on $\mathbb{A}$ and uses one fewer disconnected rules than the original derivation, contradicting its minimality. Assume there exists a structure $\mathbb{A}'$ on which Π_{i+1} can derive the goal predicate and Π_i can not. Since Π_i solves $\text{CSP}(\mathbb{B})$, we have $\mathbb{A}' \rightarrow \mathbb{B}$. It is easy to see that Π_i can derive the goal predicate on the disjoint union of $\mathbb{D}_1, \dots, \mathbb{D}_m$, and $\mathbb{A}'$ by replacing $\tilde{R}_k$ with R_k in the derivation of Π_{i+1} and, if necessary, adding parts of the original derivation. Since each of these structures admits a homomorphism to $\mathbb{B}$ so does their disjoint union. This contradicts the fact that Π_i solves $\text{CSP}(\mathbb{B})$. Hence, such an $\mathbb{A}'$ can not exist, and therefore Π_{i+1} solves $\text{CSP}(\mathbb{B})$. Note that if $\phi_0 = G$ we can not be in this case.

In the second case there is a j for which no such structure $\mathbb{D}_j$ exists. Then Π_{i+1} is obtained from Π_i by adding the connected rule $G :- \psi_j$, which we denote by $\tilde{R}_k$. Again, this rule was not already a rule of Π_i , since otherwise

$$\mathbb{A}_0 \vdash_{R_0} \dots \vdash_{R_{k-1}} \mathbb{A}_k \vdash_{\tilde{R}_k} \tilde{\mathbb{A}}_{k+1}$$

where $\tilde{\mathbb{A}}_{k+1}$ is obtained from $\mathbb{A}_k$ by adding $()$ to $G^{\mathbb{A}_k}$, would be a derivation of Π_i that derives the goal predicate on $\mathbb{A}$ that uses at least one fewer disconnected rules than the original derivation, again contradicting its minimality. Assume that there is a structure $\mathbb{A}'$ on which Π_{i+1} can derive the goal predicate and Π_i can not. Since Π_i solves $\text{CSP}(\mathbb{B})$ we have $\mathbb{A}' \rightarrow \mathbb{B}$. Clearly, Π_{i+1} must use $\tilde{R}_k$ in its derivation of the goal predicate. Therefore, there is a derivation of Π_i on $\mathbb{A}'$ such that the canonical database of ψ_j has a homomorphism into the derived structure. A contradiction. Therefore, such an $\mathbb{A}'$ can not exist, and hence Π_{i+1} solves $\text{CSP}(\mathbb{B})$.

Note that all rules we add in this process are connected. Therefore, the disconnected rules do not change and there are only finitely many rules that we could add. Hence, the process eventually stabilizes, and there exists an i for which the Datalog program Π_i never has to use any of its disconnected rules to derive the goal predicate. We obtain the desired program Π' by removing all disconnected rules from Π_i .

If Π is symmetric we can modify this procedure. Whenever we add a rule in the first case with an IDB in its body we also add the reversed rule. Since a symmetric Datalog programs does not need to contain the reverse of rules of the form $G :- \psi_j$, we have that Π_i is symmetric. The argument that Π_i solves $\text{CSP}(\mathbb{B})$ is analogous to the original argument. We replace R_k and $\tilde{R}_k$ by their reverse rule and $\mathbb{D}_1, \dots, \mathbb{D}_m$ stay the same. $\square$

D. Symmetrizing Linear Arc Monadic Datalog

The following section is devoted to the implication from (5) to (3) in the proof of Theorem 4.1. Before we come to Datalog we introduce an alternative way of defining unfoldings. Let $\mathbb{C}_1$ and $\mathbb{C}_2$ be caterpillars with spines $(a_0^1, \dots, a_{m(1)}^1)$ and $(a_0^2, \dots, a_{m(2)}^2)$, respectively. The *concatenation of $\mathbb{C}_1$ and $\mathbb{C}_2$* , denoted $\mathbb{C}_1 \circ \mathbb{C}_2$, is the caterpillar obtained from the disjoint union of $\mathbb{C}_1$ and $\mathbb{C}_2$ by identifying $a_{m(1)}^1$ and a_0^2 . Note that for a caterpillar $\mathbb{C}$ with spine $(a_0, \dots, a_n)$ and $i, j \in [0, n]$ with $i < j$ the (a_i, a_j) -unfolding of $\mathbb{C}$ is isomorphic to $\mathbb{C}[0, j] \circ \mathbb{C}(j, i) \circ \mathbb{C}(i, n]$.

Let $\mathbb{C}$ be a caterpillar with conjunctive query representation $(\phi \wedge \psi)(a_0, \dots, a_n)$ and let $n_1, \dots, n_k \in [0, n]$ with k even and $0 < n_1 > n_2 < n_3 > n_4 < \dots > n_k < n$. The $(n_1, \dots, n_k)$ -unfolding of $\mathbb{C}$ is $\mathbb{C}[0, n_1] \circ \mathbb{C}(n_1, n_2) \circ \mathbb{C}(n_2, n_3) \circ \mathbb{C}(n_3, n_4) \circ \dots \circ \mathbb{C}(n_k, n]$. This unfolding fixes (a_i, a_{i+1}) if there is an $\ell \in [0, k]$ such that $n_1, \dots, n_\ell \leq i$ and $n_{\ell+1}, \dots, n_k \geq i + 1$. Note that for $i, j \in [0, n]$ with $i < j$ the (j, i) -unfolding of $\mathbb{C}$ is isomorphic to the (a_i, a_j) -unfolding of $\mathbb{C}$ and that for both notions of unfolding the set of fixed vertebrae is the same.

Lemma D.1. *Let $\mathbb{C}$ be a caterpillar and $\mathbb{C}'$ be the $(n_1, \dots, n_k)$ -unfolding of $\mathbb{C}$ fixing $v_1, \dots, v_n$, then $\mathbb{C}'$ is isomorphic to an unfolding of $\mathbb{C}$ fixing $v_1, \dots, v_n$.*

Proof. Let $(a_0, \dots, a_m)$ be the spine of $\mathbb{C}$. If $k = 0$, then $\mathbb{C}' = \mathbb{C}[0, m] = \mathbb{C}$ is an unfolding of $\mathbb{C}$ fixing $v_1, \dots, v_n$. If $k > 0$, then let $d_0 = n_1, d_1 = |n_2 - n_1|, \dots, d_{k-1} = |n_k - n_{k-1}|, d_k = m - n_k$. Note that if there is an $\ell \in [1, k-1]$ with $d_{\ell-1} \geq d_\ell \leq d_{\ell+1}$, then $\mathbb{C}'$ is isomorphic to the $(a_{d_0+\dots+d_{\ell-1}-d_\ell}^{\prime\prime}, a_{d_0+\dots+d_{\ell-1}}^{\prime\prime})$ -unfolding of $\mathbb{C}''$, where $\mathbb{C}''$ is the $(n_1, \dots, n_{\ell-1}, n_{\ell+2}, \dots, n_k)$ -unfolding of $\mathbb{C}$ and $(a_0^{\prime\prime}, \dots, a_{m''}^{\prime\prime})$ the spine of $\mathbb{C}''$. Clearly, both unfoldings fix $v_1, \dots, v_n$. By induction hypothesis $\mathbb{C}''$ is isomorphic to an unfolding of $\mathbb{C}$ fixing $v_1, \dots, v_n$. Hence, by definition of unfoldings, $\mathbb{C}'$ is isomorphic to an unfolding of $\mathbb{C}$ fixing $v_1, \dots, v_n$. It remains to be shown that such an ℓ exists. Assume that there is no such ℓ in $[1, k-1]$. Then $d_\ell > \min(d_{\ell-1}, d_{\ell+1})$ for all $\ell \in [1, k-1]$. Recall that $n_2 \leq n_1$, hence, $d_0 = n_1 \geq d_1 > d_2$. Now $d_2 > \min(d_1, d_3)$ implies $d_2 > d_3$. Analogously, $d_3 > \dots > d_k$. Since k is even we have $n_k < n_{k-1}$. Therefore $d_k = m - n_k \geq n_{k-1} - n_k = |n_{k-1} - n_k| = d_{k-1}$, a contradiction. $\square$

The converse is also true but we do not need it for the following proofs. Our goal in this section will be to show that for a structure $\mathbb{B}$ with caterpillar duality such that $\overline{\text{CSP}(\mathbb{B})}$ is closed under n -fixed unfoldings of caterpillars we have that the canonical lam Datalog program Π for $\mathbb{B}$ and the canonical s_n lam Datalog program Π_S for $\mathbb{B}$ derive the goal predicate on the same instances. Note that when Π is deriving the goal predicate we can use the ‘strongest possible rules’, i.e., if we have different rules with the same body we always use the one with the smallest (with respect to inclusion) IDB in the head. However, Π_S might need to use weaker rules in order to be able to apply symmetric rules later on in the derivation. See Example 3.6 in [2] for a derivation where this is necessary. Before we come the the proof we take a look at the close link of derivations of canonical lam Datalog programs and caterpillars.

Observation D.2. Let Π be a linear arc monadic Datalog program that solves the CSP of a finite structure $\mathbb{B}$ and $\mathbb{A}$ be a structure on which Π can derive the goal predicate. Let $R_0, \dots, R_{n+1}$ be the rules of Π used in such an efficient derivation. Since Π is linear, we can use the suggestive notation

$$\vdash_{R_0} P_0(a_0) \vdash_{R_1} \dots \vdash_{R_n} P_n(a_n) \vdash_{R_{n+1}} G$$

for this derivation. By Lemmata C.2 and C.3 we can assume without loss of generality that all rules used in this derivation are injective and connected. The rule R_0 is of the form $P_0(x) :- \Psi_0(x_1, \dots, x_k)$ and R_{n+1} is of the form $G :- P_n(x) \wedge \Psi_{n+1}(x_1, \dots, x_k)$, where Ψ_0 and Ψ_{n+1} are atomic formulas. Let $\Phi_0(x)$ and $\Phi_{n+1}(x)$ be obtained from Ψ_0 and Ψ_{n+1} , respectively, by existentially quantifying all variables except for x . Note that, since the derivation is efficient, any other rule R_i is either of the form $P_i(x) :- P_{i-1}(y) \wedge \Psi_i(x_1, \dots, x_k)$ with $x \neq y$ or $P_i(x) :- P_{i-1}(x) \wedge \Psi_i(x_1, \dots, x_k)$. Analogously to $\Phi_0(x)$ and $\Phi_{n+1}(x)$ define the formula $\Phi_i(x, y)$ or $\Phi_i(x)$. Let $0 < i_0 < \dots < i_{m-1} < n+1$ such that $\Phi_{i_0}, \dots, \Phi_{i_{m-1}}$ are exactly the formulas with two free variables. For $\ell \in [0, m-1]$ define $\phi_\ell(x, y)$ as $\Phi_{i_\ell}(x, y)$. For $\ell \in [0, m]$ define $\psi_\ell(x)$ as $\Phi_{i_{\ell-1}+1}(x) \wedge \dots \wedge \Phi_{i_\ell-1}(x)$, where $i_{-1} = -1$ and $i_m = n+2$. Note that ψ_ℓ can be the empty conjunction. Observe that

$$\phi_0(x_0, x_1) \wedge \dots \wedge \phi_{m-1}(x_{m-1}, x_m) \wedge \psi_0(x_0) \wedge \dots \wedge \psi_m(x_m)$$

is the conjunctive query representation of a caterpillar $\mathbb{C}$. Note that this construction is essentially a proof of (2) $\Rightarrow$ (1) in Theorem 2.8. We will use the construction for something different. For $\ell \in [0, m-1]$ define S_ℓ as the rule R_{i_ℓ} and for $\ell \in [1, m-1]$ define H_ℓ as a rule with head $P_{i_\ell}(x)$ and body the canonical conjunctive query of the canonical database of $P_{i_{\ell-1}+1}(x) \wedge \psi_\ell(x)$. The rules H_0 and H_m are defined analogously. Observe that adding the rules $H_0, \dots, H_m$ to Π does not change the set of structures on which Π can derive the goal predicate. Hence, we can without loss of generality say that

$$\vdash_{H_0} \tilde{P}_0(\tilde{a}_0) \vdash_{S_0} \tilde{Q}_1(\tilde{a}_1) \vdash_{H_1} \tilde{P}_1(\tilde{a}_1) \vdash_{S_1} \dots \vdash_{H_{m-1}} \tilde{P}_{m-1}(\tilde{a}_{m-1}) \vdash_{S_{m-1}} \tilde{Q}_m(\tilde{a}_m) \vdash_{H_m} G$$

is a derivation of Π on $\mathbb{A}$ that derives the goal predicate, where $\tilde{P}_\ell := P_{i_\ell}$, $\tilde{Q}_\ell := P_{i_{\ell-1}+1}$, $\tilde{a}_\ell := a_{i_\ell}$, and $\tilde{a}_m := a_{i_{m-1}+1}$. The advantage of the new derivation is that it is easier to translate between the derivation and the corresponding caterpillar $\mathbb{C}$.

Now we are ready to prove that we can transform derivations of Π into derivations of Π_S .

Lemma D.3. *Let $\mathbb{B}$ be a finite structure with relational signature τ such that $\mathbb{B}$ has caterpillar duality and $\overline{\text{CSP}}(\mathbb{B})$ is closed under n -fixed unfoldings of caterpillars. Let Π be the canonical linear arc monadic Datalog program for $\mathbb{B}$ and Π_S be the canonical s_n lam Datalog program for $\mathbb{B}$. Then Π can derive the goal predicate on a finite τ -structure $\mathbb{A}$ if and only if Π_S can derive the goal predicate on $\mathbb{A}$.*

The proof is a generalization of the proof of Lemma 3.4 in [2].

Proof. Note that every rule of Π_S is also a rule of Π . Hence if Π_S can derive the goal predicate on $\mathbb{A}$, then so can Π . Consider a derivation of the goal predicate G for the Datalog program Π on $\mathbb{A}$. Let $R_0, \dots, R_{m+1}$ be the rules of Π used in this derivation. We assume that the derivation has the form discussed in Observation D.2

$$\vdash_{H_0} P_0(a_0) \vdash_{S_0} Q_1(a_1) \vdash_{H_1} P_1(a_1) \vdash_{S_1} \dots \vdash_{H_{m-1}} P_{m-1}(a_{m-1}) \vdash_{S_{m-1}} Q_m(a_m) \vdash_{H_m} G$$

and that $\phi_0(x_1, x_2), \dots, \phi_{m-1}(x_{m-1}, x_m)$ and $\psi_0(x_0), \dots, \psi_m(x_m)$ are the formulas corresponding to $S_0, \dots, S_{m-1}$ and $H_0, \dots, H_m$, respectively. Since Π is canonical we may assume without loss of generality that all rules strongest possible, i.e.,

$$(C1) \ P_0 = \{b \in B \mid \mathbb{B} \models \Phi_0(b)\} = \Phi_0^{\mathbb{B}},$$

$$(C2) \ P_i = \{b \in Q_i \mid \mathbb{B} \models \psi_i(b)\} \text{ for all } i \in [m-1], \text{ and}$$

(C3) $Q_i = \{b' \in B \mid \exists b \in P_{i-1} : \mathbb{B} \models \phi_{i-1}(b, b')\}$ for all $i \in [m]$.

Note that $\phi_0(x_1, x_2) \wedge \dots \wedge \phi_{m-1}(x_{m-1}, x_m) \wedge \psi_0(x_0) \wedge \dots \wedge \psi_m(x_m)$ is the conjunctive query representation of a caterpillar $\mathbb{C}$ with spine $(x_0, \dots, x_m)$. Replacing a_i by x_i in the above derivation we obtain a derivation of Π that derives the goal predicate on $\mathbb{C}$. Therefore $\mathbb{C} \not\models \mathbb{B}$. By assumption there are vertebrae $v_1, \dots, v_n$ such that no unfolding of $\mathbb{C}$ that fixes $v_1, \dots, v_n$ has a homomorphism to $\mathbb{B}$.

Let $\tilde{Q}_1, \dots, \tilde{Q}_m, \tilde{P}_0, \dots, \tilde{P}_{m-1} \subseteq B$ be the smallest sets such that

(P) $P_i \subseteq \tilde{P}_i$ and $Q_{i+1} \subseteq \tilde{Q}_{i+1}$ for all $i \in [m-1]$,

(H1) $b \in \tilde{Q}_i$ and $\mathbb{B} \models \psi_i(b)$ implies $b \in \tilde{P}_i$ for all $i \in [m-1]$,

(H2) $b \in \tilde{P}_i$ and $\mathbb{B} \models \psi_i(b)$ implies $b \in \tilde{Q}_i$ for all $i \in [m-1]$,

(S1) $b \in \tilde{P}_i$ and $\mathbb{B} \models \phi_i(b, b')$ implies $b' \in \tilde{Q}_{i+1}$ for all $i \in [0, m-1]$, and

(S2) $b' \in \tilde{Q}_{i+1}$ and $\mathbb{B} \models \phi_i(b, b')$ implies $b \in \tilde{P}_i$ for all $i \in [0, m-1]$ with $(x_i, x_{i+1}) \notin \{v_1, \dots, v_n\}$.

Let $\tilde{H}_0, \dots, \tilde{H}_m, \tilde{S}_0, \dots, \tilde{S}_{m-1}$ be the rules obtained from $H_0, \dots, H_m, S_0, \dots, S_{m-1}$ by replacing each occurrence of P_i and Q_i by $\tilde{P}_i$ and $\tilde{Q}_i$, respectively. We will now show that

$$\vdash_{\tilde{H}_0} \tilde{P}_0(a_0) \vdash_{\tilde{S}_0} \tilde{Q}_1(a_1) \vdash_{\tilde{H}_1} \tilde{P}_1(a_1) \vdash_{\tilde{S}_1} \dots \vdash_{\tilde{H}_{m-1}} \tilde{P}_{m-1}(a_{m-1}) \vdash_{\tilde{S}_{m-1}} \tilde{Q}_m(a_m) \vdash_{\tilde{H}_m} G$$

is a derivation of Π_S that derives the goal predicate on $\mathbb{A}$. Recall that Π_S consists of $n+1$ many copies of the canonical slam Datalog program that are connected by rules of the canonical lam Datalog program. Hence, strictly speaking, renaming the IDBs is necessary to obtain rules in Π_S . We will omit giving the renaming explicitly and instead show that $\tilde{H}_0, \dots, \tilde{H}_m, \tilde{S}_0, \dots, \tilde{S}_{m-1}$ are rules of Π (the canonical lam Datalog program for $\mathbb{B}$) and that all but at most n of those rules are symmetric. By definition $P_0 \subseteq \tilde{P}_0$. Hence $\mathbb{B} \models \Phi_0(b)$ implies $b \in \tilde{P}_0$ and $\tilde{H}_0$ is a rule of Π . Since there is no IDB in its body the rule is also symmetric. Let $i \in [0, m-1]$. By (S1) $\tilde{S}_i$ is a rule of Π . If $(x_i, x_{i+1}) \notin \{v_1, \dots, v_n\}$, then, by (S2), the reversed rule of S_i is also a rule of Π . Hence all but at most n of the rules $\tilde{S}_0, \dots, \tilde{S}_{m-1}$ are symmetric rules of Π . Let $i \in [m-1]$. By (H1) and (H2) $\tilde{H}_i$ and its reverse are rules of Π . Finally we need to show that $\tilde{H}_m$ is a symmetric rule of Π . Note that since it derives the goal predicate it automatically is a symmetric rule. We need to show that there is no $b \in \tilde{Q}_m$ such that $\mathbb{B} \models \psi_m(b)$. Assume that such a b exists. Then $b \in \tilde{Q}_m$ because of (P) or because of (S1). By (C3), in both cases there is a $b_M \in \tilde{P}_{m-1}$ such that $\phi_{m-1}(b_M, b)$. The element b_M was added to $\tilde{P}_{m-1}$ because of

- (P) in which case, by (C2), $\mathbb{B} \models \psi_{m-1}(b_M)$,
- (H1) in which case $\mathbb{B} \models \psi_{m-1}(b_M)$, or
- (S2) in which case there exists $b_{M-1} \in Q_m$ with $\mathbb{B} \models \phi_{m-1}(b_{M-1}, b_M)$.

Continuing in this manner we can construct a path $b_M, \dots, b_0$ to an element $b_0 \in \tilde{P}_0$. Along with constructing this path we also construct the map $\iota : [0, M] \rightarrow [0, m]$ to store the information from which P_ℓ or Q_ℓ the element b_ℓ was taken. Note that ι satisfies $b_\ell \in P_{\iota(\ell)}$ or $b_\ell \in Q_{\iota(\ell)}$ for all ℓ . Let $0 < n_1 < \dots < n_k < M$ be all numbers ℓ such that the sequence $(\iota(\ell-1), \iota(\ell), \iota(\ell+1))$ is not monotone. Note that $\iota(n_1), \dots, \iota(n_k)$ mark the orientation changes of the path $b_0, \dots, b_M, b$. This path yields an $(\iota(n_1), \dots, \iota(n_k))$ -unfolding $\mathbb{C}'$ of $\mathbb{C}$. Observe that, by (S2), $\mathbb{C}'$ fixes $v_1, \dots, v_n$. By Lemma D.1 $\mathbb{C}'$ is an unfolding of $\mathbb{C}$ that fixes $v_1, \dots, v_n$ with spine $b_0, \dots, b_M, b$ such that each b_j is an element of the corresponding $\tilde{P}_i$ and/or $\tilde{Q}_i$. By construction the map $\mathbb{C}' \rightarrow \mathbb{B}, b_j \mapsto b_j, b \mapsto b$ can be extended to a homomorphism. A contradiction. Therefore such a b can not exist and $\tilde{H}_m$ is a rule of Π . In conclusion, Π_S can derive the goal predicate on $\mathbb{A}$ as desired. $\square$

E. Going back from Datalog to Caterpillars

Let Π be a lam Datalog program, $\mathbb{C}$ be a caterpillar, $v = (a, b)$ be a vertebra of $\mathbb{C}$, and let $t \in R^{\mathbb{C}}$ be the unique R and t for which a, b occur in t . A derivation of Π applies non-symmetric rule to v if the derivation contains a rule R that can only be applied because its EDB matches $R(t)$ and the reversed rule $\bar{R}$ is not a rule of Π .

Lemma E.1. *Let Π be a lam Datalog program and $\mathbb{C}$ be a caterpillar. If Π can derive the goal predicate on $\mathbb{C}$ and $v_1, \dots, v_n$ are the vertebrae of $\mathbb{C}$ to which this derivation applied non-symmetric rules, then Π can derive the goal predicate on any unfolding of $\mathbb{C}$ that fixes $v_1, \dots, v_n$.*

Proof. Let $(a_0, \dots, a_m)$ be the spine of $\mathbb{C}$ and $\phi \wedge \psi$ be the conjunctive query representation of $\mathbb{C}$. Consider a derivation of Π that derives the goal predicate on $\mathbb{C}$. Let $R_0, \dots, R_{k+1}$ be the rules of Π used in this derivation. Since Π is linear and monadic, we can use the notation

$$\vdash_{R_0} P_0(c_0) \vdash_{R_1} \dots \vdash_{R_k} P_k(c_k) \vdash_{R_{k+1}} G$$

for this derivation. Let $\mathbb{C}'$ be an (a, b) -unfolding of $\mathbb{C}$ such that no $v_1, \dots, v_n$ lies between a and b . We will show that Π can derive the goal predicate on $\mathbb{C}'$ without applying non-symmetric rules to $v_1, \dots, v_n$. The lemma then follows by induction.

Note that if $P_\ell(c_\ell) \vdash_{R_{\ell+1}} P_{\ell+1}(c_{\ell+1}) \vdash_{R_{\ell+2}} P_{\ell+2}(c_{\ell+2})$ and there is no relation in $\mathbb{C}$ that contains a tuple which contains both c_ℓ and $c_{\ell+2}$, then $c_{\ell+1}$ is a joint. Hence the derivation can not jump over joints, i.e., for all $i < j$ and all joints c between c_i and c_j there is a $\ell \in [i, j]$ with $c_\ell = c$. Consider an interval

$$P_i(c_i) \vdash_{R_{i+1}} \dots \vdash_{R_j} P_j(c_j)$$

of the derivation on $\mathbb{C}$ such that $\{c_i, c_j\} = \{a, b\}$ and $c_{i+1}, \dots, c_{j-1} \notin \{a, b\}$. If $c_i = a$ and $c_j = b$, then replace this interval with

$$\begin{aligned} & P_i(a) \vdash_{R_{i+1}} P_{i+1}(c'_{i+1}) \vdash_{R_{i+2}} \dots \vdash_{R_{j-1}} P_{j-1}(c'_{j-1}) \vdash_{R_j} P_j(b') \\ & P_i(a'') \dashv_{\tilde{R}_{i+1}} P_{i+1}(c_{i+1}) \dashv_{\tilde{R}_{i+2}} \dots \dashv_{\tilde{R}_{j-1}} P_{j-1}(c_{j-1}) \dashv_{\tilde{R}_j} P_j(b') \\ & P_i(a'') \vdash_{R_{i+1}} P_{i+1}(c''_{i+1}) \vdash_{R_{i+2}} \dots \vdash_{R_{j-1}} P_{j-1}(c''_{j-1}) \vdash_{R_j} P_j(b) \end{aligned}$$

where $\tilde{R}_\ell$ is the reversed rule of R_ℓ and c' and c'' come from the naming convention fixed in Definition 2.5. Since no $v_1, \dots, v_n$ lies between a and b the rules $R_{i+1}, \dots, R_j$ are symmetric. Hence $\tilde{R}_{i+1}, \dots, \tilde{R}_j$ are also rules of Π . If $c_i = b$ and $c_j = a$, then we replace the interval analogously. The resulting derivation is a derivation of Π on $\mathbb{C}'$ that derives the goal predicate and never applies a non-symmetric rule to $v_1, \dots, v_n$. $\square$

Lemma E.2. *If $\text{CSP}(\mathbb{B})$ is solved by a s_n lam Datalog program Π , then $\mathbb{B}$ has caterpillar duality and $\overline{\text{CSP}(\mathbb{B})}$ is closed under n -fixed unfoldings of caterpillars.*

Proof. The Datalog program Π is, in particular, linear and monadic. Hence, by Theorem 2.8, $\mathbb{B}$ has caterpillar duality. Let $\mathbb{C}$ be a caterpillar that has no homomorphism to $\mathbb{B}$. Since Π solves $\text{CSP}(\mathbb{B})$, there exists a derivation of Π on $\mathbb{C}$ that derives the goal predicate. Let $v_1, \dots, v_m$ be the vertebrae of $\mathbb{C}$ to which non-symmetric rules are applied. By Lemma E.1, every unfolding of $\mathbb{C}$ that fixes $v_1, \dots, v_m$ is also an element of $\overline{\text{CSP}(\mathbb{B})}$. Since Π is a s_n lam Datalog program, we have that $m \leq n$. Therefore, $\overline{\text{CSP}(\mathbb{B})}$ is closed under n -fixed unfoldings of caterpillars. $\square$

F. From Caterpillars to Hagemann-Mitschke Chains

In this section we prove (6) $\Rightarrow$ (7) from Theorem 4.1. First we present a well known tool to test whether the polymorphism clone of a structure satisfies a minor condition.

Definition F.1. Let $\mathbb{B}$ be a relational τ -structure and let Σ be a minor condition with a single function symbol f of arity m . Let $\sim$ be the smallest equivalence relation on B^m such that $a \sim b$ if Σ contains $f(x_1, \dots, x_m) \approx f(y_1, \dots, y_m)$ and there is a map $s: \{x_1, \dots, x_m, y_1, \dots, y_m\} \rightarrow B$ with $a = (s(x_1), \dots, s(x_m))$ and $b = (s(y_1), \dots, s(y_m))$. Note that if $a \sim b$, then Σ implies $f(a) \approx f(b)$, here the entries of a and b are viewed as variables. Then the *indicator structure of Σ with respect to $\mathbb{B}$* is the τ -structure $\mathbb{B}^m / \sim$, where the tuple $([a_1]_\sim, \dots, [a_n]_\sim)$ is relation $R^{\mathbb{B}^m / \sim}$ if there are $a'_1 \in [a_1]_\sim, \dots, a'_n \in [a_n]_\sim$ such that $(a'_1, \dots, a'_n) \in R^{\mathbb{B}^m}$.

The following is straightforward from the definitions.

Lemma F.2. Let $\mathbb{B}$ be a structure and Σ be a minor condition with a single function symbol. Then $\mathbb{B}$ has a polymorphism satisfying Σ if and only if the indicator structure of Σ with respect to $\mathbb{B}$ has a homomorphism to $\mathbb{B}$.

It is well known that for clones over a finite domain the restriction to minor conditions with a single function symbol is inessential, for a proof see Lemma 2.21 in [2].

Lemma F.3. Let Σ be a minor condition. Then there exists a minor condition Σ' with a single function symbol such that a clone over a finite domain satisfies Σ if and only if it satisfies Σ' .

The main result of this section is the following.

Lemma F.4. Let $n \geq 0$ and let $\mathbb{B}$ be a relational structure with n -fixed unfolded caterpillar duality. If $\text{Pol}(\mathbb{B})$ does not satisfy a minor condition Σ , then Σ implies (for finite clones) $\text{HM}(n)$.

The proof is a generalization of the proof of Lemma 3.10 in [2]. The rough idea of the proof is the following:

1. Consider the indicator structure $\mathbb{I}$ of Σ with respect to $\mathbb{B}$.
2. Since $\mathbb{B} \not\models \Sigma$, there must be a forbidden caterpillar $\mathbb{C}$ that has a homomorphism h into $\mathbb{I}$.
3. Find the Hagemann-Mitschke chain in the image of h .

Proof. Let τ be the signature of $\mathbb{B}$. Suppose that $\mathbb{B}$ has N -fixed unfolded caterpillar duality and that $\text{Pol}(\mathbb{B}) \not\models \Sigma$. By Lemma F.3, we may assume that Σ only involves a single function symbol f of arity K . Let $\mathbb{I}$ be the indicator structure of Σ with respect to $\mathbb{B}$. Then $\mathbb{I} \not\models \mathbb{B}$ (Lemma F.2). By definition $\mathbb{B}$ has caterpillar duality, hence there must be a caterpillar $\mathbb{C}$ with a homomorphism to $\mathbb{I}$ that does not have a homomorphism to $\mathbb{B}$. Let $(s_0, \dots, s_m)$ be the spine of $\mathbb{C}$. By N -fixed unfolded caterpillar duality there is an $n \leq N$ and vertebrae $v_1, \dots, v_n$ of $\mathbb{C}$ such that no unfolding of $\mathbb{C}$ that fixes $v_1, \dots, v_n$ has a homomorphism to $\mathbb{B}$. In the following we will show that

- Σ implies $\text{HM}(n)$, or that,
- there exists an unfolding of $\mathbb{C}$ fixing $v_1, \dots, v_n$ that has a homomorphism to $\mathbb{B}$, which contradicts that $v_1, \dots, v_n$ witness the N -fixed unfolded caterpillar duality of $\mathbb{B}$ for the caterpillar $\mathbb{C} \not\models \mathbb{B}$.

It is well known and easy to see that $\text{HM}(n)$ implies (for minions) $\text{HM}(N)$. Hence, showing that there are only these two cases concludes the proof.

Let $\phi \wedge \psi(s_0, \dots, s_m)$ be the conjunctive query representation of $\mathbb{C}$ and let h be the maximal number of hairs of any s_i , i.e. the maximal number of conjuncts in any ψ_i . First note that it is impossible that simultaneously $m = 0$ and $h = 1$, since otherwise $R^{\mathbb{I}} \neq \emptyset$ and $R^{\mathbb{B}} = \emptyset$ for the single relationsymbol R occurring in ϕ_0 . This can not be by construction of $\mathbb{I}$.

Let α be a homomorphism from $\mathbb{C}$ to $\mathbb{I}$. Note that the element $\alpha(s_i)$ of the indicator structure $\mathbb{I}$ is a set of K -tuples of elements in B . To be able to better work with α we will choose representatives of these sets. We pick them in such a way that they witness that α is a homomorphism. Formally, fix tuples $t_0^0, \dots, t_0^{h+1}, t_1^0, \dots, t_1^{h+1}, \dots, t_m^0, \dots, t_m^{h+1}$ in $\mathbb{B}^K$ such that

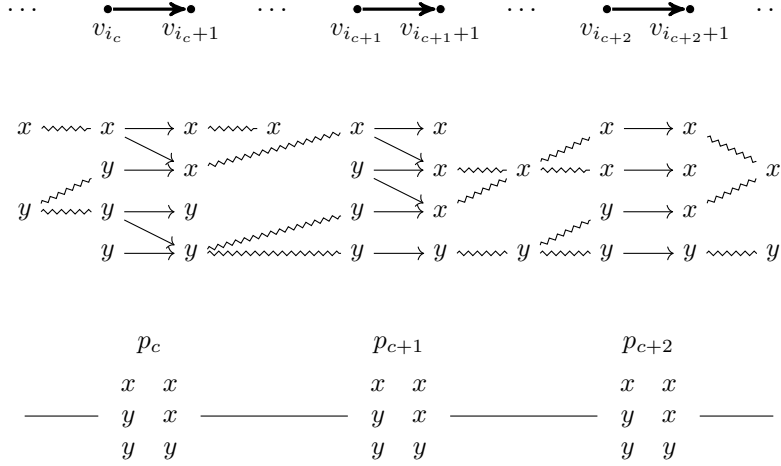


Figure 7: At the top there is a picture of two fixed vertebrae of a caterpillar $\mathbb{C}$. Below there is a visualization of the map π and the relations $\rightsquigarrow_c, \rightarrow_{i_c}, \rightsquigarrow_{c+1}, \rightarrow_{i_{c+1}}, \rightsquigarrow_{c+2}, \rightarrow_{i_{c+2}},$ and $\rightsquigarrow_{c+3}$. Observe that there is a 3-braid embedded in this picture. At the bottom we see that this construction yields the identities $p_c(x, x, y) \approx p_{c+1}(x, y, y)$ and $p_{c+1}(x, x, y) \approx p_{c+2}(x, y, y)$.

- (P1) for every $c \in [0, m]$ we have $t_c^0, \dots, t_c^{h+1} \in \alpha(s_c) \subseteq B^K$,
- (P2) for every $c \in [0, m]$ and every conjunct $\psi(s_c)$ of $\psi_c(s_c)$ there is an $i \in [h]$ such that $\mathbb{B}^K \models \psi(t_c^i)$ and
- (P3) for every $c \in [0, m-1]$ we have $\mathbb{B}^K \models \phi_c(t_c^{h+1}, t_{c+1}^0)$.

The existence of such tuples follows directly from α being a homomorphism and $\mathbb{I}$ being an indicator structure. Note that $\mathbb{B}^K \models \psi(t_c^i)$ is equivalent to $\mathbb{B} \models \psi((t_c^i)_k)$ for all $k \in [K]$ and that, similarly, $\mathbb{B}^K \models \phi_c(t_c^{h+1}, t_{c+1}^0)$ is equivalent to $\mathbb{B} \models \phi_c((t_c^{h+1})_k, (t_{c+1}^0)_k)$ for all $k \in [K]$.

Define binary relations $\sim$ and $\rightarrow_a$ on $[0, m] \times [0, h+1] \times [K]$ for $c \in [0, m-1]$:

- $(c, i, k) \sim (c, i+1, \ell)$ if and only if $(t_c^i)_k = (t_c^{i+1})_\ell$, note that the tuples t_c^i and t_c^{i+1} represent the same element in $\mathbb{I}$
- $(c, h+1, k) \rightarrow_c (c+1, 0, \ell)$ if and only if $\mathbb{B} \models \phi_c((t_c^{h+1})_k, (t_{c+1}^0)_\ell)$, note that because of (P3) this conditions holds whenever $k = \ell$, though it may also hold for some $k \neq \ell$.

We write $\leftarrow_c$ for the converse of $\rightarrow_c$. For $c \in [n]$ the vertebra v_c is of the form (s_{i_c}, s_{i_c+1}) . Define $i_0 = -1$ and $i_{n+1} = m$. For $c \in [0, n]$ we define the binary relation $\rightsquigarrow_{c+1}$ on $[0, m] \times [0, h+1] \times [K]$ as the reflexive transitive symmetric closure of $\sim \cup \rightarrow_{i_c+1} \cup \dots \cup \rightarrow_{i_{c+1}-1}$. Now we consider two cases. The first case is that there are no $k, \ell \in [K]$ such that

$$(0, 0, k) \rightsquigarrow_1 \circ \rightarrow_{i_1} \circ \rightsquigarrow_2 \circ \rightarrow_{i_2} \circ \dots \circ \rightsquigarrow_n \circ \rightarrow_{i_n} \circ \rightsquigarrow_{n+1}(m, h+1, \ell).$$

Define the map π from $[0, m] \times [0, h+1] \times [K]$ to $\{x, y\}$ with $\pi(c, i, \ell) = x$ if there exists $k \in [K]$ and $d \in [0, n]$ such that

$$(0, 0, k) \rightsquigarrow_1 \circ \rightarrow_{i_1} \circ \rightsquigarrow_2 \circ \dots \circ \rightarrow_{i_d} \circ \rightsquigarrow_{d+1}(c, i, \ell)$$

and $\pi(c, i, \ell) = y$ otherwise.

See Figure 7 for a visualization. Note the similarities with Figure 1. Observe the following.

1. By definition $\pi((0, 0, k)) = x$ for all $k \in [K]$.
2. By assumption $\pi((m, h+1, \ell)) = y$ for all $\ell \in [K]$.

3. Let $c \in [0, m]$ and let $i \in [0, h]$. By (P1) both t_c^i and t_c^{i+1} represent the element s_c in the indicator structure $\mathbb{I}$. Hence, by definition of the indicator structure, when viewing the entries of t_c^i and t_c^{i+1} as variables, Σ implies $f(t_c^i) \approx f(t_c^{i+1})$. By definition of π and $\sim$ we have that

$$(t_c^i)_{k_1} = (t_c^{i+1})_{k_2} \text{ implies } \pi(c, i, k_1) = \pi(c, i+1, k_2) \text{ for all } k_1, k_2 \in [K].$$

Therefore the minor condition $f(t_{c,i}) \approx f(t_{c,i+1})$ implies the minor condition

$$f(\pi(c, i, 1), \dots, \pi(c, i, K)) \approx f(\pi(c, i+1, 1), \dots, \pi(c, i+1, K)).$$

4. Let $c \in [0, m-1]$. If $c \notin \{i_1, \dots, i_n\}$, then, by definition of $\rightarrow_c$ and (P3), we have that $\pi(c, h+1, k) = \pi(c+1, 0, k)$ for all $k \in [K]$. Hence,

$$f(\pi(c, h+1, 1), \dots, \pi(c, h+1, K)) = f(\pi(c+1, 0, 1), \dots, \pi(c+1, 0, K)).$$

If $c \in \{i_1, \dots, i_n\}$, then, by definition of $\rightarrow_c$ and (P3), we have that $\pi(c, h+1, k) = x$ implies $\pi(c+1, 0, k) = x$ for all $k \in [K]$.

Define the height-1 condition Σ' as the set consisting of all identities from Σ and the identities

- $p_0(x) \approx f(x, \dots, x)$,
- for $c \in [n]$

$$p_c(x, y, z) \approx f(\tau(\sigma(c, 1)), \dots, \tau(\sigma(c, K)))$$

where $\sigma(c, j) = (\pi(i_c, h+1, j), \pi(i_c+1, 0, j))$ and $\tau(x, x) = x$, $\tau(y, x) = y$, and $\tau(y, y) = z$, and

- $p_{n+1}(y) \approx f(y, \dots, y)$.

Since each p_c only occurs in one identity of Σ' and in this identity all variables also occur as parameters of p_c we have that Σ' and Σ are equivalent. Note that $p_c(x, y, z) = p_c(\tau(x, x), \tau(y, x), \tau(y, y))$ and we could have also used (x, x) , (y, x) , and (y, y) as variable names, τ is just a renaming for better readability.

Define $\pi_1(x_1, x_2) = x_1$ and $\pi_2(x_1, x_2) = x_2$. Then, Σ' (and hence Σ) implies

$$\begin{aligned} p_0(x) &\approx f(x, \dots, x) \\ &= f(\pi(0, 0, 1), \dots, \pi(0, 0, K)) \\ &\approx f(\pi(0, 1, 1), \dots, \pi(0, 1, K)) \\ &\vdots \\ &\approx f(\pi(i_1, h+1, 1), \dots, \pi(i_1, h+1, K)) \\ &= f(\pi_1(\sigma(1, 1)), \dots, \pi_1(\sigma(1, K))) \\ &\approx p_1(\pi_1(x, x), \pi_1(y, x), \pi_1(y, y)) \\ &= p_1(x, y, y) \end{aligned}$$

for $c \in [n-1]$

$$\begin{aligned} p_c(x, x, y) &= p_c(\pi_2(x, x), \pi_2(y, x), \pi_2(y, y)) \\ &\approx f(\pi_2(\sigma(c, 1)), \dots, \pi_2(\sigma(c, K))) \\ &= f(\pi(i_c+1, 0, 1), \dots, \pi(i_c+1, 0, K)) \\ &\approx f(\pi(i_c+1, 1, 1), \dots, \pi(i_c+1, 1, K)) \\ &\vdots \end{aligned}$$

$$\begin{aligned}
&\approx f(\pi(i_{c+1}, h+1, 1), \dots, \pi(i_{c+1}, h+1, K)) \\
&= f(\pi_1(\sigma(c+1, 1)), \dots, \pi_1(\sigma(c+1, K))) \\
&\approx p_{c+1}(\pi_1(x, x), \pi_1(y, x), \pi_1(y, y)) \\
&= p_{c+1}(x, y, y)
\end{aligned}$$

and

$$\begin{aligned}
p_n(x, x, y) &= p_n(\pi_2(x, x), \pi_2(y, x), \pi_2(y, y)) \\
&\approx f(\pi_2(\sigma(n, 1)), \dots, \pi_2(\sigma(n, K))) \\
&= f(\pi(i_n + 1, 0, 1), \dots, \pi(i_n + 1, 0, K)) \\
&\approx f(\pi(i_n + 1, 1, 1), \dots, \pi(i_n + 1, 1, K)) \\
&\vdots \\
&\approx f(\pi(m, h+1, 1), \dots, \pi(m, h+1, K)) \\
&= f(y, \dots, y) \\
&\approx p_{n+1}(y).
\end{aligned}$$

Hence, Σ implies $\text{HM}(n)$, as desired. The case $n = 0$ is slightly different, for a detailed discussion of this case see proof of Lemma 3.10 in [2].

Now we consider the case that there are $k, \ell \in [K]$ such that

$$(0, 0, k) \leftarrow_{i_1} \rightarrow_{i_1} \leftarrow_{i_2} \rightarrow_{i_2} \dots \leftarrow_{i_n} \rightarrow_{i_n} \leftarrow_{n+1} (m, h+1, \ell).$$

Then there exists a sequence $a_0, \dots, a_M$ of distinct elements in $[0, m] \times [0, h+1] \times [K]$ such that $a_0 = (0, 0, k)$, $a_M = (m, h+1, \ell)$, and for every $i \in [M]$ we have $a_{i-1} \sim a_i$, $a_{i-1} \rightarrow_j a_i$ for some $j \in [m]$, or $a_{i-1} \leftarrow_j a_i$ for some $j \in [m] \setminus \{i_1, \dots, i_n\}$. We will use this sequence to construct an unfolding of $\mathbb{C}$ that fixes $v_1, \dots, v_n$ and has a homomorphism to $\mathbb{B}$, contradicting that $\mathbb{B}$ has n -fixed unfolded caterpillar duality while $\mathbb{C} \not\rightarrow \mathbb{B}$. See Figure 4 in [2] for a visualization of the construction of the unfolding of $\mathbb{C}$ that follows. First partition the sequence $0, \dots, M$ into the sequence $A_0, \dots, A_{M'}$ such that

- $0 \in A_0$ and $M \in A_{M'}$,
- each A_i consists of an interval, i.e., for all $i \in [0, M']$ there are $0 \leq j_1 \leq j_2 \leq M$ such that $A_i = \{j_1, j_1 + 1, \dots, j_2\}$,
- for all $i \in [0, M']$ and $j \in A_i$ with $j+1 \in A_i$ we have $a_j \sim a_{j+1}$,
- for all $0 \leq i_1 < i_2 \leq M'$ and all $j_1 \in A_{i_1}, j_2 \in A_{i_2}$ we have $j_1 < j_2$, and
- for all $i \in [M']$ we have $\max(A_{i-1}) + 1 = \min(A_i)$ and there is a $c \in [m]$ with $a_{\max(A_{i-1})} \rightarrow_c a_{\min(A_i)}$ or $a_{\max(A_{i-1})} \leftarrow_c a_{\min(A_i)}$.

We write $A_i \rightarrow_c A_{i+1}$ if $a_{\max(A_{i-1})} \rightarrow_c a_{\min(A_i)}$ and $A_i \leftarrow_c A_{i+1}$ if $a_{\max(A_{i-1})} \leftarrow_c a_{\min(A_i)}$. Note that

$$|\{c \mid (c, i', k) \in \{a_j \mid j \in A_i\}\}| = 1. \quad (*)$$

Define the primitive positive formula $\Phi(0, \dots, M')$ by adding conjuncts in the following way: Add the conjuncts of $\psi_0(0)$ and of $\psi_m(M')$. For all i and all c

- if $A_i \rightarrow_c A_{i+1}$, then add the conjunct $\phi_c(i, i+1)$,
- if $A_i \leftarrow_c A_{i+1}$, then add the conjunct $\phi_c(i+1, i)$,
- if $A_{i-1} \rightarrow_c A_i \rightarrow_{c+1} A_{i+1}$, then add all conjuncts of $\psi_c(i)$, and

- if $A_{i-1} \leftarrow_{c+1} A_i \leftarrow_c A_{i+1}$, then add all conjuncts of $\psi_c(i)$.

Note that in the last two cases, by definition of $\sim$,

$$[0, h+1] = \{i' \mid (c, i', k) \in \{a_j \mid j \in A_i\}\}. \quad (**)$$

Denote the canonical database of Φ by $\mathbb{C}'$. Note that $\mathbb{C}'$ is a caterpillar with spine $(0, \dots, M')$. Observe that, by $(*)$, for all $i \in [0, M']$ and $j_1, j_2 \in A_i$ with $a_{j_1} = (c, i', k)$ and $a_{j_2} = (d, j', \ell)$ we have that $c = d$. Hence $t_c^{i'}$ and $t_d^{j'}$ both represent the element s_c in $\mathbb{C}$. By definition of A_i and $\sim$ we have that $(t_c^{i'})_k = (t_d^{j'})_\ell$. Hence, the map $\iota: [0, M'] \rightarrow B$ that maps i to $(t_c^{i'})_k$ for some (any) (c, i', k) for which there is a $j \in A_i$ with $(c, i', k) = a_j$ is well defined. The map ι is a satisfying assignment of Φ in $\mathbb{B}$:

- If $A_i \rightarrow_c A_{i+1}$, then $(c, i', k) = a_{\max(A_i)} \rightarrow_j a_{\min(A_{i+1})} = (d, j', \ell)$ and $\iota(i) = (t_c^{i'})_k$ and $\iota(i+1) = (t_d^{j'})_\ell$. By definition of $\rightarrow_j$ we have $\mathbb{B} \models \phi_c((t_c^{i'})_k, (t_d^{j'})_\ell)$.
- The case $A_i \leftarrow_c A_{i+1}$ is analogous.
- If $A_{i-1} \rightarrow_c A_i \rightarrow_{c+1} A_{i+1}$ and $\psi(i)$ is a conjunct of $\psi_c(i)$, then, by (P2), there is a $i' \in [h]$ such that $\mathbb{B} \models \phi((t_c^{i'})_k)$ for all $k \in [K]$. By $(**)$ there is some $\ell \in [K]$ and some $j \in A_i$ such that $(c, i', \ell) = a_j$. Hence $\iota(i) = (t_c^{i'})_\ell$ and $\mathbb{B} \models \psi(\iota(i))$.
- The case $A_{i-1} \leftarrow_{c+1} A_i \leftarrow_c A_{i+1}$ is analogous.

Therefore, ι can be extended to a homomorphism from $\mathbb{C}'$ to $\mathbb{B}$. Similar to the proof of Lemma D.3 we can use Lemma D.1 to show that $\mathbb{C}'$ is an unfolding of $\mathbb{C}$ that fixes $v_1, \dots, v_n$. This concludes the proof. $\square$

It is possible extend the definition of indicator structures to minor conditions Σ with multiple function symbols $f_1, \dots, f_n$ of arities $k_1, \dots, k_n$, respectively. Let $\mathbb{B}$ be a structure. $\mathbb{B}^{f_1, \dots, f_n}$ is the structure with domain $\{f_i(b_1, \dots, b_{k_i}) \mid i \in [n], b_1, \dots, b_{k_i} \in B\}$ such that for each $i \in [n]$ the map $h_i: \mathbb{B}^{k_i} \rightarrow \mathbb{B}^{f_1, \dots, f_n}, t \mapsto f_i(t)$ is an embedding and $\mathbb{B}^{f_1, \dots, f_n}$ is the disjoint union of the images of $h_1, \dots, h_n$. Let $\sim$ be the smallest equivalence relation on $B^{f_1, \dots, f_n}$ such that $f_i(a) \sim f_j(b)$ if Σ contains $f_i(x_1, \dots, x_{k_i}) \approx f_j(y_1, \dots, y_{k_j})$ and there is a map $s: \{x_1, \dots, x_{k_i}, y_1, \dots, y_{k_j}\} \rightarrow B$ with $a = (s(x_1), \dots, s(x_{k_i}))$ and $b = (s(y_1), \dots, s(y_{k_j}))$. Then the *indicator structure of Σ with respect to $\mathbb{B}$* is $\mathbb{B}^{f_1, \dots, f_n} / \sim$. Using these indicator structures the proof of Lemma F.4 with (for finite clones) replaced by (for minions) works essentially in the same way, it just becomes even more technical.

G. Transitive tournaments have elevator chains

To prove (8) $\Rightarrow$ (1) we need the following statement.

Lemma G.1. *Let $n \in \mathbb{N}$. Then $\text{Pol}(\mathbb{T}_{n+1})$ contains operations that form an elevator chain of length n .*

This proof was discovered in discussion with Marlena Wünnig.

Proof. For $n = 0$ the structure $\mathbb{T}_{0+1}$ consists of a single isolated point and has a constant polymorphism satisfying $\text{Ele}(0)$. Now we consider the case $n \geq 1$. Let $\mathbb{I}$ be the indicator structure of $\text{Ele}(n)$ with respect to $\mathbb{T}_{n+1}$. Note that for every element s of $\mathbb{I}$ there are $a_1, \dots, a_n \in \mathbb{T}_{n+1}$ such that either

- s is a singleton and there is an $i \in [n]$ and $a'_i, a''_i \in \mathbb{T}_n$ with

$$s = \{e_i(a_1, \dots, a_{i-1}, a_i, a'_i, a''_i, a_{i+1}, \dots, a_n)\},$$

$$a_i \neq a'_i, \text{ and } a'_i \neq a''_i \text{ or}$$

- s contains $e_0(a_1, \dots, a_n)$ and

$$e_i(a_1, \dots, a_{i-1}, b, b, a_i, a_{i+1}, \dots, a_n) \text{ and } e_i(a_1, \dots, a_{i-1}, a_i, b, b, a_{i+1}, \dots, a_n)$$

for all $b \in \mathbb{T}_{n+1}$ and all $i \in [n]$.

We define a map f from $\mathbb{I}$ to $[0, n]^n$. Elements of the first form are mapped to

$$(a_1, \dots, a_{i-1}, \min(a_i, a'_i, a''_i), a_{i+1}, \dots, a_n)$$

and elements of the second form are mapped to $(a_1, \dots, a_n)$. Clearly, f is well defined.

We define the digraph ${}^1\tilde{\mathbb{T}}_{n+1}^n$ with domain $[0, n]^n$ that has an edge from $(a_1, \dots, a_n)$ to $(b_1, \dots, b_n)$ if and only if for all but at most one $j \in [n]$ we have $a_j < b_j$ and if $a_j \geq b_j$, then $a_j < n$ and $0 < b_j$.

Now we will show that f is a homomorphism from $\mathbb{I}$ to ${}^1\tilde{\mathbb{T}}_{n+1}^n$. Let (s, t) be an edge in $\mathbb{I}$ and

$$e_i(a_1, \dots, a_{i-1}, a_i, a'_i, a''_i, a_{i+1}, \dots, a_n) \in s \text{ and } e_i(b_1, \dots, b_{i-1}, b_i, b'_i, b''_i, b_{i+1}, \dots, b_n) \in t$$

be witnesses for this edge. Note that it is always possible to pick witnesses with $i \notin \{0, n+1\}$. Then $a_1 < b_1, \dots, a_n < b_n, a'_i < b'_i$, and $a''_i < b''_i$. Hence $f(s)_j = a_j < b_j = f(t)_j$ for all $j \in [n] \setminus \{i\}$. Furthermore, $a_i < b_i, a'_i < b'_i$, and $a''_i < b''_i$ imply that $a_i, a'_i, a''_i < n$ and that $0 < b_i, b'_i, b''_i$. Since $f(s)_i \in \{a_i, a'_i, a''_i\}$ and $f(t)_i \in \{b_i, b'_i, b''_i\}$ we get $f(s)_i < n$ and $0 < f(t)_i$. Hence, f is a homomorphism. Note that it can be the case that $f(s)_j \geq f(t)_j$, if for example $a_i = 0 = a'_i, b'_i = n = b''_i$, and $n > a''_i \geq b_i > 0$.

Finally, we show ${}^1\tilde{\mathbb{T}}_{n+1}^n \rightarrow \mathbb{T}_{n+1}$. Since $(\{\mathbb{P}_{n+1}\}, \mathbb{T}_{n+1})$ is a duality pair it suffices to show that $\mathbb{P}_{n+1} \not\rightarrow {}^1\tilde{\mathbb{T}}_{n+1}^n$. Let $(a_1^0, \dots, a_n^0), \dots, (a_1^m, \dots, a_n^m)$ be a directed path in ${}^1\tilde{\mathbb{T}}_{n+1}^n$. We want to show that $m \leq n$. By definition of the edge relation $a_1^1, \dots, a_n^1 \geq 1$. Since in an edge all but at most one entry strictly increases we have that there are at least $n - j + 1$ $i \in [n]$ such that $a_i^1 < \dots < a_i^j$ for all $j \in [1, m]$. Hence for $j = n$ there is at least one $i \in [n]$ such that $a_i^n \geq n$. Since $a_i^n \in [0, n]$ we conclude $a_i^n = n$ and by definition of the edge relation $(a_1^n, \dots, a_n^n)$ does not have an outgoing edge. Therefore $m \leq n$ and $\mathbb{P}_{n+1} \not\rightarrow {}^1\tilde{\mathbb{T}}_{n+1}^n$. In conclusion $\mathbb{I} \rightarrow {}^1\tilde{\mathbb{T}}_{n+1}^n \rightarrow \mathbb{T}_{n+1}$. Hence $\mathbb{T}_{n+1} \models \text{Ele}(n)$. $\square$

Note that ${}^1\tilde{\mathbb{T}}_{n+1}^n$ is similar but not the same as the one-tolerant n -th power ${}^1\mathbb{T}_{n+1}^n$ of $\mathbb{T}_{n+1}$ introduced by Larose, Loten, and Tardif in Section 4.2 in [13]. In fact, the smallest k such that ${}^1\mathbb{T}_{n+1}^k$ has a homomorphism to $\mathbb{T}_{n+1}$ is $n+1$. We have seen in Lemma 3.4 that $\text{Ele}(n)$ implies $\text{HM}(n)$. In combination with Lemma G.1 we obtain a new proof of the following well known fact.

Corollary G.2. *Let $n \in \mathbb{N}$. Then $\text{Pol}(\mathbb{T}_{n+1}) \models \text{HM}(n)$.*

H. Proof of the main result

Combining the previous section we can prove the main result.

Proof of Theorem 4.1. Let $n \in \mathbb{N}$ and let $\mathbb{B}$ be a structure with a finite domain and a finite relational signature. The implication (1) $\Rightarrow$ (2) follows from Lemma B.1 and Theorem 2.8, the implication (2) $\Rightarrow$ (5) from Lemma B.6, the implication (5) $\Rightarrow$ (6) is trivial, the implication (6) $\Rightarrow$ (7) from Lemma F.4. For the implication (7) $\Rightarrow$ (8) suppose that Σ is a minor condition that holds in $\text{Pol}(\mathbb{T}_{n+2})$. By Theorem 2.10 we have $\mathbb{T}_{n+2} \not\models \text{HM}(n)$. Hence, Σ can not imply (for finite clones) $\text{HM}(n)$. Therefore (7) implies $\mathbb{B} \models \Sigma$, as desired. For the implication (8) $\Rightarrow$ (1) it suffices to show that $\text{Pol}(\mathbb{T}_{n+2})$ contains the desired operations. Since $(\{\mathbb{P}_{n+2}\}, \mathbb{T}_{n+2})$ is a duality pair $\mathbb{T}_{n+2}$ has caterpillar duality. Hence, by Theorem 2.8, $\text{Pol}(\mathbb{T}_{n+2})$ contains k -absorptive operations of arity ℓk , for all $\ell, k \geq 1$. By Lemma G.1 $\text{Pol}(\mathbb{T}_{n+2})$ contains operations that form an elevator chain of length $n+1$.

For the implication $(5) \Rightarrow (3)$ let Π_S be the canonical s_n lam Datalog program for $\mathbb{B}$. Since $\mathbb{B}$ has caterpillar duality Theorem 2.8 implies that the canonical lam Datalog program Π for $\mathbb{B}$ solves $\text{CSP}(\mathbb{B})$. Lemma D.3 implies that Π and Π_S derive the goal predicate on the same instances of $\text{CSP}(\mathbb{B})$. Therefore, Π_S solves $\text{CSP}(\mathbb{B})$. The implication $(3) \Rightarrow (4)$ is trivial, the implication $(4) \Rightarrow (5)$ is Lemma E.2.

The equivalence of $(8) - (10)$ follows from Theorem 2.1. That $(1) \Leftrightarrow (11)$ follow from the equivalence of items (3) and (4) in Theorem 2.8 and the fact that the existence of an elevator chain of length $n + 1$ is preserved under homomorphic equivalence. $\square$