DIPLOMARBEIT

On the Descriptive Complexity of Phylogeny Constraint Satisfaction Problems

zur Erlangung des akademischen Grades
Diplom-Ingenieur

im Rahmen des Studiums
Technische Mathematik

ausgeführt am
Institut für Diskrete Mathematik und Geometrie
der Technischen Universität Wien

unter der Anleitung von
Assoc. Prof. Dr. Michael Pinsker

und
Dr.-Ing. Jakub Rydval

durch
Paul Winkler

Wien, im Mai 2024

Acknowledgements

I would like to thank Jakub Rydval, who guided me through the djungle of papers and concepts, provided me with relevant questions and always had creative ideas when I got stuck. I am also very grateful to Professor Pinsker for giving me the opportunity to work on this interesting topic and for his various helpful comments and explanations.

Ich danke meinen Eltern für ein Vierteljahrhundert Unterstützung, insbesondere für die Finanzierung meines Studiums.

Kurzfassung

Auf den Blättern eines Wurzelbaums ist die ternäre Relation C definiert durch $C(x, y, z)$ genau dann wenn der jüngste gemeinsame Vorfahre von x, y und z strikt näher bei der Wurzel liegt als der jüngste gemeinsame Vorfahre von y und z. Ein phylogenetisches Problem ist ein Berechnungsproblem, dessen Instanzen Instanziierungen einer fixen Menge Boolescher Kombinationen von Formeln der Form $C(x, y, z)$ oder $x = y$ sind; die Frage ist, ob es einen Wurzelbaum und eine Abbildung von den Variablen auf die Blätter dieses Baumes gibt, sodass alle Formeln erfüllt sind.

Jedes phylogenetische Problem entspricht einem Bedingungserfüllungsproblem (engl. Constraint Satisfaction Problem, kurz CSP) einer speziellen unendlichen Struktur, die ω-kategorisch und in den wichtigsten Fällen auch homogen ist. Es ist bekannt, dass ein phylogenetisches CSP in P liegt, wenn diese Struktur eine bestimmte algebraische Eigenschaft aufweist; andernfalls ist es NP-vollständig. In dieser Arbeit wollen wir die deskriptive Komplexität solcher Probleme untersuchen, insbesondere ihre Ausdrückbarkeit in Fixpunktlogiken.

Einerseits präsentieren wir ein phylogenetisches Problem, das zwar in P liegt, sich aber nicht in Fixpunktlogik ausdrücken lässt, nicht einmal in der Erweiterung durch Zählquantoren. Andererseits führen wir Boolesche Hornformeln ein; dabei handelt es sich um eine syntaktische Einschränkung von affinen Hornformeln. Wir zeigen, dass alle phylogenetischen CSPs, die eine Vorlage haben, deren Relationen sich mittels Booleschen Hornformeln definieren lassen, in Fixpunktlogik ausdrückbar sind. Außerdem gibt es eine spezielle Struktur, die alle solchen Strukturen pp-definiert. Unter einer zusätzlichen Bedingung sind diese Strukturen genau die, die von einem bestimmten Polymorphismus bewahrt werden.

Abstract

On the leafs of a rooted tree, the ternary relation C is given by $C(x, y, z)$ if and only if the youngest common ancestor of x, y and z is strictly closer to the root than the youngest common ancestor of y and z. A phylogeny problem is a computational decision problem whose instances are instantiations of a fixed set of Boolean combinations of formulas of the form $C(x, y, z)$ and $x = y$; the question is whether there is a rooted tree and a mapping from the variables to the leaves of the tree such that all formulas are satisfied.

Each phylogeny problem corresponds with a constraint satisfaction problem (CSP) of a specific infinite structure, which is ω-categorical and in the most important cases also homogeneous. It has been shown that a phylogeny CSP is in P if this structure fulfilles a certain algebraic condition, and NP-complete otherwhise. In this thesis, we want to study the descriptive complexity of such problems, especially their expressibility in fixed point logics.

On the one hand, we present a phylogeny problem which is tractable, but inexpressible in fixed point logic, even with counting. On the other hand, we introduce Boolean Horn formulas; they are a further syntactic restriction of affine Horn formulas. It turns out that all phylogeny CSPs with a template whose relations can be defined by Boolean Horn formulas are expressible in fixed point logic. Moreover, there is a specific structure which pp-defines all such structures. Under an additional assumption, those structures are characterized as being preserved by a specific polymorphism.

Contents

1 Introduction

Phylogeny problems are computation decision problems motivated by evolutionary biology. Every species uniquely stems from a prior species, and all species have a common ancestor. A natural question is how to reconstruct the tree of life (or parts of it) using only observations about recent species, i. e. the leaves of the tree.

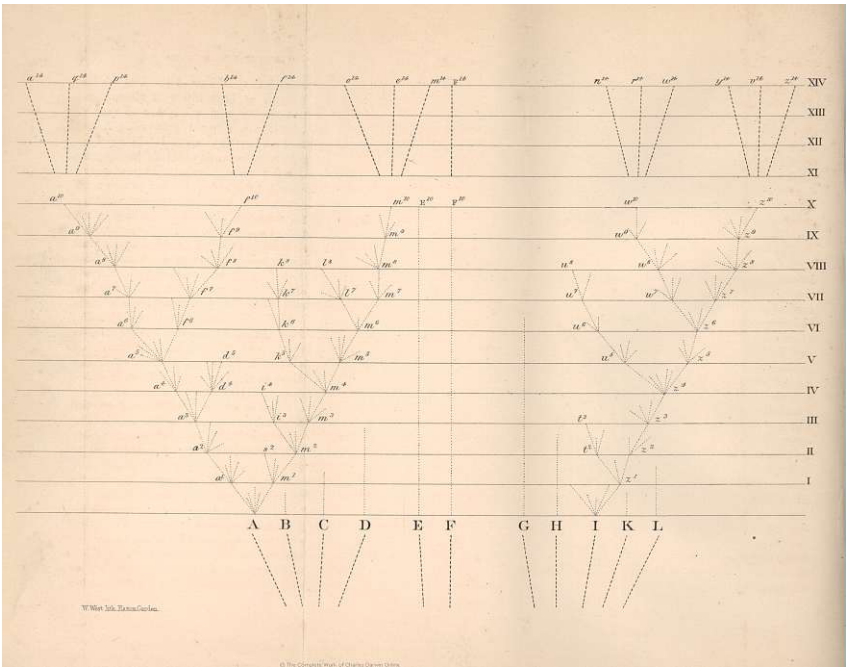


Figure 1: Illustration of an evolutionary tree in Darwin's *Origin of species* (1859).

For the leaves x, y, z of a rooted tree, we write $x|yz$ if the youngest common ancestor of y and z lies strictly below the youngest common ancestor of x, y and z. The set of all leaves of a rooted tree together with this ternary relation is called the *leaf structure* of the tree. The *basic phylogeny problem* asks whether, for a given set of constraints of the form $x_i|x_jx_k$ over variables V, there exists a mapping s from V to the leaves of some rooted tree such that $s(x_i)|s(x_j)s(x_k)$ holds for every constraint $x_i|x_jx_k$. It is known that this problem can be solved in quadratic time using a simple divide-and-conquer algorithm [1]. Matters get more difficult, however, when we allow Boolean combinations as constraints. For example, consider the ternary relation N defined by the formula $(x|yz \vee z|xy)$. The problem whether there exists a rooted tree whose leaves satisfy a given set of constraints of the form $N(x_i, x_j, x_k)$ is NP-hard [1]. The class of all phylogeny problems is obtained by considering all sets of relations which are specified by Boolean combinations of formulas of the form $(x|yz)$ or $(x = y)$.

1.1 The Algebraic Complexity Dichotomy

The key to a successful analysis of phylogeny problems was the observation that they can be viewed as *Constraint Satisfaction Problems* (CSPs) of highly symmetrical infinite structures [1]. The CSP of a structure Γ with a finite relational signature, denoted CSP(Γ), is the following computational decision problem:

CSP(Γ)
INSTANCE: A finite conjunction $\psi(x_1, \ldots, x_k)$ of atomic formulas in the signature of Γ.
QUESTION: Are there elements $a_1, \ldots, a_k \in \Gamma$ such that $\Gamma \models \psi(a_1, \ldots, a_k)$?

The relational structure Γ is called a *template* of the CSP, and the conjuncts in ψ are called *constraints*. The equivalent formulation below is more suitable in some contexts:

CSP(Γ)
INSTANCE: A finite structure Δ with the same signature as Γ.
QUESTION: Is there a homomorphism from Δ to Γ?

Every phylogeny problem is of the form CSP(Γ) for an *ω-categorical* structure Γ; we can therefore speak of »phylogeny CSPs« instead of »phylogeny problems«. In the most important cases, Γ is also *homogeneous*. The main result of [1] states that a phylogeny CSP is solvable in polynomial time if the associated ω-categorical template fulfills a certain algebraic condition, and NP-hard otherwise. Moreover, there is a uniform source of hardness: the ability to *primitive positively construct* a finite structure which does not satisfy this algebraic condition and whose CSP is NP-hard. Results of this form are referred to as algebraic complexity dichotomies. The advantage over a plain complexity classification is that the result remains nontrivial even if P = NP.

1.2 Descriptive Complexity

What is also unaffected by the P-NP problem is the expressibility of CSPs in *fixed point logics*. Those are extensions of first-order logic by expressions whose semantics is given in terms of fixed points of operators specified by first-order formulas. For instance, if $G = (V; E)$ is a finite undirected graph, then there will be an expression for the relation containing all pairs (x, y) contained in U_i at some stage of the following recursion:

$$U_0 := \varnothing, \qquad U_{i+1} := U_i \cup \{(x, y) \in V^2 \mid E(x, y) \vee \exists z(E(x, z) \wedge U_i(z, y))\}.$$

It is easy to see that the *fixed-point* $U_\infty := \bigcup_i U_i$ is the transitive closure of E. Using the limits of such sentences, it becomes possible to define properties of structures that are not definable in first-order logic, e. g. that G is connected: $\forall x \forall y(U_\infty(x, y))$ [20].

In [3], a classification of *temporal CSPs* with respect to expressibility in some important fixed point logics was given. Temporal CSPs are CSPs with a template that is a (first-order) reduct of $(\mathbb{Q}; <)$; they are structurally similar to phylogeny CSPs and both appear as special cases of CSPs of reducts of the universal homogeneous binary tree [2]. In particular, the question arises which phylogeny CSPs are expressible in fixed point logic, and how (in)expressibility can be characterized algebraically.

We will identify a large class of phylogeny CSPs which are expressible, as well as an example of a tractable phylogeny CSP which is inexpressible in the basic fixed point logic FP.

1.3 Organization of the Thesis

In Chapter 2, we will collect various definitions and known results which we will need in later chapters. In particular, we will introduce polymorphisms, ω-categoricity and

homogeneity, the concepts of Fraïssé amalgamation and pp-constructions as well as fixed point logics. In the last subsection, we will establish a dichotomy for equality CSPs with respect to their descriptive complexity.

In Chapter 3, we will formalize leaf structures and phylogeny problems. We will show how to amalgamate two leaf structures, which yields the generic leaf structure $(\mathbb{L}; C)$ as a Fraïssé limit, and how every phylogeny problem can be interpreted as the CSP of a reduct of $(\mathbb{L}; C)$. Subsequently, we will use a known classification of such CSPs to study reducts of $(\mathbb{L}; C)$ which do not pp-define C (the »degenerative« cases) and establish a dichotomy for phylogeny CSPs in Datalog, a fragment of the basic fixed point logic FP.

In Chapter 4, we present a phylogeny problem which is tractable, but inexpressible in fixed point logic, even with counting.

In Chapter 5, we introduce Boolean Horn formulas; they are a syntactic restriction of affine Horn formulas introduced in [1]. It turns out that all phylogeny CSPs with a template whose relations can be defined by Boolean Horn formulas are expressible in fixed point logic. Moreover, there is a specific structure which pp-defines all such structures. We also characterize such structures in terms of having a specific polymorphism.

2 Preliminaries

2.1 Basic Definitions and Notation

We start with some basic vocabulary from model theory and logic. A *relational structure* with signature $\tau = \{R_1, R_2, \ldots\}$ is a tuple $\Gamma = (D_\Gamma; R_1^\Gamma, R_2^\Gamma, \ldots)$ consisting of a *domain* D_Γ and relations $R_i^\Gamma \subseteq (D_\Gamma)^{\mathrm{ar}(R_i)}$ for each relation symbol $R_i \in \tau$, where $\mathrm{ar}(R)$ denotes the arity of R. For notational convenience, we will use R_i and R_i^Γ interchangeably, when there is no ambiguity. Moreover, we will not always rigorously distinguish between a structure and its domain. Unless otherwise stated, all structures considered are assumed to be relational; we will use constant symbols though as an abbreviation for unary relations containing only one element.

Let Γ be a τ-structure. A *reduct* of Γ is a τ-structure Δ with domain D_Γ such that for all $R \in \tau$, R^Δ has a first-order definition in Γ. Let $S \subseteq D_\Gamma$, then we call the τ-structure Δ with domain S and relations $R^\Delta = R^\Gamma \cap S^{\mathrm{ar}(R)}$ the *substructure of* Γ *induced by* S; it is denoted by $\Gamma[S]$. If $f\colon \Gamma[S] \to \Gamma[f(S)]$ is an isomorphism, we call it a *partial isomorphism of* Γ.

If a is a tuple, we write a_i for the i-th component of a. Let Γ, Δ be two τ-structures, $R \in \tau$ and let $h\colon \Gamma \to \Delta$ be a function. We say that h *preserves* R, denoted by $h \curvearrowright R$, if $(h(a_1), \ldots, h(a_n)) \in R^\Delta$ for all $a \in R^\Gamma$. If h preserves all relations of Γ, we say that h is a *homomorphism* between Γ and Δ. We write $\Gamma \to \Delta$ if there is a homomorphism from Γ to Δ. If $\Gamma \to \Delta$ and $\Delta \to \Gamma$, we say that Γ and Δ are *homomorphically equivalent*. If h is an injective homomorphism that preserves all relations *strongly*, i. e. $(h(a_1), \ldots, h(a_n)) \in R^\Delta$ if and only if $a \in R^\Gamma$, h is called an *embedding*. An *endomorphism* is a homomorphism from a structure to itself. An *automorphism* is a bijective embedding of a structure to itself, or, equivalently, a bijective endomorphism such that its inverse is an endomorphism as well. The sets of endomorphisms, embeddings and automorphisms of Γ are denoted by $\mathrm{End}(\Gamma), \mathrm{Emb}(\Gamma)$ and $\mathrm{Aut}(\Gamma)$, respectively.

Let F be some set of functions from Γ to itself and let a be a tuple from Γ^n, then the *orbit of* a *(with respect to* F*)* is the set $\mathcal{O}(a) := \{(\alpha(a_1), \ldots, \alpha(a_n)) \mid \alpha \in F\}$. F *locally interpolates* a function g if for every finite $A \subseteq D_\Gamma$, there is some $f \in F$ such that $f\big|_A = g\big|_A$. We denote the set of functions which can be locally interpolated by functions from F by $\overline{F}$.

We work with classical first-order logic, using $\bot$, the Boolean connectives $\wedge, \vee, \neg$, the quantifiers $\forall, \exists$ and equality. $\varphi \to \psi$ is an abbreviation for $\neg\varphi \vee \psi$. The symbol $\equiv$ stands for syntactic equality of formulas. A formula is called *existential positive* if it does not contain universal quantifiers or negation symbols. A formula is *primitive positive* (shortly *pp*) if it is of the form

$$\exists x_1 \cdots \exists x_k \, (\psi_1 \wedge \cdots \wedge \psi_n),$$

where each ψ_i is atomic, i. e. of the form $\bot, x = y$ or $R(x_{i_1}, \ldots, x_{i_l})$.

For a structure Γ, we write $\langle\Gamma\rangle$ for the set of relations which have a pp-definition in Γ.

2.2 Polymorphisms

Let $m, n \geq 1$, let $f\colon A^n \to B$ be a function of arity n and let $a_1, \ldots, a_n \in A^m$. We can extend f to a function $(A^m)^n \to B^m$ by setting

$$f(a_1, \ldots, a_n) \coloneqq (f(a_{11}, \ldots, a_{n1}), \ldots, f(a_{1m}, \ldots, a_{nm})). \tag{1}$$

This action of f on n tuples can be visualized by viewing those tuples as the rows of a matrix and applying f column-wise; the i-th component of the m-ary result is f applied to the i-th column:

$$\begin{array}{c|c} & \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ & & \vdots & \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix} \\ f \downarrow & \\ \hline & f(a_1, \ldots, a_n) \end{array}$$

This gives rise to a central tool in universal algebra:

2.1 Definition. An n-ary *polymorphism* of a k-ary relation R is a function $f\colon D^n \to D$ such that for every n tuples $a_1, \ldots, a_n \in R$, we have $f(a_1, \ldots, a_n) \in R$, where $f(a_1, \ldots, a_n)$ is defined by (1). We also say that f *preserves* R and write $f \curvearrowright R$; if this is not the case, f *violates* R. A polymorphism of a structure Γ is a function $f\colon D^n \to D$ that preserves all relations of Γ. We denote this by $f \curvearrowright \Gamma$.

We write $\mathrm{Pol}^{(n)}(\Gamma)$ for the set of n-ary polymorphisms of Γ and $\mathrm{Pol}(\Gamma)$ for the set of all of its polymorphisms.

Alternatively, we can define the n-ary polymorphisms via the direct product: The direct product of two τ-structures Γ_1, Γ_2 is the τ-structure $\Gamma_1 \times \Gamma_2$ with domain $D_{\Gamma_1} \times D_{\Gamma_2}$, where for every $R \in \tau$, we have $R^{\Gamma_1 \times \Gamma_2}((a_1, b_1), \ldots, (a_k, b_k))$ if and only if $R^{\Gamma_1}(a_1, \ldots, a_k)$ and $R^{\Gamma_2}(b_1, \ldots, b_k)$. Now the n-ary polymorphisms of Γ are the homomorphisms from $\Gamma^n = \Gamma \times \cdots \times \Gamma$ to Γ. Note that $\mathrm{Pol}^{(1)}(\Gamma) = \mathrm{End}(\Gamma)$.

We will later need the following fact:

2.2 Lemma ([9, Lemma 10])**.** *Let Γ be a relational structure and R be a k-ary relation that is contained in a union of l orbits of k-tuples of $\mathrm{Aut}(\Gamma)$. If R is violated by some polymorphism $g \in \mathrm{Pol}^{(m)}(\Gamma)$ with $m \geq l$, then R is also violated by some polymorphism $h \in \mathrm{Pol}^{(l)}(\Gamma)$.*

2.3 Some Important Concepts from Model Theory

In the following, we will introduce a few key concepts from model theory, in particular ω-categoricity and homogeneity, and the relationships between them:

2.3 Definition. A first-order theory T is called *ω-categorical* if it has at most one countably infinite model up to isomorphism. A structure Γ is called *ω-categorical* if its first-order theory is ω-categorical.

An ω-categorical theory in a countable language either has a finite model, or exactly one countably infinite model up to isomorphism (this follows from the Löwenheim-Skolem Theorem, see e. g. [19, Corollary 3.1.5]). The following theorem plays a crucial role in the study of countably infinite ω-categorical structures, as it shows that such structures behave similiarly to finite structures:

2.4 Theorem (Ryll-Nardzewski, e. g. [19, Theorem 7.3.1])**.** *Let Γ be a countably infinite structure. Then the following statements are equivalent:*

(1) *Γ is ω-categorical.*

(2) $\mathrm{Aut}(\Gamma)$ *is* oligomorphic*, i. e. for all $n \geq 1$, there are only finitely many orbits of n-tuples with respect to* $\mathrm{Aut}(\Gamma)$.

(3) *For all $n \geq 1$, there are only finitely many inequivalent formulas over Γ whose free variables are from $x_1, \ldots, x_n$.*

2.5 Corollary. *Let Γ be a countably infinite ω-categorical structure and let Δ be some reduct of Γ. Then Δ is ω-categorical as well.*

Proof. We have $\mathrm{Aut}(\Gamma) \subseteq \mathrm{Aut}(\Delta)$, since Δ is a reduct of Γ (cf. Theorem 2.6 (1)). Hence, if $\mathrm{Aut}(\Gamma)$ has only finitely many orbits of n-tuples, so does $\mathrm{Aut}(\Delta)$. The statement now follows from Theorem 2.4. □

The following theorem shows that important syntactic restrictions of first-order logic have an algebraic counterpart. Statement (1) is well-known and has an easy inductive proof which uses Theorem 2.4; for (2) and (3), see [11, Proposition 12]; a proof of (4) can be found in [13, Theorem 4].

2.6 Theorem. *Let Γ be a finite or countably infinite ω-categorical structure and R a relation. Then the following statements hold:*

(1) *R is first-order definable in Γ if and only if* $\mathrm{Aut}(\Gamma) \curvearrowright R$.

(2) *R has an existential definition in Γ if and only if* $\mathrm{Emb}(\Gamma) \curvearrowright R$.

(3) *R has an existential positive definition in Γ if and only if* $\mathrm{End}(\Gamma) \curvearrowright R$.

(4) *R is pp-definable in Γ if and only if* $\mathrm{Pol}(\Gamma) \curvearrowright R$.

2.7 Definition. A structure Γ is called *homogeneous* if every partial isomorphism between substructures of Γ with finite domain can be extended to an automorphism of Γ.

2.8 Proposition. *Let Γ be a countably infinite, homogeneous structure with finite signature. Then Γ is ω-categorical.*

Proof. By Theorem 2.4, it suffices to show that for every $n \geq 1$, $\mathrm{Aut}(\Gamma)$ has only finitely many orbits of n-tuples. For $a \in (D_\Gamma)^n$, let $\Phi_a := \{\psi(x_1, \ldots, x_n) \mid \psi \text{ quantifier-free}, \Gamma \models \psi(a_1, \ldots, a_n)\}$. Since Γ has finite signature, there are only finitely many inequivalent quantifier-free formulas with free variables $x_1, \ldots, x_n$. Hence, there are only finitely many sets of the form Φ_a. If $\Phi_a = \Phi_b$, then $\Gamma[\{a_1, \ldots, a_n\}]$ is isomorphic to $\Gamma[\{b_1, \ldots, b_n\}]$; thus, a and b lie in the same orbit of $\mathrm{Aut}(\Gamma)$ due to the homogeneity of Γ. □

2.9 Theorem (see [19, Corollary 7.4.2])**.** *Let* Γ *be an* ω*-categorical structure. Then* Γ *is homogeneous if and only if it has* quantifier elimination*, i. e. if every first-order formula is equivalent to a quantifier-free formula over* Γ*.*

2.10 Definition. A structure Γ is called a *core* if $\mathrm{End}(\Gamma) = \mathrm{Emb}(\Gamma)$.

2.11 Definition. A first-order theory T is called *model-complete* if all embeddings between models of T preserve all first-order formulas. A structure Γ is called *model-complete* if its first-order theory is model-complete.

2.12 Lemma. *Let* Γ *be a homogeneous,* ω*-categorical structure. Then* Γ *is model-complete.*

Proof. Let Λ, Ξ be two models of the theory of Γ, $f\colon \Lambda \to \Xi$ an embedding and $\varphi(x_1, \dots, x_n)$ a formula. We have to show that $\Lambda \models \varphi(a_1, \dots, a_n) \iff \Xi \models \varphi(f(a_1), \dots, f(a_n))$ for arbitary $a_1, \dots, a_n \in D_\Lambda$. As Γ has quantifier elimination by Theorem 2.9, there is a quantifier-free formula ψ such that $\Gamma \models \varphi(x_1, \dots, x_n) \leftrightarrow \psi(x_1, \dots, x_n)$. As they have the same theory as Γ, φ and ψ are also equivalent over Λ and Ξ. Hence, it suffices to verify $\Lambda \models \psi(a_1, \dots, a_n) \iff \Xi \models \psi(f(a_1), \dots, f(a_n))$; but this holds because ψ is quantifier-free and f is an embedding. □

2.13 Theorem ([10, Theorem 16])**.** *Let* Γ *be an* ω*-categorical structure. Then* Γ *is homomorphically equivalent to a model-complete core* Δ*, which is* ω*-categorical and unique up to isomorphism.*

Hence, it is justified to speak of *the* model-complete core of a certain ω-categorical structure. It has the following algebraic characterization:

2.14 Theorem (cf. [11, Lemma 13])**.** *Let* Γ *be a countably infinite* ω*-categorical structure. Then* Γ *is a model-complete core if and only if* $\mathrm{Aut}(\Gamma)$ *locally interpolates* $\mathrm{End}(\Gamma)$*.*

2.15 Lemma. *Let* Δ *be a reduct of a model-complete core* Γ *such that* Δ *existentially positively defines* Γ*. Then* Δ *is a model-complete core, too.*

Proof. By Theorem 2.14, it suffices to show that $\mathrm{End}(\Delta) \subseteq \overline{\mathrm{Aut}(\Delta)}$. Since Γ has an existentially positive definition in Δ, we have $\mathrm{End}(\Delta) \subseteq \mathrm{End}(\Gamma)$ by Theorem 2.6. As Δ is a reduct of Γ, we obtain $\mathrm{Aut}(\Gamma) \subseteq \mathrm{Aut}(\Delta)$ again by Theorem 2.6. Hence, overall, $\mathrm{End}(\Delta) \subseteq \mathrm{End}(\Gamma) \subseteq \overline{\mathrm{Aut}(\Gamma)} \subseteq \overline{\mathrm{Aut}(\Delta)}$, as required. □

2.4 Fraïssé Limits

For a class $\mathcal{A}$ of finite structures fulfilling certain conditions, there exists a countably infinite, homogeneous structure whose finite substructures are exactly the elements of $\mathcal{A}$. As the central structure of this thesis is obtained in that way, we will summarize the main idea behind its construction.

2.16 Definition. Let $\mathcal{A}$ be a class of finite structures with signature τ. $\mathcal{A}$ is called an *amalgamation class* if it is closed under isomorphisms, induced substructures and *amalgamation,* i. e. it satisfies the following property: For all $\Gamma_1, \Gamma_2 \in \mathcal{A}$ such that $\Gamma_1 \cap \Gamma_2$ is an induced substructure of both Γ_1 and Γ_2, there is a structure $\mathfrak{A}(\Gamma_1, \Gamma_2) \in \mathcal{A}$, called the *amalgam* of Γ_1 and Γ_2, and embeddings $f_i\colon \Gamma_i \to \mathfrak{A}(\Gamma_1, \Gamma_2)$ for $i \in \{1, 2\}$ such that $f_1\big|_{\Gamma_1 \cap \Gamma_2} = f_2\big|_{\Gamma_1 \cap \Gamma_2}$.

The following is a central theorem in model theory; a detailed proof of a more general statement can be found in [19, § 7]. A short self-contained proof for relational structures only can be found in the Appendix of [12].

2.17 Theorem (Fraïssé)**.** *Let $\mathcal{A}$ be an amalgamation class that contains countably many non-isomorphic τ-structures. Then there is a countably infinite, homogeneous τ-structure $\mathfrak{F}(\mathcal{A})$, called the* Fraïssé limit *of $\mathcal{A}$, such that $\mathcal{A}$ is exactly the set of finite structures that can be embedded into $\mathfrak{F}(\mathcal{A})$. Moreover, $\mathfrak{F}(\mathcal{A})$ is unique up to isomorphism.*

Proof sketch. Since $\mathcal{A}$ is closed under isomorphisms, the amalgamation property as stated in Definition 2.16 can equivalently be formulated as follows: For all structures $\Gamma_0, \Gamma_1, \Gamma_2 \in \mathcal{A}$ and embeddings $f_i \colon \Gamma_0 \to \Gamma_i$, $i \in \{1, 2\}$, there is a structure $\Gamma \in \mathcal{A}$ and embeddings $g_i \colon \Gamma_i \to \Gamma$, $i \in \{1, 2\}$, such that $g_1 \circ f_1 = g_2 \circ f_2$. Γ is called an *amalgam of Γ_1 and Γ_2 over Γ with respect to f_1 and f_2*.

Let $\Gamma_0, \Gamma_1, \Gamma_2, \ldots$ be an enumeration of representatives of all distinct isomorphism classes of structures in $\mathcal{A}$. We construct a chain $\Delta_0 \subseteq \Delta_1 \subseteq \Delta_2 \subseteq \ldots$ of structures in $\mathcal{A}$ as follows: Let $\Delta_0 \coloneqq \Gamma_0$ and assume that $\Delta_0, \ldots, \Delta_n$ are already constructed. Let $(\Gamma_{i_k}, \Gamma_{j_k}, f_k, g_k)_{1 \leq k \leq m}$ be an enumeration of all possible combinations of structures $\Gamma_{i_k}, \Gamma_{j_k}$ with $i_k, j_k < n$ and of embeddings $f_k \colon \Gamma_{i_k} \to \Gamma_{j_k}$ and $g_k \colon \Gamma_{i_k} \to \Delta_n$. We build a nested chain $\Delta_n \subseteq \Lambda_0 \subseteq \Lambda_1 \subseteq \ldots \subseteq \Lambda_m =: \Delta_{n+1}$ of structures in $\mathcal{A}$, where Λ_0 is an amalgam of Δ_n and Γ_n and, for $1 \leq k \leq m$, Λ_k is an amalgam of Λ_{k-1} and Γ_{j_k} over Γ_{i_k} with respect to f_{i_k} and g_{i_k}. Finally, let $\mathfrak{F}(\mathcal{A}) \coloneqq \bigcup_{n \in \mathbb{N}} \Delta_n$.

By construction, all structures in $\mathcal{A}$ embed into $\mathfrak{F}(\mathcal{A})$, and the following holds: For all structures $\Gamma_1, \Gamma_2 \in \mathcal{A}$, if Γ_1 is an induced substructure of Γ_2 and $f_1 \colon \Gamma_1 \to \mathfrak{F}(\mathcal{A})$ is an embedding, then there is some embedding $f_2 \colon \Gamma_2 \to \mathfrak{F}(\mathcal{A})$ that extends f_1. With a back-and-forth argument, it is easy show that in the case of countable structures, the latter property implies homogeneity. The uniqueness of $\mathfrak{F}(\mathcal{A})$ up to isomorphism can be shown with a back-and-forth argument as well. □

2.18 Example. The class $\mathcal{A}$ of finite structures with an irreflexive linear order is an amalgamation class; its Fraïssé limit is isomorphic to $(\mathbb{Q}; <)$. Note that $\mathcal{A}$ is the class of finite structures that embed into $(\mathbb{N}, <)$ and $(\mathbb{Z}, <)$ as well; those are however not homogeneous (e. g., the mapping $1 \mapsto 1, 2 \mapsto 3$ is a partial isomorphism of both structures that cannot be extended to an automorphism). Another example of an amalgamation class is the class of finite undirected graphs; its Fraïssé limit is the Erdős-Rényi random graph.

The following observation is helpful for working with concrete Fraïssé limits:

2.19 Lemma. *Let $\mathcal{A}$ be a class of τ-structures with Fraïssé limit $\mathfrak{F}(\mathcal{A})$ and let φ be some universal τ-sentence. Then φ holds in $\mathfrak{F}(\mathcal{A})$ if and only if φ holds in all structures of $\mathcal{A}$.*

Proof. Let $\varphi \equiv \forall x_1 \cdots \forall x_n \, (\psi(x_1, \ldots, x_n))$, where ψ is quantifier-free. Every $\Gamma \in \mathcal{A}$ is isomorphic to a substructure of $\mathfrak{F}(\mathcal{A})$ by the definition of Fraïssé limits. It is clear that universal sentences are preserved under taking substructures. Hence, if $\mathfrak{F}(\mathcal{A}) \models \varphi$, then also $\Gamma \models \varphi$. Conversely, let φ be true in every $\Gamma \in \mathcal{A}$ and let $a_1, \ldots, a_n \in \mathfrak{F}(\mathcal{A})$. Then $\mathfrak{F}(\mathcal{A})[\{a_1, \ldots, a_n\}] \in \mathcal{A}$ by the definition of Fraïssé limits, thus $\mathfrak{F}(\mathcal{A})[\{a_1, \ldots, a_n\}] \models \psi(a_1, \ldots, a_n)$ by assumption. Because ψ is quantifier-free, we obtain $\mathfrak{F}(\mathcal{A}) \models \psi(a_1, \ldots, a_n)$. □

2.5 Relational Reductions between Structures

In this section, we will introduce a central tool to compare the CSPs of two fixed templates with respect to their computational and descriptive complexity, namely pp-constructions. A structure Γ pp-constructs a structure Δ if it can »simulate« it in a certain sense; in particular, the existence of such a reduction implies that $\mathrm{CSP}(\Delta)$ is log-space reducible to $\mathrm{CSP}(\Gamma)$. As we will see in the next chapter, pp-constructions also preserve the expressibility of CSPs in fixed point logics.

The following are standard definitions:

2.20 Definition. Let Γ, Δ be two structures with the same domain. We say that Γ *pp-defines* Δ if each relation of Δ can be defined by a primitive positive formula over Γ.

2.21 Definition. Let Γ, Δ be two structures with signatures σ, τ, respectively. We say that Δ is a *pp-power* of Γ if there is some natural number $d \geq 1$ (the *dimension*) such that $D_\Delta = (D_\Gamma)^d$, and for every k-ary $R \in \tau$, the relation $\{(a_{11}, \ldots, a_{1d}, \ldots, a_{k1}, \ldots, a_{kd}) \mid (a_1, \ldots, a_k) \in R^\Delta\} \subseteq (D_\Gamma)^{d \cdot k}$ has a pp-definition in Γ.

2.22 Definition. We say that Γ *pp-contructs* Δ if Δ is homomorphically equivalent to a pp-power of Γ.

2.23 Definition. Let Γ, Δ be a structures with signatures τ, σ, respectively. We say that Γ *pp-interprets* Δ if there is a natural number $d \geq 1$ (the *dimension*), a subset $S \subseteq (D_\Gamma)^d$, an equivalence relation $\vartheta \subseteq S^2$ and a surjection $f \colon S \to D_\Delta$ such that $\ker(f) = \vartheta$, S and ϑ have a pp-definition in Γ and, for every relation $R \in \sigma$, the set $f^{-1}(R) \coloneqq \{(a_{11}, \ldots a_{1d}, \ldots, a_{k1}, \ldots, a_{kd}) \mid a_1, \ldots, a_k \in S \wedge (f(a_1), \ldots, f(a_k)) \in R^\Delta\}$ has a pp-definition in Γ as well.

2.24 Definition. For a class $\mathcal{K}$ of structures, let

- $\mathbf{D}(\mathcal{K})$ be the class of structures which are pp-definable in some structure in $\mathcal{K}$,
- $\mathbf{P}(\mathcal{K})$ be the class of structures which are a pp-power of some structure in $\mathcal{K}$,
- $\mathbf{I}(\mathcal{K})$ be the class of structures which are pp-interpretable in some structure in $\mathcal{K}$ and
- $\mathbf{H}(\mathcal{K})$ be the class of structures which are homomorphically equivalent to some structure in $\mathcal{K}$.

2.25 Lemma ([4, Lemma 3.8])**.** *Let $\mathcal{K}$ be a class of structures. Then*

$$\mathbf{D}(\mathcal{K}) \overset{(a)}{\subseteq} \mathbf{P}(\mathcal{K}) \overset{(b)}{\subseteq} \mathbf{I}(\mathcal{K}) \overset{(c)}{\subseteq} \mathbf{HP}(\mathcal{K}).$$

Moreover, $\mathbf{HH}(\mathcal{K}) \overset{(d)}{=} \mathbf{H}(\mathcal{K}), \mathbf{PP}(\mathcal{K}) \overset{(e)}{=} \mathbf{P}(\mathcal{K})$ *and* $\mathbf{PH}(\mathcal{K}) \overset{(f)}{\subseteq} \mathbf{HP}(\mathcal{K})$. *In particular,* $\mathbf{HPHP}(\mathcal{K}) = \mathbf{HP}(\mathcal{K})$, *i. e. pp-constructibility is transitive.*

Proof. (a) is immediate, as pp-definitions are pp-powers with dimension $d = 1$. Regarding (b), if Δ is a pp-power of Γ, it also has a pp-interpretation in Γ: The dimension of the pp-interpretation is the dimension of the pp-power and, using the notation of Definition 2.23, $S = (D_\Gamma)^d$ and ϑ is the equality relation.

For (c), let Γ, Δ be structures with signatures τ, σ, respectively, such that Γ pp-interprets Δ; let d, S, ϑ and f be as in Definition 2.23. Let Ξ be the σ-structure with domain $D_\Xi := (D_\Gamma)^d$ whose relations are given by $R^\Xi := f^{-1}(R^\Delta)$. Extend f arbitrarily to a mapping $f' \colon D_\Xi \to D_\Delta$ and choose some $g \colon D_\Delta \to S$ such that $f' \circ g = \mathrm{id}_{D_\Delta}$. Then $\Xi \to \Delta$ via f and $\Delta \to \Xi$ via g, hence Ξ and Δ are homomorphically equivalent. Thus, since Ξ is a pp-power of Γ, we obtain $\Delta \in \mathbf{HP}(\Gamma)$.

(d) holds because the composition of two homomorphisms is again a homomorphism. (e) follows more or less trivially from the definitions.

It remains to show (f). Let $\Delta \in \mathbf{PH}(\Gamma)$, i. e. there is some structure Λ such that Γ and Λ are homomorphically equivalent and Δ is a pp-power of Λ of dimension, say, d. We want to show $\Delta \in \mathbf{HP}(\Gamma)$, so we want to find a structure Ξ which is a pp-power of Γ such that Δ and Ξ are homomorphically equivalent. Define a d-dimensional pp-power of Γ with the same signature as Δ like this: We know that, by the definition of pp-powers, there is a pp-formula ψ for every relation symbol R such that $R^\Delta(a_1, \ldots, a_k) \iff \Lambda \models \psi(a_{11}, \ldots, a_{1d}, \ldots, a_{k1}, \ldots, a_{kd})$; now define $R^\Xi(b_1, \ldots, b_k) :\iff \Gamma \models \psi(b_{11}, \ldots, b_{1d}, \ldots, b_{k1}, \ldots, b_{kd})$. Let $f \colon \Lambda \to \Gamma$ and $g \colon \Gamma \to \Lambda$ be homomorphisms, then the component-wise actions of f and g on $(D_\Lambda)^d$ and $(D_\Gamma)^d$, respectively, homomorphically map Δ and Ξ to each other. □

The following statement demonstrates the utility of pp-constructions in the context of CSPs:

2.26 Proposition. *If Γ pp-constructs Δ, then* $\mathrm{CSP}(\Delta)$ *is log-space reducible to* $\mathrm{CSP}(\Gamma)$.

Proof. We first observe that two homomorphically equivalent structures have the same CSP: Homomorphisms transform solutions of an instance of one CSP into solutions of the corresponding instance of the other CSP, and vice versa. Hence, we can assume that Δ is a pp-power of Γ; let d be its dimension.

Let φ be an arbitrary instance of $\mathrm{CSP}(\Delta)$. Without loss of generality, φ does not contain any equality constraints. We transform φ into an instance $\tilde{\varphi}$ of $\mathrm{CSP}(\Gamma)$ as follows: For each constraint $R(x_1, \ldots, x_k)$ of φ, there is, by assumption, a pp-definition $\psi_R(x_{11}, \ldots, x_{1d}, \ldots, x_{k1}, \ldots, x_{kd})$ of $\{(a_{11}, \ldots, a_{1d}, \ldots, a_{k1}, \ldots, a_{kd}) \mid (a_1, \ldots, a_k) \in R^\Delta\}$. We replace $R(x_1, \ldots, x_k)$ by ψ_R, where, if necessary, we rename variables such that for any two different relation symbols R and S, ψ_R and ψ_S do not have any existentially bound variables in common. Subsequently, we remove all quantifiers, which yields a conjunction $\tilde{\varphi}(x_{11}, \ldots, x_{1d}, \ldots, x_{k1}, \ldots, x_{kd})$ of atomic formulas. By construction, φ is satisfiable over Δ if and only if $\tilde{\varphi}$ is satisfiable over Γ. The reduction is obviously feasible in logarithmic space. □

2.27 Lemma ([4, Lemma 3.9])**.** *Let Γ be an at most countable ω-categorical, model-complete core and let Δ be the expansion of Γ by a constant. Then* $\Delta \in \mathbf{HP}(\Gamma)$.

2.6 Fixed Point Logics

Let S be some finite set and let 2^S be the power set of S. Take an arbitrary operator $F\colon 2^S \to 2^S$ and consider the sequences defined by

$$U_0 \coloneqq \varnothing, \quad U_{i+1} \coloneqq U_i \cup F(U_i) \tag{2}$$

and

$$U_0 \coloneqq S, \quad U_{i+1} \coloneqq U_i \cap F(U_i). \tag{3}$$

Sequence (2) is increasing, sequence (3) is decreasing; since S is finite, in both cases there will eventually be some index i such that $U_i = U_{i+1}$. This implies $U_i = U_j$ for all $j \geq i$; we call U_i the *limit* of the sequence. The limit of (2) is called the *inflationary fixed point* of F and denoted by $\mathrm{Ifp}(F)$; the limit of (3) is called the *deflationary fixed point* of F and denoted by $\mathrm{Dfp}(F)$.[†]

Let Γ be a σ-structure, let R be a k-ary relation symbol with $R \notin \sigma$ and let $\varphi(x_1, \ldots, x_k, y_1, \ldots, y_l)$ be a $(\sigma \cup \{R\})$-formula. For some $X \subseteq (D_\Gamma)^k$, let (Γ, X) be the $(\sigma \cup \{R\})$-structure extending Γ where R is interpreted as X.

For $c_1, \ldots, c_l \in D_\Gamma$, we define the operator $F^\Gamma_{\varphi(x_1,\ldots,x_k,c_1,\ldots,c_l)}$ as follows:

$$F^\Gamma_{\varphi(x_1,\ldots,x_k,c_1,\ldots,c_l)}\colon \begin{cases} (D_\Gamma)^k \to (D_\Gamma)^k \\ X \mapsto \{(a_1, \ldots, a_k)\colon (\Gamma, X) \models \varphi(a_1, \ldots, a_k, c_1, \ldots, c_l)\}. \end{cases}$$

We are now ready to define *inflationary fixed point logic,* denoted by IFP:

2.28 Definition (Syntax of IFP)**.** Let σ be some signature. Then the set of IFP σ-formulas is inductively defined as follows:

- Every atomic σ-formula is an IFP σ-formula.
- Every formula built from IFP σ-formulas by the first-order constructors $\wedge, \vee, \neg, \exists$ and $\forall$ is an IFP σ-formula.
- If $\varphi(x_1, \ldots, x_k, y_1, \ldots, y_l)$ is an IFP $(\sigma \cup \{R\})$-formula for some k-ary relation symbol $R \notin \sigma$, then $\big[\mathsf{ifp}_{R,(x_1,\ldots,x_k)}\varphi\big]$ is an IFP σ-formula with free variables $x_1, \ldots, x_k, y_1, \ldots, y_l$.

2.29 Definition (Semantics of IFP)**.** The semantics of atomic formulas and of $\wedge, \vee, \neg, \exists$ and $\forall$ is defined in the usual way like for first-order logic. Let Γ be a σ-structure and let $\varphi(x_1, \ldots, x_k, y_1, \ldots, y_l)$ be an IFP $(\sigma \cup \{R\})$-formula for some k-ary relation symbol $R \notin \sigma$. Then for $a_1, \ldots, a_k, c_1, \ldots, c_l \in D_\Gamma$, we define

$$\Gamma \models \big[\mathsf{ifp}_{R,(x_1,\ldots,x_k)}\varphi\big](a_1, \ldots, a_k, c_1, \ldots, c_l) \;:\Longleftrightarrow\; (a_1, \ldots, a_k) \in \mathrm{Ifp}\big(F^\Gamma_{\varphi(x_1,\ldots,x_k,c_1,\ldots,c_l)}\big).$$

We use $\big[\mathsf{dfp}_{R,(x_1,\ldots,x_k)}\varphi\big]$ as an abbreviation for $\neg\big[\mathsf{ifp}_{R,(x_1,\ldots,x_k)}\neg\varphi[R\backslash\neg R]\big]$, where $\varphi[R\backslash\neg R]$ is the formula obtained from φ when R is replaced by $\neg R$ everywhere. It is straightforward to show that, as the notation suggests, $\Gamma \models \big[\mathsf{dfp}_{R,x}\varphi\big](a, c) \iff a \in \mathrm{Dfp}(F^\Gamma_{\varphi(x,c)})$. We will need one more logic:

[†] This terminology can be explained as follows: A set X is called a *fixed point* of an operator G if $G(X) = X$. Now $\mathrm{Ifp}(F)$ is a fixed point of the operator G defined by $G(X) \coloneqq X \cup F(X)$ and $\mathrm{Dfp}(F)$ is a fixed point of the operator G defined by $G(X) \coloneqq X \cap F(X)$.

2.30 Definition. Datalog is the existential positive fragment of IFP.

We say that a computational problem is *expressible* in a logic L if there is a sentence φ in L that defines the class of all structures which are negative instances of the problem.[†]

It is well-known that graph connectivity is not definable by a first-order formula. It can, however, be expressed in IFP, as the following standard example shows:

2.31 Example. Let $G = (V; E)$ be a finite graph, $a, c \in V$ and let U be a new unary relation symbol. Let $\varphi(x, y) \equiv x = y \vee \exists z(E(z, x) \wedge U(z))$. Consider sequence (2) for the operator $F := F^G_{\varphi(x,c)}$: With an easy induction, we obtain that U_i is the set of vertices reachable from c on a path of length at most i; thus, $\mathrm{Ifp}(F)$ is the set of all vertices reachable from c. Hence, $G \models \big[\mathsf{ifp}_{U,x}\varphi(x, y)\big](a, c)$ if and only if a is reachable from c, which implies that $\forall v \forall w \big[\mathsf{ifp}_{U,x}\varphi(x, y)\big](v, w)$ defines the class of connected finite graphs.

Besides IFP, there are various other fixed point logics like *least fixed point logic* (LFP) and *deflationary fixed point logic* (DFP). They all have, however, equal expressive power, for which reason they are often simply referred to as *fixed point logic* (FP). It has been shown that FP captures polynomial time over the class of ordered structures (see e. g. [20, Theorem 10.14]). An important extension of FP is *fixed point logic with counting* (FPC); we will use this logic as a black box here. There was a prospect that FPC would capture all of polynomial time, as it captures it over many important classes like trees or planar graphs ([14]); however, it turned out that the problem of whether a system of linear equations over a finite, nontrivial abelian group has a solution is not expressible in FPC, not even in simple cases:

2.32 Theorem ([15])**.** *Let* $\mathrm{E}_{\mathbb{Z}_2,3}$ *be the structure* $(\mathbb{Z}_2; R^2_0, R^3_0, R^3_1)$*, where* $R^i_a = \{(x_1, \ldots, x_i) \colon x_1 + \ldots + x_i = a\}$*. Then* $\mathrm{CSP}(\mathrm{E}_{\mathbb{Z}_2,3})$ *is not expressible in* FPC*.*

We will cite an important relationship between the logics we defined and the concept of pp-constructibility:

2.33 Theorem (see [18, Theorem 8.7.7])**.** *Let* Γ, Δ *be structures with finite signatures such that* $\mathrm{CSP}(\Gamma)$ *is expressible* FPC *(resp.* FP*,* Datalog*) and* $\Delta \in \mathbf{HP}(\Gamma)$*. Then* $\mathrm{CSP}(\Delta)$ *is expressible in* FPC *(resp.* FP*,* Datalog*) as well.*

Standard structures in the context of pp-constructions are

- $(\{0, 1\}; \mathrm{NAE})$, where $\mathrm{NAE} = \{0, 1\}^3 \backslash \{(0, 0, 0), (1, 1, 1)\}$,
- $(\{0, 1\}; \mathrm{1IN3})$, where $\mathrm{1IN3} = \{(0, 0, 1), (0, 1, 0), (1, 0, 0)\}$ and
- $\mathbb{K}_3 = (\{0, 1, 2\}; \neq)$.

They are all known to have NP-complete CSPs. Furthermore, they have the following remarkable property:

2.34 Proposition. $(\{0, 1\}; \mathrm{NAE}), (\{0, 1\}; \mathrm{1IN3})$ *and* $\mathbb{K}_3$ *pp-construct all finite structures.*

[†] Note that some logics like Datalog do not contain negation.

2.35 Corollary. *If a structure* Γ *pp-constructs one of* $(\{0,1\};\mathrm{NAE}), (\{0,1\};\mathrm{1IN3})$ *and* $\mathbb{K}_3$*, then* $\mathrm{CSP}(\Gamma)$ *is not expressible in* FPC.

Proof. If Γ pp-constructs one of these structures, it pp-constructs all finite structures by Proposition 2.34 and the transitivity of pp-constructibility (Lemma 2.25). In particular, it pp-constructs the structure $\mathrm{E}_{\mathbb{Z}_2,3}$ from Theorem 2.32, whose CSP is not expressible in FPC. Hence, $\mathrm{CSP}(\Gamma)$ cannot be in FPC either by Theorem 2.33. □

Note that a structure that pp-constructs a structure with an NP-complete CSP has itself an NP-complete CSP by Proposition 2.26. However, statements about the expressibility of CSPs in fixed point logics remain interesting even if P = NP.

We conclude the section with the definition of a finite-variable logic with counting and its relationship with FPC.

2.36 Definition. $\mathcal{L}^k$ is the fragment of first-order logic that only uses the variables $x_1, \ldots, x_k$. $\mathcal{C}^k$ is the extension of $\mathcal{L}^k$ by a *counting quantifier* $\exists^i$ for each $i \in \mathbb{N}$, where $\Gamma \models \exists^i x(\varphi(x))$ if and only if there are at least i distinct elements $a \in D_\Gamma$ such that $\Gamma \models \varphi(a)$.

The *bijective k-pebble game* is a game played on two structures Γ and Δ of the same signature by two players, which are called the *Spoiler* and the *Duplicator*. The setup includes k pairs of *pebbles* $(a_1, b_1), \ldots, (a_k, b_k)$. The game consists of potentially infinitely many rounds, each of which proceeds as follows:

- The Spoiler chooses some $i \in \{1, \ldots, k\}$.
- The Duplicator selects a bijection $f\colon \Gamma \to \Delta$ under the following restriction: For each $j \in \{1, \ldots, k\} \backslash \{i\}$ such that (a_i, b_i) has already been placed on the board, $f(a_j) = b_j$ must hold.
- The Spoiler places a_i on any element of Γ and b_i on $f(a_i)$.

If, after any round, the partial map between Γ and Δ induced by the pebbles on the board is not a partial isomorphism, the Spoiler wins the game. If the game lasts forever, the Duplicator wins.

Let Γ and Δ be two τ-structures. We write $\Gamma \equiv_{\mathcal{C}^k} \Delta$ if for every τ-sentence $\varphi \in \mathcal{C}^k$, $\Gamma \models \varphi \iff \Delta \models \varphi$.

2.37 Lemma ([16])**.** *For two* τ*-structures* Γ *and* Δ*, we have* $\Gamma \equiv_{\mathcal{C}^k} \Delta$ *if and only if the Duplicator has a winning strategy for the bijective k-pebble game on* Γ *and* Δ*.*

2.38 Theorem (Immerman-Lander, see e. g. [17])**.** *Let* φ *be an* FPC τ*-sentence. Then there is some* $k \in \mathbb{N}$ *such that, for all finite* τ*-structures* Γ *and* Δ*, if* $\Gamma \equiv_{\mathcal{C}^k} \Delta$*, then* $\Gamma \models \varphi \iff \Delta \models \varphi$.

2.7 Equality Constraint Satisfaction Problems

An *equality constraint language* is a reduct of $(\mathbb{N}; =)$. Note that we could choose any countable domain (e. g. $\mathbb{Q}$) instead and would obtain an isomorphic structure. In this section, we will show that the CSP of an equality constraint language is either expressible in Datalog, or it pp-constructs $\mathbb{K}_3$.

We first summarize an argument which is only implicit in [7]:

2.39 Proposition. *Let Γ be an equality constraint language. If Γ does not have a constant unary or an injective binary polymorphism, then it pp-defines every relation R with a first-order definition in $(\mathbb{N};=)$.*

Proof. As Γ is a reduct of $(\mathbb{N};=)$, its automorphism group is the full symmetric group on $\mathbb{N}$, which is obviously oligomorphic. Hence, R is the union of finitely many orbits $\mathcal{O}(a_1),\dots,\mathcal{O}(a_l)$. For a tuple $a \in \mathbb{N}^k$, the formula $\varrho_a(x) \equiv \bigwedge_{a_i=a_j} x_i = x_j \wedge \bigwedge_{a_i\neq a_j} x_i \neq x_j$ defines the orbit of a; hence, $R(x) \iff \bigvee_{1\leq i\leq l} \varrho_{a_i}(x)$. Since Γ does not have a constant unary polymorphism, the binary inequality relation has a pp-definition in Γ by [7, Lemma 4]. Therefore, R has an existential positive definition in Γ, which implies $\mathrm{End}(\Gamma) \curvearrowright R$ by Theorem 2.6. Again because Γ does not have a constant unary or an injective binary polymorphism, all polymorphisms of Γ only depend on one argument by [7, Lemma 5 and Theorem 4]. Thus, $\mathrm{End}(\Gamma) \curvearrowright R$ implies $\mathrm{Pol}(\Gamma) \curvearrowright R$, so Γ pp-defines R by Theorem 2.6. □

The following Lemma is inspired by a gadget reduction in [7, Lemma 6].

2.40 Lemma. *Let $T := \{(x,y,z) \in \mathbb{N}^3 \mid (x = y \wedge y \neq z) \vee (x \neq y \wedge y = z)\}$. Then $(\mathbb{N};T,0,1,2)$ pp-interprets $\mathbb{K}_3$.*

Proof. We work with the notation from Definition 2.23. The dimension is $d = 2$. The subset S is given by $S = \{(0,0),(1,1),(2,0),(2,1)\} = \{(c,d) \in \mathbb{N}^2 \colon T(d,c,2) \wedge T(0,d,1)\}$ and the equivalence relation ϑ by $\vartheta = \{(a,b) \in S^2 \colon a_1 = b_1\}$ (remember that equality is always part of the language). The mapping f does $(0,0) \mapsto 0, (1,1) \mapsto 1, (2,0) \mapsto 2$ and $(2,1) \mapsto 2$. Now, for $(a,b) \in S^2$, we have $\mathbb{K}_3 \models f(a) \neq f(b) \iff (\mathbb{N};T,0,1,2) \models a_1 \neq b_1$. Because $T(x,x,y)$ is a pp-definition of $x \neq y$, this implies that $f^{-1}(\neq)$ is also pp-definable in $(\mathbb{N};T,0,1,2)$. □

2.41 Proposition. *Let Γ be an equality constraint language. If Γ does not have a constant unary or injective binary polymorphism, then Γ pp-constructs $\mathbb{K}_3$.*

Proof. As Γ is a reduct $(\mathbb{N};=)$, it is ω-categorical and has the full symmetric group on $\mathbb{N}$ as its automorphism group. Since Γ does not have a constant endomorphism, every endomorphism of Γ is injective by [7, Lemma 3] and hence locally interpolated by automorphisms of Γ. Hence, Γ is a model-complete core by Theorem 2.14.

By Proposition 2.39, we know that Γ pp-defines the relation T defined in Lemma 2.40. Hence, by Lemma 2.40, $\mathbb{K}_3 \in \mathbf{I}(\Gamma,0,1,2)$. Because Γ is an ω-categorical, model-complete core, Lemma 2.25 and Lemma 2.27 imply that $\mathbb{K}_3 \in \mathbf{HP}(\Gamma)$. □

It remains to study the complexity of equality languages which are preserved by a constant unary or an injective binary polymorphism. Consider the relation

$$I := \{(a,b,c,d) \in \mathbb{N}^4 \mid a = b \to c = d\}.$$

2.42 Lemma ([8, Proposition 43])**.** *Let R be a relation with a first-order definition in $(\mathbb{N};=)$. Then the following two statements are equivalent:*

(1) *R is preserved by an injective binary polymorphism.*

(2) *R has a pp-definition in $(\mathbb{N};I,\neq)$.*

Algorithm SolveE
Input: An instance Γ of $\mathrm{CSP}(\mathbb{N}; I, \neq)$
Output: $\top$ or $\bot$

$\Theta \leftarrow \{(x, x) \colon x \in \Gamma\}$
while Θ changes:
  for $(x_1, x_2, x_3, x_4) \in I^\Gamma$:
    if $(x_1, x_2) \in \Theta$:
      $\Theta \leftarrow \mathrm{tcl}(\Theta \cup \{(x_3, x_4), (x_4, x_3)\})$ // tcl is the transitive closure
for $(x, y) \in {\neq}^\Gamma$:
  if $(x, y) \in \Theta$:
    return $\bot$
return $\top$

Figure 2. A Datalog algorithm for a generic equality CSP preserved by an injective binary polymorphism.

2.43 Lemma. *The algorithm in Figure 2 is sound and complete for* $\mathrm{CSP}(\mathbb{N}; I, \neq)$, *i. e.*

$$\Gamma \to (\mathbb{N}; I, \neq) \iff \textsc{SolveE}(\Gamma) = \top.$$

Proof. For some $i \geq 0$, we denote by Θ^i the state of the program variable Θ after the i-th iteration of the while-loop. Note that $\Theta^i \subseteq \Theta^{i+1}$; in particular, Θ reaches a limit Θ^∞ after finitely many iterations.

»⇒«: Let $t \colon \Gamma \to (\mathbb{N}, I, \neq)$ be a homomorphism. By induction on i, it is immediate that $\Theta^i(x, y)$ implies $t(x) = t(y)$. Whenever $(x, y) \in {\neq}^\Gamma$, we have $t(x) \neq t(y)$, since t is a homomorphism; hence, $\Theta^\infty(x, y)$ cannot hold by the previous argument.

»⇐«: Let $t \colon \Gamma \to \mathbb{N}$ be any mapping such that $t(x) = t(y)$ if and only if $(x, y) \in \Theta^\infty$. For some constraint ${\neq}^\Gamma(x, y)$, it holds that $(x, y) \notin \Theta^\infty$ since $\textsc{SolveE}(\Gamma) = \top$; thus, $t(x) \neq t(y)$. For a constraint of the form $I^\Gamma(x_1, x_2, x_3, x_4)$, we distinguish two cases: If $(x_1, x_2) \notin \Theta^\infty$, then $t(x_1) \neq t(x_2)$, which implies $I^\mathbb{N}(t(x_1), t(x_2), t(x_3), t(x_4))$. If $(x_1, x_2) \in \Theta^\infty$, however, then (x_3, x_4) is added to Θ in some iteration of the first for-loop; hence $t(x_3) \neq t(x_4)$, which implies $I^\mathbb{N}(t(x_1), t(x_2), t(x_3), t(x_4))$ too. □

2.44 Corollary. $\mathrm{CSP}(\mathbb{N}, I, \neq)$ *is expressible in* Datalog.

Proof. We rewrite the algorithm in Figure 2 into a Datalog formula. It is easy to see that $\textsc{SolveE}(\Gamma) = \bot \iff \Gamma \models \varphi$, where

$$\varphi \equiv \exists x, y \left({\neq}(x, y) \wedge \left[\mathsf{ifp}_{\Theta, (x_3, x_4)} \left[\mathrm{tcl}\, \varrho(x_3, x_4) \right] \right](x, y) \right)$$

and

$$\varrho(x_3, x_4) \equiv x_3 = x_4 \vee \exists x_1, x_2 \left(\Theta(x_1, x_2) \wedge I(x_1, x_2, x_3, x_4) \right).$$

$\left[\mathrm{tcl}\, \varrho\right](x_3, x_4)$ in turn is an abbreviation for the formula

$$\left[\mathsf{ifp}_{Z, (s,t)}\, \varrho(s, t) \vee \exists z (\varrho(s, z) \wedge Z(z, t)) \right](x_3, x_4);$$

it computes the transitive closure of ϱ, cf. Example 2.31.

Hence, $\Gamma \not\to (\mathbb{N}; I, \neq) \iff \Gamma \models \varphi$ by Lemma 2.7. □

2.45 Proposition. *Let* Γ *be an equality constraint language with finite signature. If* Γ *has a constant unary or injective binary polymorphism, then* CSP(Γ) *is expressible in* Datalog.

Proof. If Γ has a constant unary polymorphism, then an instance of CSP(Γ) is unsatisfiable if and only if it contains an empty relation, which is clearly definable in Datalog, because Γ has finite signature. If it has an injective binary polymorphism, we have $\Gamma \in \mathbf{D}(\mathbb{N}, I, \neq)$ by Lemma 2.42. As CSP($\mathbb{N}, I, \neq$) is expressible in Datalog by Corollary 2.44, so is CSP(Γ) by Theorem 2.33. □

We obtain the following corollary:

2.46 Theorem. *Let* Γ *be an equality constraint language with finite signature. Then either* Γ *pp-constructs* $\mathbb{K}_3$, *or* CSP(Γ) *is expressible in* Datalog.

Proof. If Γ has a constant unary or an injective binary polymorphism, then CSP(Γ) is expressible in Datalog by Proposition 2.45. If this is not the case, then Γ pp-constructs $\mathbb{K}_3$ by Proposition 2.41. □

3 Phylogeny Problems

In this chapter, we will introduce phylogeny problems, which are computational problems concerning the hierarchy of common ancestors of leaves in binary trees. A key result is the existence of the generic leaf structure $(\mathbb{L}; C)$, which is obtained as a Fraïssé limit, such that every phylogeny problem is equivalent to the CSP of a reduct of $(\mathbb{L}; C)$. A study of some of these reducts yields a classification of phylogeny problems regarding their expressibility in Datalog.

3.1 Leaf Structures

A *tree* is a finite undirected graph which is connected and acyclic. A *binary rooted tree* is a tree with a designated vertex r, the *root,* such that r has degree 2 and all other vertices either have degree 3 or 1; the latter are called the *leaves.* We denote the set of vertices of a tree T by $V(T)$ and the set of its leaves by $L(T)$.

It is easy to see that in every tree, there is a unique shortest path between any two distinct vertices. For $x, y \in V(T)$, we write $x \leq y$ if the path from x to the root contains y and $x < y$ if $x \leq y$ and $x \neq y$. Note that $\leq$ is a partial order on $V(T)$. The *youngest common ancestor* of a set $X \subseteq V(T)$, shortly $\mathrm{yca}(X)$, is the lowest upper bound of X with respect to $\leq$; it is uniquely determined by X.

3.1 Definition. Let T be a tree. On $L(T)$, we can define a ternary relation C by

$$C(x, y, z) \iff \mathrm{yca}(\{y, z\}) < \mathrm{yca}(\{x, y, z\}).$$

The *leaf structure* of T is the structure $(L(T); C)$ and T is called its *underlying tree.*

3.2 Definition. Let T be a rooted tree and $X_1, X_2 \subseteq V(T)$. We write $X_1|X_2$ if $\mathrm{yca}(X_1) \not\leq \mathrm{yca}(X_2)$ and $\mathrm{yca}(X_2) \not\leq \mathrm{yca}(X_1)$. We also write $x_1, \ldots, x_n|y_1, \ldots, y_m$ if $\{x_1, \ldots, x_n\}|\{y_1, \ldots, y_m\}$.

3.3 Lemma. *Let T be a binary rooted tree with leaf structure $(L(T); C)$ and x, y, z, u, $x_1, \ldots, x_n$, $y_1, \ldots, y_m \in L(T)$. Then the following holds:*

(1) $C(x, y, z) \iff x|yz$.

(2) *Exactly one of* $x|yz$, $y|xz$, $z|xy$ *and* $x = y = z$ *holds.*

(3) $x_1, \ldots, x_n|y_1, \ldots, y_m \iff \bigwedge_{i,j \leq n} \bigwedge_{k,l \leq m} (x_i x_j|y_k \wedge x_i|y_k y_l)$.

(4) $y|xz \wedge x|zu \implies y|xu$.

(5) $x|yu \wedge x|zu \implies x|yz$.

Proof. (1) follows directly from the definitions; (2) holds because T is binary. »$\overset{(3)}{\Rightarrow}$« is trivial; for »$\overset{(3)}{\Leftarrow}$«, let $X := \{x_1, \ldots, x_n\}, Y := \{y_1, \ldots, y_m\}$. The claim is trivial for $|Y| = 1$, so let $|Y| \geq 2$ assume towards contradiction that (without loss of generality) $\mathrm{yca}(X) \leq \mathrm{yca}(Y)$. Because $\mathrm{yca}(Y)$ is the least upper bound of Y, there must be some $y_1 \in Y$ in the left subtree of the subtree of T rooted at $\mathrm{yca}(Y)$ and some $y_2 \in Y$ in the right subtree. Due

to $\mathrm{yca}(X) \leq \mathrm{yca}(Y)$, one of those subtrees must contain some $x \in X$, hence $xy_1|y_2$ or $xy_2|y_1$, contradiction. (4) holds because $\mathrm{yca}(\{x,u\}) = \mathrm{yca}(\{x,z\}) < \mathrm{yca}(\{y,x\})$. For (5), suppose that $x|yz$ does not hold, but instead, without loss of generality, $y|xz$. From $x|zu$ it follows by (4) that $y|xu$, contradicting the assumption $x|yu$. □

3.4 Lemma. *Let T be an arbitrary finite rooted tree. Then there is a finite binary rooted tree T' with the same set of leaves such that, for all $a,b,c \in L(T)$, $a|bc$ in the leaf structure of T implies $a|bc$ in the leaf structure of T'.*

Proof. We inductively transform T into T'. If $|T| = 1$, then we set $T' := T$. Let $|T| \geq 2$, let $T_1, \ldots, T_n$ be the (maximal) subtrees of T rooted at the children of the root of T and assume that $T'_1, \ldots, T'_n$ are already constructed. Let T' be the tree obtained from T as in Figure 3 and let $a,b,c \in L(T)$ such that $a|bc$.

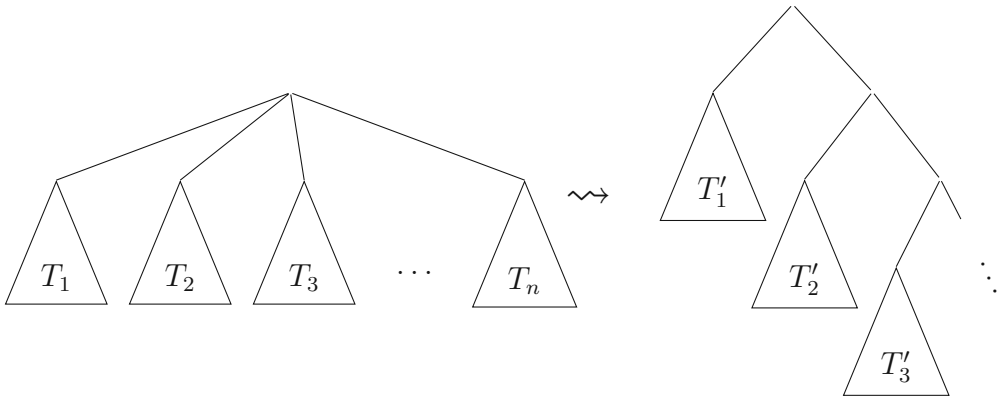


Figure 3: Transformation of an arbitrary rooted tree into a binary rooted tree that preserves the C-relation.

We distinguish two cases: If there is some $i \in \{1, \ldots, n\}$ such that $a,b,c \in T_i$, then we have $a|bc$ in T'_i by the induction hypothesis. The second possibility is that there are $i \neq j \in \{1, \ldots, n\}$ such that $a \in T_i$ and $b,c \in T_j$. In this case, $a|bc$ holds in the leaf structure of T' by the construction of T'. □

From now on, unless otherwhise stated, all considered trees will be binary rooted trees; this will, in the light of Lemma 3.4, make no difference in many problems we will discuss.

3.5 Definition. A *phylogeny formula* is a quantifier-free formula built from atomic formulas of the form $x|yz$ or $x = y$.

For a finite set Φ of phylogeny formulas, we are interested in the following problem:

Phylo(Φ)
INSTANCE: A finite set V of variables and a set Ψ of formulas, where each $\psi \in \Psi$ is obtained from some $\varphi \in \Phi$ by replacing all variables from φ by variables from V.
QUESTION: Is there a binary rooted tree T and an assignment $s\colon V \to L(T)$, such that $(L(T); C)$ satisfies all formulas in Ψ under the assignment s?

3.2 The Fraïssé Limit of the Class of Leaf Structures

Crucially, there is a special structure $(\mathbb{L}; C)$ such that every phylogeny problem is equivalent to the CSP of a reduct of $(\mathbb{L}; C)$. The construction of this generic structure is our next goal:

3.6 Proposition. *The class of all leaf structures of finite binary rooted trees is an amalgamation class.*

Proof. Let T_1, T_2 be two finite trees. Let T_{11}, T_{21} and T_{12}, T_{22} denote the left and right subtrees of them, respectively; let L_i and L_{ij} denote the leaf structures of T_i and T_{ij} for $i, j \in \{1, 2\}$.

Assume that $L_1 \cap L_2$ induces a substructure of both L_1 and L_2. We will show the existence of a tree T and of embeddings $f_i \colon L_i \to (L(T); C)$ for $i \in \{1, 2\}$ such that $f_1(a) = f_2(a)$ for all $a \in L_1 \cap L_2$. We proceed by induction on $|L_1| + |L_2|$ and distinguish, without loss of generality, two cases:

- Case 1: $L_1 \cap L_2 = (L_{11} \cap L_{21}) \cup (L_{12} \cap L_{22})$.

  By induction hypothesis, there are amalgams $\mathfrak{A}(L_{11}, L_{21})$ and $\mathfrak{A}(L_{12}, L_{22})$ of L_{11}, L_{21} and L_{12}, L_{22}, respectively, as well as suitable embeddings $f_{11}, f_{21}, f_{12}, f_{22}$. Let T be the tree with the underlying tree of $\mathfrak{A}(L_{11}, L_{21})$ as its left subtree and the underlying tree of $\mathfrak{A}(L_{12}, L_{22})$ as its right subtree. Obviously, L_1 and L_2 embed into $(L(T); C)$ via the functions $f_1 \coloneqq f_{11} \cup f_{12}$ and $f_2 \coloneqq f_{21} \cup f_{22}$, respectively, and those mappings coincide on $L_1 \cap L_2$.

- Case 2: $L_{11} \cap L_{21} \neq \varnothing$ and $L_{11} \cap L_{22} \neq \varnothing$.

  Let $a \in L_{11} \cap L_{21}$ and $b \in L_{11} \cap L_{22}$. We observe that $L_{12} \cap L_2 = \varnothing$: If there was some $c \in L_{12} \cap L_{21}$, then $L_1 \models ab|c$, but $L_2 \models ac|b$, contradiction. Similiarily, $L_{12} \cap L_{22} = \varnothing$, for if there was some $d \in L_{12} \cap L_{22}$, we would obtain $L_1 \models ab|d$ and $L_2 \models a|bd$. By induction hypothesis, there is an amalgam $\mathfrak{A}(L_{11}, L_2)$ of L_{11} and L_2 with embeddings f_{11} and f_2, respectively. Let T be the tree with the underlying tree of $\mathfrak{A}(L_{11}, L_2)$ as its left subtree and T_{12} as its right subtree. Since $L_{12} \cap L_2 = \varnothing$, we get that $f_1 \coloneqq f_{11} \cup \mathrm{id}_{L_{12}}$ and f_2 are suitable embeddings of L_1 and L_2, respectively, into $(L(T); C)$. □

Combining Theorem 2.17 and Proposition 3.6, we obtain the Fraïssé limit of the class of all finite leaf structures. We denote this special structure by $(\mathbb{L}; C)$. We can take statements (1) and (3) of Lemma 3.3 as definitions of $|$ on $(\mathbb{L}; C)$ and observe that, for arbitrary $x, y, z, u \in \mathbb{L}$, statements (2), (4) and (5) are true as well; this follows a fortiori from Lemma 2.19.

For an arbitrary finite set $X \subseteq \mathbb{L}$ with $|X| \geq 2$, there is a partition of X into two nonempty sets X_0 and X_1 such that $X_0|X_1$, viz. the elements of X which are leaves of the right and left subtree of the underlying tree of $(\mathbb{L}; C)[X]$. This partition is unique up to interchanging X_0 and X_1. It is sometimes useful to exclude this ambiguity; this can be done as follows (see [5, § 3.5]): A linear order $\prec$ on a leaf structure $(L; C)$ is called *convex* if $x \prec y \prec z$ implies $x|yz$ or $xy|z$. The class $\mathcal{A}'$ of all leaf structures with a convex linear order is an amalgamation class as well; its Fraïssé limit is isomorphic to an extension

$(\mathbb{L}; C, \prec)$ of $(\mathbb{L}; C)$ by a convex linear order $\prec$.[†] Now there is a unique nontrivial partition $\{X_0, X_1\}$ of X such that $X_0 \prec X_1$.

As $(\mathbb{L}; C)$ is homogeneous and has a finite signature, we get the following result from Proposition 2.8 and Theorem 2.9:

3.7 Proposition. *$(\mathbb{L}; C)$ is ω-categorical and admits quantifier-elimination.*

Now we are able to prove a statement which explains the importance of $(\mathbb{L}; C)$ in the context of phylogeny problems:

3.8 Proposition (cf. [1, p. 6 f.])**.** *There is the following correspondence between phylogeny problems and CSPs of reducts of* $(\mathbb{L}; C)$:

(1) *For every phylogeny problem* $\mathrm{Phylo}(\Phi)$*, there is a reduct* Γ_Φ *of* $(\mathbb{L}; C)$ *such that* $\mathrm{Phylo}(\Phi)$ *is (trivially) reducible to* $\mathrm{CSP}(\Gamma_\Phi)$.

(2) *For every reduct* $\Gamma = (\mathbb{L}; R_1, \dots, R_n)$ *of* $(\mathbb{L}; C)$*, there is a set* Φ *of phylogeny formulas such that* $\mathrm{CSP}(\Gamma)$ *is (trivially) reducible to* $\mathrm{Phylo}(\Phi)$.

Proof. (1) For every $\varphi(x_1, \dots, x_k) \in \Phi$, let R_φ be a relation symbol of arity k. For any leaf structure $(L; C)$, we define $R_\varphi^{(L;C)} := \{(a_1, \dots, a_k) \in L \colon (L; C) \models \varphi(a_1, \dots, a_k)\}$. The structure Γ_Φ is given by $\Gamma_\Phi := (\mathbb{L}; (R_\varphi^{(\mathbb{L};C)})_{\varphi\in\Phi})$.

Let (V, Ψ) be an arbitrary instance of $\mathrm{Phylo}(\Phi)$. The corresponding instance ψ of $\mathrm{CSP}(\Gamma_\Phi)$ is defined as the conjunction of all formulas $R_\varphi(x_{i_1}, \dots, x_{i_l})$, where Ψ contains the formula $\varphi(x_{i_1}, \dots, x_{i_l})$. Now it holds that

$$
\begin{aligned}
& (V, \Psi) \text{ is a positive instance of } \mathrm{Phylo}(\Phi) \\
\iff\ & \text{there is a tree } T \text{ and an assignment } s \colon V \to L(T) \text{ such that} \\
& (L(T); C) \models \bigwedge \Psi \text{ under the assignment } s \\
\iff\ & \text{there is a tree } T \text{ and an assignment } s \colon V \to L(T) \text{ such that} \\
& (L(T); (R_\varphi^{(L;C)})_{\varphi\in\Phi}) \models \psi \text{ under the assignment } s \\
\iff\ & \text{there is a substructure } (L; (R_\varphi)_{\varphi\in\Phi}) \text{ of } \Gamma_\Phi \text{ and an assignment } s \colon V \to L \text{ such that} \\
& (L; (R_\varphi)_{\varphi\in\Phi}) \models \psi \text{ under the assignment } s \\
\iff\ & \psi \text{ is a positive instance of } \mathrm{CSP}(\Gamma_\Phi).
\end{aligned}
$$

(2) By Proposition 3.7, all relations R_i of $(\mathbb{L}; R_1, \dots, R_n)$ have a quantifier-free definition φ_i over $(\mathbb{L}; C)$. Define $\Phi := \{\varphi_1, \dots, \varphi_n\}$. Let ψ be an arbitrary instance of $\mathrm{CSP}(\Gamma)$; we can reduce it to the following instance (V, Ψ) of $\mathrm{Phylo}(\Phi)$: V is the set of variables occurring in ψ and, for every conjunct $R_i(x_{i_1}, \dots, x_{i_l})$ of ψ, Ψ contains the formula $\varphi_i(x_{i_1}, \dots, x_{i_l})$. The correctness of this reduction is immediate. □

[†] Remarkably, $(\mathbb{L}; \prec)$ is a dense linear order without endpoints, so it is isomorphic to $(\mathbb{Q}; <)$ by Cantor's isomorphism theorem.

We will later need the following lemma:

3.9 Lemma. *Let L_1, L_2 be two finite subsets of $\mathbb{L}$. Then there is some $\alpha \in \mathrm{Aut}(\mathbb{L}; C)$ such that $L_1 | \alpha(L_2)$.*

Proof. Note that L_1 and L_2 do not have to be disjoint. Consider the finite tree T which has L_1 as the leaf structure of its left subtree and a disjoint copy L_2' of L_2 as the leaf structure of its right subtree. $(L(T); C)$ embeds into $(\mathbb{L}; C)$ via some embedding β. $\beta^{-1}\big|_{\beta(L_1)}$ is a partial isomorphism of $(\mathbb{L}; C)$, so it can be extended to some $\gamma \in \mathrm{Aut}(\mathbb{L}; C)$ by the homogeneity of $(\mathbb{L}; C)$. Let $\alpha' \colon L_2 \to L_2'$ be an isomorphism. $\gamma \circ \beta \circ \alpha' \colon L_2 \to \gamma(\beta(L_2'))$ is a partial isomorphism of $(\mathbb{L}; C)$, so it can be extended to some $\alpha \in \mathrm{Aut}(\mathbb{L}; C)$. Now $L_1 | \alpha(L_2)$, as required. □

3.3 Reducts of the C-Relation and Phylogeny CSPs in Datalog

In this section, we will study the following important reducts of $(\mathbb{L}; C)$:

$$
\begin{aligned}
C_d &:= \{(x,y,z) \in \mathbb{L}^3 \colon x | yz \wedge y \neq z\},\\
Q &:= \{(x,y,u,v) \in \mathbb{L}^4 \colon (xy|u \wedge xy|v) \vee (x|uv \wedge y|uv)\},\\
Q_d &:= \{(x,y,u,v) \in \mathbb{L}^4 \colon Q(x,y,u,v) \wedge x \neq y \wedge u \neq v\},\\
N &:= \{(x,y,z) \in \mathbb{L}^3 \colon x|yz \vee xy|z\},\\
N_d &:= \{(x,y,z) \in \mathbb{L}^3 \colon N(x,y,z) \wedge x \neq y \wedge y \neq z\}.
\end{aligned}
$$

By examining reducts of $(\mathbb{L}; C)$ which do not pp-define C, we will obtain a dichotomy for phylogeny CSPs regarding their expressibility in Datalog.

3.10 Lemma ([1, Lemma 3.1])**.** *$\langle(\mathbb{L}; C)\rangle = \langle(\mathbb{L}; C_d)\rangle, \langle(\mathbb{L}; Q)\rangle = \langle(\mathbb{L}; Q_d)\rangle$ and $\langle(\mathbb{L}; N)\rangle = \langle(\mathbb{L}; N_d)\rangle$.*

Proof. $\exists u\,(C(x,y,u)), \exists u, v\,(Q(u,x,v,y))$ and $\exists u\,(N(x,y,u))$ are pp-formulas equivalent to $x \neq y$, thus $C_d \in \langle(\mathbb{L}; C)\rangle, Q_d \in \langle(\mathbb{L}; Q)\rangle$ and $N_d \in \langle(\mathbb{L}; N)\rangle$. For the inverse inclusions, it is not difficult to show that

$$
\begin{aligned}
C(x,y,z) &\overset{(\dagger)}{\iff} \exists u\ (C_d(x,y,u) \wedge C_d(x,z,u)),\\
Q(x,y,z,t) &\iff \exists u, v\ (Q_d(u,x,v,z) \wedge Q_d(u,x,v,t) \wedge Q_d(u,y,v,z) \wedge Q_d(u,y,v,t)) \text{ and}\\
N(x,y,z) &\iff \exists u, v\ (N_d(v,x,u) \wedge N_d(v,u,x) \wedge N_d(u,v,y) \wedge N_d(u,y,v) \wedge N_d(u,z,v)).
\end{aligned}
$$

We exemplarily give a proof of (†). »$\overset{(\dagger)}{\Leftarrow}$« is a direct consequence of Lemma 3.3 (5) and Lemma 2.19. For »$\overset{(\dagger)}{\Rightarrow}$«, let $x, y, z \in \mathbb{L}$ such that $C(x,y,z)$. Because $(\mathbb{L}; C)$ is the Fraïssé limit of the class of all leaf structures of finite binary rooted trees, we can assume that the leaf structures of the trees depicted in Figure 4 are substructures of $(\mathbb{L}; C)$.

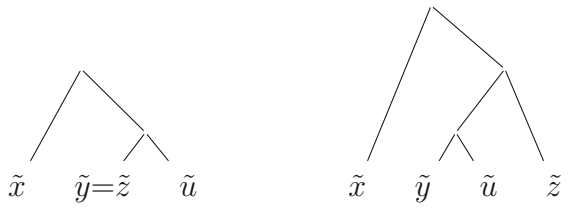


Figure 4: Two binary rooted trees, whose leaf structures can be assumed to be substructures of $(\mathbb{L}; C)$.

We distinguish two cases: If $y = z$, consider the tree on the left; if $y \neq z$, consider the right tree. In both cases, the map β which is defined by $\beta(\tilde{x}) = x, \beta(\tilde{y}) = y$ and $\beta(\tilde{z}) = z$ is a partial isomorphism between $(\mathbb{L}; C)[\{x, y, z\}]$ and the leaf structure of the respective tree. By the homogeneity of $(\mathbb{L}; C)$, β can be extended to some $\alpha \in \mathrm{Aut}(\mathbb{L}; C)$. Let $u \coloneqq \alpha(\tilde{u})$. The leaf structures of the depicted trees fulfill $C_d(\tilde{x}, \tilde{y}, \tilde{u}) \wedge C_d(\tilde{x}, \tilde{z}, \tilde{u})$, so $(\mathbb{L}; C) \models C_d(\tilde{x}, \tilde{y}, \tilde{u}) \wedge C_d(\tilde{x}, \tilde{z}, \tilde{u})$. Since $\alpha \in \mathrm{Aut}(\mathbb{L}; C)$, we obtain $(\mathbb{L}; C) \models C_d(\alpha(\tilde{x}), \alpha(\tilde{y}), \alpha(\tilde{u})) \wedge C_d(\alpha(\tilde{x}), \alpha(\tilde{z}), \alpha(\tilde{u}))$, i. e. $(\mathbb{L}; C) \models C_d(x, y, u) \wedge C_d(x, z, u)$. □

3.11 Lemma ([1, Lemma 3.7])**.** *$(\mathbb{L}; C)$ and $(\mathbb{L}; Q)$ are ω-categorical, model-complete cores.*

Proof. Let $f \in \mathrm{End}(\mathbb{L}; C)$ be arbitrary; we first show that $f \in \mathrm{Emb}(\mathbb{L}; C)$, i. e. that f is injective and preserves all relations strongly. If $u, v \in \mathbb{L}$ are distinct, then $uu|v$ and consequently $f(u)f(u)|f(v)$ hold; the latter implies $f(u) \neq f(v)$. The formula $\neg(x|yz)$ is equivalent to the existential positive formula $(x = y = z) \vee y|xz \vee z|xy$, hence f preserves C strongly. Model-completeness follows via Lemma 2.12, as $(\mathbb{L}; C)$ is ω-categorical and homogeneous.

Now let $g \in \mathrm{End}(\mathbb{L}; Q)$. From $u \neq v \iff Q(u, u, v, v)$ we conclude that g is injective. For g being an embedding, it suffices that $\neg Q(x, y, z, t)$ implies $\neg Q(g(x), g(y), g(z), g(t))$. We have

$$\begin{aligned}
\neg Q(x, y, z, t) &\iff \neg\big((xy|z \;\wedge\; xy|t) \;\vee\; (x|zt \;\wedge\; y|zt)\big) \\
&\iff (\neg xy|z \;\vee\; \neg xy|t) \;\wedge\; (\neg x|zt \;\vee\; \neg y|zt) \\
&\iff \big((x = y = z) \;\vee\; x|yz \;\vee\; y|xz \;\vee\; (x = y = t) \;\vee\; x|yt \;\vee\; y|xt\big) \wedge \\
&\qquad \big((x = z = t) \;\vee\; t|xz \;\vee\; z|xt \;\vee\; (y = z = t) \;\vee\; z|yt \;\vee\; t|yz\big).
\end{aligned}$$

If one of the four equality disjuncts holds true, we are done: E. g., if $x = y = z$, then $g(x) = g(y) = g(z)$, which implies $\neg Q(g(x), g(y), g(z), g(t))$. Otherwise, one of the four remaining disjuncts in the first conjunct and one of the four remaining disjuncts in the second conjunct must be true. If e. g. $x|yz$ and $t|xz$ are satisfied, we can infer $t|yz$ and, subsequently, $Q(x, t, y, z)$. Thus, $Q(g(x), g(t), g(y), g(z))$, which implies $\neg Q(g(x), g(y), g(z), g(t))$. The other cases can be treated similarly. The ω-categoricity of $(\mathbb{L}; Q)$ follows from Corollary 2.5. The proof of the homogeneity of $(\mathbb{L}; Q)$ is a bit subtle, it can be found in [5, Lemma 14]. Its model-completeness follows again from Lemma 2.12. □

3.12 Lemma. *$(\mathbb{L}; N)$ is an ω-categorical, model-complete core.*

Proof. Follows immediately from Lemma 2.15 and Lemma 3.11, since $C(x, y, z) \iff N(x, y, z) \wedge N(x, z, y)$. □

3.13 Proposition ([1, Proposition 8.3])**.** *Let a, b be two distinct elements from $\mathbb{L}$. Then $(\mathbb{L}; N, a, b)$ pp-interprets $(\{0, 1\}; \mathrm{NAE})$.*

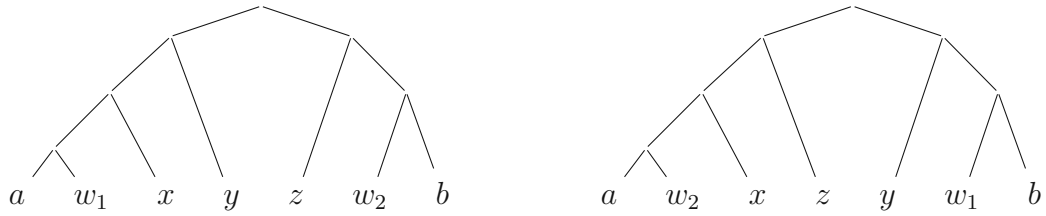
Proof. In the notation of Definition 2.23, the interpretation is as follows: The dimension is $d = 1$, the subset S is given by $S = \{x \in \mathbb{L} \colon N_d(a, x, b)\}$, ϑ is the equivalence relation on S with the two classes $S_0 = \{x \in S \colon ax|b\}$ and $S_1 = \{x \in S \colon a|xb\}$ and f is the function that maps x to 0 if $x \in S_0$ and to 1 if $x \in S_1$. Since $N_d \in \langle(\mathbb{L}; N)\rangle$ by Lemma 3.10 and $C \in \langle(\mathbb{L}; N)\rangle$, we observe that S and ϑ are pp-definable in $(\mathbb{L}; N)$. It remains to show

that $f^{-1}(\mathrm{NAE}) = \{(x,y,z) \in \mathbb{L}^3 \mid (f(x),f(y),f(z)) \in \mathrm{NAE}\}$ is pp-definable in $(\mathbb{L};N)$. We claim that $(f(x),f(y),f(z)) \in \mathrm{NAE} \iff (\mathbb{L};N) \models \varphi(x,y,z)$, where

$$\varphi(x,y,z) \equiv \exists w_1,w_2\,(N_d(x,w_1,y) \wedge N_d(w_1,w_2,z) \wedge N_d(w_1,a,w_2) \wedge N_d(w_1,b,w_2)).$$

For »⇐«, assume towards contradiction that $\varphi(x,y,z)$ holds, but $(f(x),f(y),f(z)) \notin \mathrm{NAE}$, without loss of generality $f(x) = f(y) = f(z) = 0$. Then there is some w_1 such that either $x|w_1y$ or $xw_1|y$ and some w_2 such that either $w_1|w_2z$ or $w_1w_2|z$. By the definition of f, we have $ax|b, ay|b$ and $az|b$, so in any case $axyzw_1w_2|b$. But, because of the last conjunct in φ, also $w_1|bw_2$ or $w_1b|w_2$ holds, contradiction.

For »⇒«, first consider the case that $f(x) = f(y) = 0$ and $f(z) = 1$, i. e. $ax|b, ay|b$ and $a|zb$. If $ax|y$, then w_1 and w_2 can be choosen according to the leaf structure of the tree in Figure 5 (a); if $ay|x$, the same tree with x and y swapped works; if $a|xy$, then a and y must be swapped. Next, assume that $f(x) = f(z) = 0$ and $f(y) = 1$. If $ax|z$, then w_1 and w_2 can be choosen as in Figure 5 (b); if $az|x$, swap x and z; if $a|xz$, swap a and z.



(a) (b)

Figure 5.

The cases considered are exhaustive since φ is symmetric in a and b as well as in x and y. □

With a bit more work, one can also show the following:

3.14 Proposition ([1, Proposition 8.4])**.** *Let a,b,c be three distinct elements from $\mathbb{L}$. Then $(\mathbb{L};Q,a,b,c)$ pp-interprets $(\{0,1\};\mathrm{NAE})$.*

3.15 Corollary. *$(\mathbb{L};N)$ and $(\mathbb{L};Q)$ pp-construct $(\{0,1\};\mathrm{NAE})$.*

Proof. For arbitrary distinct $a,b \in \mathbb{L}$, we have $(\{0,1\};\mathrm{NAE}) \in \mathbf{I}(\mathbb{L};N,a,b)$ by Proposition 3.13; hence, $(\{0,1\};\mathrm{NAE}) \in \mathbf{HP}(\mathbb{L};N,a,b)$ by Proposition 2.25 (c). By Lemma 3.12, $(\mathbb{L};N)$ is an ω-categorical, model-complete core, thus $(\mathbb{L};N,a,b) \in \mathbf{HP}(\mathbb{L};N)$ by Lemma 2.27. Putting this together, we obtain $(\{0,1\};\mathrm{NAE}) \in \mathbf{HPHP}(\mathbb{L};N)$. The claim follows for $(\mathbb{L};N)$ since $\mathbf{HPHP}(\mathbb{L};N) = \mathbf{HP}(\mathbb{L};N)$ again by Proposition 2.25. For $(\mathbb{L};Q)$, we can argue in a similar way. □

3.16 Theorem ([5])**.** *Let Γ be a reduct of $(\mathbb{L};C)$. Then one of the following applies:*

(1) $\mathrm{End}(\Gamma)$ *contains a constant operation.*

(2) *The model-complete core of Γ is isomorphic to a reduct of $(\mathbb{L};=)$.*

(3) $\mathrm{End}(\Gamma) = \mathrm{End}(\mathbb{L};Q)$.

(4) $\mathrm{End}(\Gamma) = \mathrm{End}(\mathbb{L};C)$.

3.17 Lemma ([1, Lemma 4.2])**.** *Let* Γ *be a reduct of* $(\mathbb{L}; C)$*. Then one of the following applies:*

(1) $\mathrm{End}(\Gamma)$ *contains a constant operation.*

(2) *The model-complete core of* Γ *is isomorphic to a reduct of* $(\mathbb{L}; =)$*.*

(3) Γ *is a model-complete core, and* $C \in \langle\Gamma\rangle$ *or* $Q \in \langle\Gamma\rangle$*.*

Proof. Observe that C_d is contained in one orbit of 3-tuples of $\mathrm{Aut}(\Gamma)$: For $x, y \in C_d$, the map $x_i \mapsto y_i$, $i \in \{1, 2, 3\}$, is a partial isomorphism of $(\mathbb{L}; C)$ and can thus be extended to an automorphism of $(\mathbb{L}; C)$ by homogeneity. Since $\mathrm{Aut}(\mathbb{L}; C) \subseteq \mathrm{Aut}(\Gamma)$, x and y are in particular in the same orbit with respect to $\mathrm{Aut}(\Gamma)$.

We distinguish two cases. First, suppose that $C \in \langle\Gamma\rangle$. Then Γ is a model-complete core by Lemma 2.15 and Lemma 3.11, and we are done. If $C \notin \langle\Gamma\rangle$, however, then, by Lemma 3.10, $C_d \notin \langle\Gamma\rangle$. By Theorem 2.6, there is some $g \in \mathrm{Pol}(\Gamma)$ that violates C_d. Because C_d is contained in one orbit of 3-tuples of $\mathrm{Aut}(\Gamma)$, there is some $f \in \mathrm{End}(\Gamma)$ that violates C_d by Lemma 2.2. f also violates C, again by Lemma 3.10; thus, $\mathrm{End}(\Gamma) \neq \mathrm{End}(\mathbb{L}; C)$. Assume that (1) and (2) do not hold; Theorem 3.16 implies that $\mathrm{End}(\Gamma) = \mathrm{End}(\mathbb{L}; Q)$.

Let $x, y \in Q_d$, then the map $x_i \mapsto y_i, i \in \{1, 2, 3, 4\}$, is a partial isomorphism of $(\mathbb{L}; Q)$; it can be extended to some $\alpha \in \mathrm{Aut}(\mathbb{L}; Q)$ by the homogeneity of $(\mathbb{L}; Q)$ (see [5, Lemma 14]). Since $\mathrm{End}(\Gamma) = \mathrm{End}(\mathbb{L}; Q)$, in particular $\mathrm{Aut}(\Gamma) = \mathrm{Aut}(\mathbb{L}; Q)$, we get $\alpha \in \mathrm{Aut}(\Gamma)$. Thus, Q_d is contained in one orbit of 4-tuples of $\mathrm{Aut}(\Gamma)$.

Now assume towards contradiction that $Q \notin \langle\Gamma\rangle$. Then $Q_d \notin \langle\Gamma\rangle$ by Lemma 3.10, hence there is some $g \in \mathrm{Pol}(\Gamma)$ that violates Q_d. Because Q_d is contained in one orbit of 4-tuples of $\mathrm{Aut}(\Gamma)$, there is some $f \in \mathrm{End}(\Gamma)$ that violates Q_d, in contradiction to $\mathrm{End}(\Gamma) = \mathrm{End}(\mathbb{L}; Q)$. Hence, $Q \in \langle\Gamma\rangle$. It follows from $\mathrm{Aut}(\Gamma) = \mathrm{Aut}(\mathbb{L}; Q)$ and Lemma 2.15 that Γ is a model-complete core. □

Using another known result, we obtain a full dichotomy for phylogeny CSPs in Datalog:

3.18 Theorem. *Let* Γ *be a reduct of* $(\mathbb{L}; C)$ *with finite signature. Then exactly one of the following holds:*

(1) $C \in \langle\Gamma\rangle$ *or* $(\{0, 1\}; \mathrm{NAE}) \in \mathbf{HP}(\Gamma)$ *(and* $\mathrm{CSP}(\Gamma)$ *is inexpressible in* Datalog*).*

(2) $\mathrm{CSP}(\Gamma)$ *is expressible in* Datalog*.*

Proof. If Γ pp-constructs $(\{0, 1\}; \mathrm{NAE})$, then $\mathrm{CSP}(\Gamma)$ is inexpressible in FPC by Corollary 2.35; in particular, it is inexpressible in Datalog, since Datalog is a fragment of FPC. $\mathrm{CSP}(\mathbb{L}; C)$ is inexpressible in Datalog by [18, Theorem 8.6.10]. Hence, each structure that pp-defines C is inexpressible in Datalog by Theorem 2.33.

So assume that (1) does not hold; we have to show that $\mathrm{CSP}(\Gamma)$ is expressible in Datalog. We make a case distinction over the three cases of Lemma 3.17. If Γ has a constant endomorphism, then $\mathrm{CSP}(\Gamma)$ is trivially expressible in Datalog since it has finite signature. If the model-complete core Δ of Γ is isomorphic to a reduct of $(\mathbb{L}; =)$, then, by Theorem 2.46, either $\mathrm{CSP}(\Gamma) = \mathrm{CSP}(\Delta)$ is expressible in Datalog, or $\mathbb{K}_3 \in \mathbf{HP}(\Delta)$. Since $(\{0, 1\}; \mathrm{NAE}) \in \mathbf{HP}(\mathbb{K}_3)$ and $\Delta \in \mathbf{H}(\Gamma)$, the latter case would imply $(\{0, 1\}; \mathrm{NAE}) \in \mathbf{HPHPH}(\Gamma) = \mathbf{HP}(\Gamma)$ by Lemma 2.25, contradiction. Otherwise, since $C \notin \langle\Gamma\rangle$ by assumption, Lemma 3.17 implies that $(\mathbb{L}; Q) \in \mathbf{D}(\Gamma)$. Since $(\{0, 1\}; \mathrm{NAE}) \in \mathbf{HP}(\mathbb{L}; Q)$ by Corollary 3.15, we obtain $(\{0, 1\}; \mathrm{NAE}) \in \mathbf{HPD}(\Gamma) = \mathbf{HP}(\Gamma)$, which is a contradiction as well. □

4 A Tractable Phylogeny Language not in FPC

This section is about the relation

$$J \coloneqq \{(x_1, x_2, x_3, x_4) \in \mathbb{L}^4 \colon \left(\bigwedge_{1 \le i < j \le 4} x_i \neq x_j \right) \wedge (x_1 x_2 | x_3 x_4 \vee x_1 x_3 | x_2 x_4 \vee x_1 x_4 | x_2 x_3)\}.$$

$J(x_1, x_2, x_3, x_4)$ asserts that two elements out of $\{x_1, x_2, x_3, x_4\}$ lie on the right side and two lie on the left side of their youngest common ancestor, and that they are all pairwise distinct. We will show that $\mathrm{CSP}(\mathbb{L}; J)$ is not expressible in FPC, although there is a simple polynomial-time algorithm for it. For auxiliary purposes, we will also use the relation

$$\widetilde{J} \coloneqq \{(x_1, x_2, x_3, x_4, x_5, x_6) \in \mathbb{L}^6 \colon \left(\bigwedge_{1 \le i < j \le 6} x_i \neq x_j \right) \wedge \exists h (J(x_1, x_2, x_3, h) \wedge J(h, x_4, x_5, x_6))\}.$$

4.1 Lemma. *J and $\widetilde{J}$ are characterized by* even splits*, i. e.*

(1) *For any injective tuple $x \in \mathbb{L}^4$, $x \in J \iff \exists \pi \in \mathrm{Sym}(4) \colon x_{\pi(1)} x_{\pi(2)} | x_{\pi(3)} x_{\pi(4)}$.*

(2) *For any injective tuple $x \in \mathbb{L}^6$, $x \in \widetilde{J} \iff \exists \pi \in \mathrm{Sym}(6) \colon x_{\pi(1)} x_{\pi(2)} | x_{\pi(3)} x_{\pi(4)} x_{\pi(5)} x_{\pi(6)}$.*

Proof. (1) follows directly from the definition of J. For »$\overset{(2)}{\Rightarrow}$«, first assume towards contradiction that $x_1 | x_2 x_3 x_4 x_5 x_6$ holds. By assumption, there is some $h \in \mathbb{L}$ such that $J(x_1, x_2, x_3, h)$ and $J(h, x_4, x_5, x_6)$. Now, on the one hand, $x_1 h | x_2 x_3$ must hold, hence $x_1 h | x_2 x_3 x_4 x_5 x_6$ and subsequently $h | x_4 x_5 x_6$, contradiction. Next, assume that $x_1 x_2 x_3 | x_4 x_5 x_6$; without loss of generality, $x_1 x_2 | x_3 h$ and $h x_4 | x_5 x_6$ shall be true. But this implies $x_1 x_2 x_3 h | x_4 x_5 x_6$ as well as $x_1 x_2 x_3 | h x_4 x_5 x_6$, which contradict each other. Finally, assume that $x_1 x_2 x_4 | x_3 x_5 x_6$. It follows that $x_1 x_2 | x_3 h$; thus, $x_4 | h x_5 x_6$, which is a contradiction as well. The case distinction is exhaustive since J is totally symmetric. For »$\overset{(2)}{\Leftarrow}$«, we only have to consider two cases, again due to the symmetry of J. First, suppose that $x_1 x_2 | x_3 x_4 x_5 x_6$. Without loss of generality, let $x_4 | x_5 x_6$. Due to the homogeneity of $(\mathbb{L}; C)$, there exists some $h \in \mathbb{L}$ such that $h x_4 | x_5 x_6$. Clearly, this implies $x_1 x_2 | x_3 h$. Secondly, consider the case that $x_1 x_4 | x_2 x_3 x_4 x_5$. The homogeneity of $(\mathbb{L}; C)$ provides some $h \in \mathbb{L}$ such that $x_1 x_4 h | x_2 x_3 x_4 x_5 x_6$. Hence, $x_1 h | x_2 x_3$ and $x_4 h | x_5 x_6$. □

Satisfiability of instances of tractable phylogeny CSPs is closely related to the solvability of linear equation systems over $\mathbb{Z}_2$. Intuitively, for some elements $x_1, \ldots, x_n \in \mathbb{L}$, we can assign 0 or 1 to each of them, depending on whether they are contained in the left or the right subtree of the underlying tree of the leaf structure $(\mathbb{L}; C)[x_1, \ldots, x_n]$. In the specific case of J, the following two computational problems are of our interest:

(2, 3, 4, 5)-Ord-Xor-Sat
INSTANCE: A finite homogeneous system of linear equations of length $l \in \{2, 3, 4, 5\}$ over $\mathbb{Z}_2$.
QUESTION: Does every nonempty subset of the equations have a solution where at least one variable occurring in this subset has value 1?

(4, 6)-Phylo-Xor-Sat
INSTANCE: A finite homogeneous system of linear equations of length $l \in \{4, 6\}$ over $\mathbb{Z}_2$.
QUESTION: Does every nonempty subset of the equations have a solution where at least one variable occurring in this subset has value 0 and at least one variable occurring in this subset has value 1?

4.2 Definition. For an instance $\Gamma = (V; J, \widetilde{J})$ of CSP($\mathbb{L}; J, \widetilde{J}$), let $\mathcal{A}(\Gamma)$ be the instance of $(4, 6)$-Phylo-Xor-Sat with domain V that contains the equation $x_1 + x_2 + x_3 + x_4 = 0$ for each constraint of the form $J^\Gamma(x_1, x_2, x_3, x_4)$ and the equation $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 0$ for each constraint of the form $\widetilde{J}^\Gamma(x_1, x_2, x_3, x_4, x_5, x_6)$.

Since $\mathcal{A}(\Gamma)$ and Γ have the same domain, we will view functions on $\mathcal{A}(\Gamma)$ also as functions on Γ in the following.

4.3 Lemma ([3, proof of Theorem 4.23])**.** *For every $k \geq 3$, there are instances $\mathcal{A}'_1$ and $\mathcal{A}'_2$ of $(2, 3, 4, 5)$-Ord-Xor-Sat such that $\mathcal{A}'_1$ is a negative instance, $\mathcal{A}'_2$ is a positive instance and $\mathcal{A}'_1 \equiv_{\mathcal{C}^k} \mathcal{A}'_2$.*

4.4 Lemma. *For every $k \geq 3$, there are instances $\mathcal{A}_1$ and $\mathcal{A}_2$ of $(4, 6)$-Phylo-Xor-Sat such that $\mathcal{A}_1$ is a negative instance, $\mathcal{A}_2$ is a positive instance and $\mathcal{A}_1 \equiv_{\mathcal{C}^k} \mathcal{A}_2$.*

Proof. Let $\mathcal{A}'_1$ and $\mathcal{A}'_2$ be the instances of $(2, 3, 4, 5)$-Ord-Xor-Sat provided by Lemma 4.3. $\mathcal{A}_1$ and $\mathcal{A}_2$ are obtained from $\mathcal{A}'_1$ and $\mathcal{A}'_2$, respectively, by introducing a fresh variable z and performing the following transformation:

$$\begin{aligned}
x_1 + x_2 &= 0 &&\rightsquigarrow & x_1 + x_2 + z + z &= 0,\\
x_1 + x_2 + x_3 &= 0 &&\rightsquigarrow & x_1 + x_2 + x_3 + z &= 0,\\
x_1 + x_2 + x_3 + x_4 &= 0 &&\rightsquigarrow & x_1 + x_2 + x_3 + x_4 + z + z &= 0,\\
x_1 + x_2 + x_3 + x_4 + x_5 &= 0 &&\rightsquigarrow & x_1 + x_2 + x_3 + x_4 + x_5 + z &= 0.
\end{aligned}$$

We have to show three things:

- $\mathcal{A}_1$ is a negative instance of $(4, 6)$-Phylo-Xor-Sat: Let F' be a subset of $\mathcal{A}'_1$ that only admits the all-zero solution and let F be the corresponding subset of $\mathcal{A}_1$ obtained from F' by the transformation above. Assume towards contradiction that F has a solution s taking values from both 0 and 1. If $s(z) = 0$, then the restriction of s to the variables without z is a solution to F' where at least one variable has value 1, contradiction. If $s(z) = 1$, then $\widetilde{s}(x) \coloneqq s(x) + 1$ is a solution of F taking values from both 0 and 1 with $\widetilde{s}(z) = 0$, which yields a contradiction as in the former case.

- $\mathcal{A}_2$ is a positive instance of $(4, 6)$-Phylo-Xor-Sat: Let F be some nonempty subset of $\mathcal{A}_2$. The corresponding subset F' of $\mathcal{A}'_2$ has a solution where at least one variable is mapped to 1. Extend it by sending z to 0 to obtain a solution of F attaining both 0 and 1 (note that z occurs in F since it occurs in every equation).

- $\mathcal{A}_1 \equiv_{\mathcal{C}^k} \mathcal{A}_2$: As we have stated in Section 2.6, $\equiv_{\mathcal{C}^k}$ is characterized by the bijective k-pebble game. Since $\mathcal{A}'_1 \equiv_{\mathcal{C}^k} \mathcal{A}'_2$, we know that the Duplicator has a winning strategy in the game played on $\mathcal{A}'_1$ and $\mathcal{A}'_2$. In the game played on $\mathcal{A}_1$ and $\mathcal{A}_2$, he

can choose the same strategy extended by always mapping z to z. We prove this for those for whom this is not obvious: Let α be the partial map induced by the pebbles after a certain round. The restriction α' of α to those pebbles which do not lie on z is the same map that would have been on the board if the game would have been played on $\mathcal{A}_1'$ and $\mathcal{A}_2'$ with the rounds removed in which the Spoiler chose z. By assumption, α' is a partial isomorphism between $\mathcal{A}_1'$ and $\mathcal{A}_2'$. Now consider for example an equation of $\mathcal{A}_1$ of the form $x_1 + x_2 + z + z = 0$ where x_1, x_2 and z are in the domain of α. It holds that

$$\begin{aligned}
\mathcal{A}_1 \models x_1 + x_2 + z + z = 0 &\iff \mathcal{A}_1' \models x_1 + x_2 = 0 \\
&\iff \mathcal{A}_2' \models \alpha'(x_1) + \alpha'(x_2) = 0 \\
&\iff \mathcal{A}_2 \models \alpha'(x_1) + \alpha'(x_2) + z + z = 0 \\
&\iff \mathcal{A}_2 \models \alpha(x_1) + \alpha(x_2) + \alpha(z) + \alpha(z) = 0,
\end{aligned}$$

so α strongly preserves the equation. The other types of equations can be treated similiarly. □

The following is a divide-and-conquer algorithm for $\mathrm{CSP}(\mathbb{L}; J, \widetilde{J})$. It clearly has polynomial runtime, because checking whether a system of linear equations over $\mathbb{Z}_2$ has a solution taking values from both 0 and 1 can be done via Gaußian elimination. The idea of solving CSPs of phylogeny languages with a divide-and-conquer approach is from [1, § 6.4].

Algorithm SolveJ

Input: An instance Γ of $\mathrm{CSP}(\mathbb{L}; J, \widetilde{J})$
Output: $\top$ or $\bot$

```
if A(Γ) = ∅:
    return ⊤
else:
    if there is no solution of A(Γ) taking values from both 0 and 1:
        return ⊥
    else:
        s ← a solution of A(Γ) taking values from both 0 and 1
        if SolveJ(Γ[s^{-1}(0)]) = ⊥:
            return ⊥
        else if SolveJ(Γ[s^{-1}(1)]) = ⊥:
            return ⊥
        else:
            return ⊤
```

Figure 6. A polynomial-time divide-and-conquer algorithm for $\mathrm{CSP}(\mathbb{L}; J, \widetilde{J})$.

4.5 Proposition. *The algorithm in Figure 6 is sound and complete for* $\mathrm{CSP}(\mathbb{L}; J, \widetilde{J})$*, i. e.*

$$\Gamma \to (\mathbb{L}; J, \widetilde{J}) \iff \textsc{SolveJ}(\Gamma) = \top.$$

Proof. We proceed by induction over the recursion levels of SolveJ. If J^Γ and $\widetilde{J}^\Gamma$ are empty, the statement is trivially true. So assume for the rest of the proof that Γ contains at least one constraint.

»⇒«: Let $t\colon \Gamma \to \mathbb{L}$ be a solution to Γ. By the definition of J and $\widetilde{J}$, $t(\Gamma)$ must contain at least four, in particular at least two distinct elements of $\mathbb{L}$. Hence, there is a partition of $t(\Gamma)$ into two nonempty sets X and Y such that $X|Y$. Define a function s on Γ by $s(x) = 0$ if $t(x) \in X$ and $s(x) = 1$ if $t(x) \in Y$. For each constraint $J(x_1, \dots, x_4)$ or $\widetilde{J}(x_1, \dots, x_6)$, an even number of its variables must be contained in $t^{-1}(X)$ and an even number must be contained in $t^{-1}(Y)$ by Lemma 4.1. Hence, s is a solution to $\mathcal{A}(\Gamma)$. It attains both the values 0 and 1 since X and Y are nonempty. Taking the restriction of t, we obtain a solution to the subproblems $\Gamma[s^{-1}(0)]$ and $\Gamma[s^{-1}(1)]$. Thus, $\textsc{SolveJ}(\Gamma[s^{-1}(0)]) = \top$ and $\textsc{SolveJ}(\Gamma[s^{-1}(1)]) = \top$ by the induction hypothesis; hence, overall, $\textsc{SolveJ}(\Gamma) = \top$.

»⇐«: We will inductively show that Γ has an *injective* solution. By assumption, there is a solution s to $\mathcal{A}(\Gamma)$ that attains both the values 0 and 1. Let $S_0 \coloneqq s^{-1}(0)$ and $S_1 \coloneqq s^{-1}(1)$. By induction hypothesis, $\Gamma[S_0] \to (\mathbb{L}; J, \widetilde{J})$ and $\Gamma[S_1] \to (\mathbb{L}; J, \widetilde{J})$ via injective functions $t_0\colon S_0 \to \mathbb{L}$ and $t_1\colon S_1 \to \mathbb{L}$. By Lemma 3.9, there is some $\alpha \in \mathrm{Aut}(\mathbb{L}; C)$ such that $t_0(S_0)|\alpha(t_1(S_1))$. We will show that the function $t\colon \Gamma \to \mathbb{L}$ defined by

$$t(x) = \begin{cases} t_0(x), & x \in S_0 \\ \alpha(t_1(x)), & x \in S_1 \end{cases}$$

is an injective solution to Γ.

The injectivity is clear since α is an automorphism. For some constraint $J(x_1, \dots, x_4)$ or $\widetilde{J}(x_1, \dots, x_6)$ of Γ, we distinguish two cases: If it contains only variables from S_0 or only variables from S_1, it is preserved by t by the induction hypothesis. If the constraint however contains variables from both S_0 and S_1, then an even number of them are contained in S_0 and S_1, respectively, since s is a solution to $\mathcal{A}(\Gamma)$. Hence, an even number of their images under t is contained in $t(S_0)$ and $t(S_1)$, respectively. Since $t(S_0)|t(S_1)$, t preserves the constraint in this case as well due to Lemma 4.1. □

4.6 Proposition. *Let* Γ *be an instance of* $\mathrm{CSP}(\mathbb{L}; J, \widetilde{J})$*. Then*

$$\Gamma \to (\mathbb{L}; J, \widetilde{J}) \iff \mathcal{A}(\Gamma) \textit{ is a positive instance of } (4,6)\textit{-Phylo-Xor-Sat.}$$

Proof. »⇒«: Let F be some nonempty subset of $\mathcal{A}(\Gamma)$. By Proposition 4.5, $\textsc{SolveJ}(\Gamma)$ returns $\top$ in every step. In some subprocedure, F is fully contained in the set of equations for the last time. Let s be a solution of the equations in this subprocedure attaining both 0 and 1. Since the variables of F are neither fully contained in $s^{-1}(0)$ nor in $s^{-1}(1)$, s is a suitable solution to F.

»⇐«: If every nonempty subset of $\mathcal{A}(\Gamma)$ has a solution attaining both 0 and 1, then $\textsc{SolveJ}(\Gamma)$ returns $\top$ in every step. Hence, $\Gamma \to (\mathbb{L}; J, \widetilde{J})$ by Proposition 4.5. □

4.7 Theorem. $\mathrm{CSP}(\mathbb{L}; J, \widetilde{J})$ *is inexpressible in* FPC.

Proof. Assume towards contradiction that there is an FPC sentence φ such that $\Gamma \models \varphi \iff \Gamma \not\to (\mathbb{L}; J, \widetilde{J})$ for all finite $\{J, \widetilde{J}\}$-structures Γ. By Theorem 2.38, there is some $k \in \mathbb{N}$ such that $\Gamma_1 \equiv_{\mathcal{C}^k} \Gamma_2$ implies $\Gamma_1 \models \varphi \iff \Gamma_2 \models \varphi$ for all finite $\{J, \widetilde{J}\}$-structures Γ_1 and Γ_2.

For k as above, there are, due to Lemma 4.4, instances $\mathcal{A}_1$ and $\mathcal{A}_2$ of $(4, 6)$-Phylo-Xor-Sat such that the former is a negative instance, the latter is a positive instance and $\mathcal{A}_1 \equiv_{\mathcal{C}^k} \mathcal{A}_2$. Let Γ_1 and Γ_2 be the corresponding $\{J, \widetilde{J}\}$-structures, i. e. $\mathcal{A}(\Gamma_1) = \mathcal{A}_1$ and $\mathcal{A}(\Gamma_2) = \mathcal{A}_2$. Obviously, $\mathcal{A}_1 \equiv_{\mathcal{C}^k} \mathcal{A}_2$ implies $\Gamma_1 \equiv_{\mathcal{C}^k} \Gamma_2$. By Proposition 4.6, $\Gamma_1 \not\to (\mathbb{L}; J, \widetilde{J})$ and $\Gamma_2 \to (\mathbb{L}; J, \widetilde{J})$; thus, $\Gamma_1 \models \varphi$ and $\Gamma_2 \not\models \varphi$, in contradiction to $\Gamma_1 \equiv_{\mathcal{C}^k} \Gamma_2$. □

4.8 Corollary. $\mathrm{CSP}(\mathbb{L}; J)$ *is inexpressible in* FPC.

Proof. This follows immediately from Theorem 4.7, since $(\mathbb{L}; J, \widetilde{J}) \in \mathbf{D}(\mathbb{L}; J) \subseteq \mathbf{HP}(\mathbb{L}; J)$ and pp-constructibility preserves expressibility in FPC by Theorem 2.33. □

5 Boolean Phylogeny CSPs

5.1 Horn Formulas and the Operation tb

5.1 Definition (Affine and Boolean Horn relations)**.** For $B \subseteq \{0,1\}^n$, we set

$$\varphi_B(z_1,\dots,z_n) \equiv (z_1 = \cdots = z_n) \vee \bigvee_{t \in B \setminus \{(0,\dots,0),(1,\dots,1)\}} \{z_i \colon t_i = 0\} | \{z_i \colon t_i = 1\}.$$

φ_B is called *affine* if $B \cup \{(0,\dots,0),(1,\dots,1)\}$ is an affine subspace of $\{0,1\}^n$, and it is called *Boolean* if $B \cup \{(0,\dots,0),(1,\dots,1)\}$ is a Boolean algebra with respect to $\min, \max$ and the Boolean complementation. An *affine* (resp. *Boolean*) *Horn clause* is a formula of the form

$$x_1 \neq y_1 \vee \cdots \vee x_m \neq y_m$$

or of the form

$$x_1 \neq y_1 \vee \cdots \vee x_m \neq y_m \vee \varphi_B(z_1,\dots,z_n),$$

where φ_B is affine (resp. Boolean). An *affine* (resp. *Boolean*) *Horn formula* is a conjunction of affine (resp. Boolean) Horn clauses.

5.2 Definition (Perfect domination)**.** Let $U, V \subseteq \mathbb{L}$. A function $f \colon \mathbb{L}^2 \to \mathbb{L}$ is called *perfectly dominated by the first argument on* $U \times V$ if the following holds:

- For all $u_1, u_2, u_3 \in U$ and $v_1, v_2, v_3 \in V$, if $u_1 | u_2 u_3$, then $f(u_1, v_1) | f(u_2, v_2) f(u_3, v_3)$.
- For all $u \in U$ and $v_1, v_2, v_3 \in V$, if $v_1 | v_2 v_3$, then $f(u, v_1) | f(u, v_2) f(u, v_3)$.

It is called *perfectly dominated by the second argument on* $U \times V$ if the function $(x, y) \mapsto f(y, x)$ is perfectly dominated by the first argument on $V \times U$.

The following property is simply called *semidomination* in [1].

5.3 Definition (Balanced semidomination)**.** Let U be a finite subset of $\mathbb{L}$ and $f \colon \mathbb{L}^2 \to \mathbb{L}$. We inductively define when f is *balanced semidominated on* $U \times U$: If $|U| \leq 1$, then f is balanced semidominated on $U \times U$. If $|U| \geq 2$, then f is balanced semidominated on $U \times U$ if there is a partition of U into nonempty sets U_0 and U_1 such that $U_0 | U_1$ and the following conditions hold:

- f is balanced semidominated on $U_0 \times U_0$ and $U_1 \times U_1$,
- f is perfectly dominated by the first argument on $U_0 \times U_1$ and perfectly dominated by the second argument on $U_1 \times U_0$,
- $f(U_0 \times U_0) \mid f(U_1 \times U_1)$,
- $f(U_0 \times U_1) \mid f(U_1 \times U_0)$ and
- $f((U_0 \times U_1) \cup (U_1 \times U_0)) \mid f((U_0 \times U_0) \cup (U_1 \times U_1))$.

5.4 Definition (Unbalanced semidomination)**.** Let U be a finite subset of $\mathbb{L}$ and $f\colon \mathbb{L}^2 \to \mathbb{L}$. We inductively define when f is *unbalanced semidominated on* $U \times U$: If $|U| \leq 1$, then f is unbalanced semidominated on $U \times U$. If $|U| \geq 2$, then f is unbalanced semidominated on $U \times U$ if there is a partition of U into nonempty sets U_0 and U_1 such that $U_0|U_1$ and the following conditions hold:

- f is unbalanced semidominated on $U_0 \times U_0$ and $U_1 \times U_1$,
- f is perfectly dominated by the first argument on $U_0 \times U_1$ and perfectly dominated by the second argument on $U_1 \times U_0$,
- $f(U_0 \times U_0) \mid f(U_1 \times U_1)$,
- $f(U_0 \times U_1) \mid f(U_1 \times U_0)$,
- $f((U_0 \times U_1) \cup (U_1 \times U_0)) \mid f(U_0 \times U_0)$ and
- $f((U_0 \times U_1) \cup (U_1 \times U_0) \times (U_0 \times U_0)) \mid f(U_1 \times U_1)$.

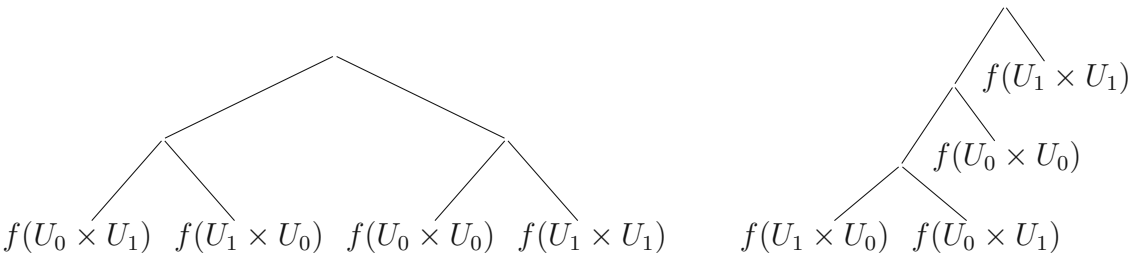


Figure 7. An illustration of balanced (left) and unbalanced (right) semidomination.

5.5 Definition (Affine and Boolean tree operations)**.** A function $f\colon \mathbb{L}^2 \to \mathbb{L}$ is called an *affine tree operation* if f is balanced semidominated on $U \times U$ for every finite $U \subseteq \mathbb{L}$; f is called a *Boolean tree operation* if it is unbalanced semidominated on $U \times U$ for every finite $U \subseteq \mathbb{L}$.

5.6 Proposition ([1, Proposition 7.3])**.** *There exists an affine tree operation* tx*. For every finite* $U \subseteq \mathbb{L}$*, there is some* $\gamma \in \mathrm{Aut}(\mathbb{L}; C)$ *such that* $\mathrm{tx}(x, y) = \gamma(\mathrm{tx}(y, x))$ *for all* $x, y \in U$*.*

We will show analagously that there exists a Boolean tree operation tb with the same property. The proof is in fact almost the same as for tx, with the conditions for balanced semidomination replaced by those for unbalanced semidomination. We first need the following lemma:

5.7 Lemma (cf. [1, Lemma 7.2])**.** *Let* $X \subseteq \mathbb{L}$ *be finite. Then there is a function* $f\colon X \times X \to \mathbb{L}$ *such that*

(1) *for every* $U \subseteq X$*,* f *is unbalanced semidominated on* $U \times U$ *and*

(2) *for all* $U_0, U_1 \subseteq X$ *such that* $U_0|U_1$ *and* $U_0 \prec U_1$*,* f *is perfectly dominated by the first argument on* $U_0 \times U_1$ *and by the second argument on* $U_1 \times U_0$*.*

Proof. f is constructed by induction on $|X|$. If $X = \{x\}$, we can take an arbitrary $a \in \mathbb{L}$ and set $f(x,x) \coloneqq a$. So let $|X| \geq 2$. Let $\{X_0, X_1\}$ be a nontrivial partition of X with $X_0|X_1$ and $X_0 \prec X_1$. By the induction hypothesis, there are functions $f_{0,0}\colon X_0 \times X_0 \to \mathbb{L}$ and $f_{1,1}\colon X_1 \times X_1 \to \mathbb{L}$ that satisfy (1) and (2) for X_0 and X_1, respectively. We can assume that $f_{0,0}(X_0 \times X_0) \mid f_{1,1}(X_1 \times X_1)$ (otherwise, there exist $\alpha, \beta \in \mathrm{Aut}(\mathbb{L}; C)$ such that $\alpha(f_{0,0}(X_0 \times X_0)) \mid \beta(f_{1,1}(X_1 \times X_1))$ by the homogeneity of $(\mathbb{L}; C)$).

Let $f_{0,1}\colon X_0 \times X_1 \to \mathbb{L}$ and $f_{1,0}\colon X_1 \times X_0 \to \mathbb{L}$ be such that $f_{0,1}$ is perfectly dominated by the first argument on $X_0 \times X_1$ and $f_{1,0}$ is perfectly dominated by the second argument on $X_1 \times X_0$. By again exploiting the homogeneity of $(\mathbb{L}; C)$, we can assume that

$$\begin{aligned} f_{0,1}(X_0 \times X_1) &\mid f_{1,0}(X_1 \times X_0), \\ f_{0,1}(X_0 \times X_1) \cup f_{1,0}(X_1 \times X_0) &\mid f_{0,0}(X_0 \times X_0) \text{ and} \\ f_{0,1}(X_0 \times X_1) \cup f_{1,0}(X_1 \times X_0) \times f_{0,0}(X_0 \times X_0)) &\mid f_{1,1}(X_1 \times X_1). \end{aligned}$$

Now let $f\colon X \times X \to \mathbb{L}$ be defined by $f(x,y) \coloneqq f_{i,j}(x,y)$ if $x \in X_i$ and $y \in X_j$. It remains to show that f satisfies (1) and (2). Let $U \subseteq X$. If $U \subseteq X_0$ or $U \subseteq X_1$, then f is semidominated on $U \times U$ since $f_{0,0}$ and $f_{1,1}$ are. Otherwise, let $\{U_0, U_1\}$ be a partition of U such that $U_0|U_1$ and $U_0 \prec U_1$. Clearly, $U_0 \subseteq X_0$ and $U_1 \subseteq X_1$. Hence,

$$\begin{aligned} f(U_0 \times U_0) &\mid f(U_1 \times U_1), \\ f(U_0 \times U_1) &\mid f(U_1 \times U_0), \\ f(U_0 \times U_1) \cup f(U_1 \times U_0) &\mid f(U_0 \times U_0) \text{ and} \\ f(U_0 \times U_1) \cup f(U_1 \times U_0) \times f(U_0 \times U_0)) &\mid f(U_1 \times U_1), \end{aligned}$$

as required. Moreover, $f_{0,1}$ is perfectly dominated by the first argument on $X_0 \times X_1$ and $f_{1,0}$ is perfectly dominated by the second argument on $X_1 \times X_0$; thus, f is perfectly dominated by the first argument on $U_0 \times U_1$ and perfectly dominated by the second argument on $U_1 \times U_0$. Because $f_{0,0}$ and $f_{1,1}$ are unbalanced semidominated on $U_0 \times U_0$ and $U_1 \times U_1$, respectively, we conclude that f is unbalanced semidominated on $U \times U$. □

5.8 Proposition (cf. [1, Proposition 7.3])**.** *There exists a Boolean tree operation* tb*. For every finite $U \subseteq \mathbb{L}$, there is some $\gamma \in \mathrm{Aut}(\mathbb{L}; C)$ such that* $\mathrm{tb}(x,y) = \gamma(\mathrm{tb}(y,x))$ *for all $x, y \in U$.*

Proof. Let $X \subseteq \mathbb{L}$ be finite and let $f, g\colon X \times X \to \mathbb{L}$ be two functions satisfying conditions (1) and (2) from Lemma 5.7. We will show by induction on $|X|$ that there is some $\alpha \in \mathrm{Aut}(\mathbb{L}; C)$ such that $f(x,y) = \alpha(g(x,y))$ for all $x, y \in X$. If $|X| \leq 1$, this trivially holds; so let $|X| \geq 2$ and let $\{X_0, X_1\}$ be a nontrivial partition of X such that $X_0|X_1$ and $X_0 \prec X_1$. By the induction hypothesis, there are $\alpha_{0,0}, \alpha_{1,1} \in \mathrm{Aut}(\mathbb{L}; C)$ such that $f(x,y) = \alpha_{0,0}(g(x,y))$ for all $x, y \in X_0$ and $f(x,y) = \alpha_{1,1}(g(x,y))$ for all $x, y \in X_1$. Since f and g are perfectly dominated by the first argument on $X_0 \times X_1$ and by the second argument on $X_1 \times X_0$, there are $\alpha_{0,1}, \alpha_{1,0} \in \mathrm{Aut}(\mathbb{L}; C)$ such that $f(x,y) = \alpha_{0,1}(g(x,y))$ for all $(x,y) \in X_0 \times X_1$ and $f(x,y) = \alpha_{1,0}(g(x,y))$ for all $(x,y) \in X_1 \times X_0$. Define $\beta\colon g(X \times X) \to f(X \times X)$ by $\beta(g(x,y)) = \alpha_{i,j}(g(x,y))$ if $(x,y) \in X_i \times X_j$. (Note that this is well-defined since g is injective due to being unbalanced semidominated.) It follows

from

$$\begin{aligned} f(X_0 \times X_0) \mid f(X_1 \times X_1),\\ f(X_0 \times X_1) \mid f(X_1 \times X_0),\\ f((X_0 \times X_1) \cup (X_1 \times X_0)) \mid f(X_0 \times X_0),\\ f((X_0 \times X_1) \cup (X_1 \times X_0) \times (X_0 \times X_0)) \mid f(X_1 \times X_1) \end{aligned}$$

and the analogous conditions for g that β is a partial isomorphism of $(\mathbb{L}; C)$ and can thus be extended to some $\alpha \in \mathrm{Aut}(\mathbb{L}; C)$ due to homogeneity.

Let X, Y be arbitary finite subsets of $\mathbb{L}$ with $X \subseteq Y$ and let $f\colon X \times X \to \mathbb{L}$ be a function satisfying the conditions (1) and (2) from Lemma 5.7. By Lemma 5.7, there is a function $g\colon Y \times Y \to \mathbb{L}$ satisfying (1) and (2). As we have shown above, there is some $\alpha \in \mathrm{Aut}(\mathbb{L}; C)$ such that $f(x, y) = \alpha(g(x, y))$ for all $x, y \in X$. Thus, $\alpha \circ g$ is an extension of f to $Y \times Y$ satisfying (1) and (2). The existence of tb follows since $\mathbb{L}$ is countable.

For the second statement, note that, since tb is injective, the function $\gamma\colon \mathrm{tb}(\mathbb{L}^2) \to \mathrm{tb}(\mathbb{L}^2)$ given by $\gamma(\mathrm{tb}(x, y)) = \mathrm{tb}(y, x)$ is well-defined. We claim that $\gamma\big|_X$ is a partial isomorphism for every finite $X \subseteq \mathbb{L}$. We will again proceed by induction on $|X|$. If $|X| \leq 1$, the claim trivially holds, so let $|X| \geq 2$ and let $\{X_0, X_1\}$ be a nontrivial partition of X such that $X_0 | X_1$. By the induction hypothesis, $\gamma\big|_{\mathrm{tb}(X_0 \times X_0)}$ and $\gamma\big|_{\mathrm{tb}(X_1 \times X_1)}$ are partial isomorphisms of $(\mathbb{L}; C)$. Because tb is perfectly dominated by the first argument on $X_0 \times X_1$ and by the second argument on $X_1 \times X_0$, $\gamma\big|_{\mathrm{tb}(X_0 \times X_1)}$ and $\gamma\big|_{\mathrm{tb}(X_1 \times X_0)}$ are partial isomorphisms of $(\mathbb{L}; C)$. Let $A_1 \coloneqq \mathrm{tb}(X_0 \times X_0), A_2 \coloneqq \mathrm{tb}(X_1 \times X_1)$ and $A_3 \coloneqq \mathrm{tb}((X_0 \times X_1) \cup (X_1 \times X_0))$. Since, for all $i \neq j$, $A_i | A_j$ and $\gamma\big|_{A_i}$ is a partial isomorphism from A_i to A_i, we obtain that $\gamma\big|_X$ is a partial isomorphism of $(\mathbb{L}; C)$. Since X was arbitrary, we get $\gamma \in \mathrm{Emb}(\mathbb{L}; C)$. Obviously, γ is self-inverse, thus even $\gamma \in \mathrm{Aut}(\mathbb{L}; C)$ holds. □

A binary function f on a structure Γ is said to be *symmetric modulo endomorphisms* if there are $e_1, e_2 \in \mathrm{End}(\Gamma)$ such that $e_1(f(x, y)) = e_2(f(y, x))$ for all $x, y \in \Gamma$. Both tx and tb have this property, as the following lemma shows:

5.9 Lemma ([1, Lemma 7.7])**.** *Let Γ be ω-categorical and $f \in \mathrm{Pol}^{(2)}(\Gamma)$. Suppose that for every finite $A \subseteq \Gamma$ there is some $\gamma \in \mathrm{Aut}(\Gamma)$ such that $f(x, y) = \gamma(f(y, x))$ for all $x, y \in A$. Then f is symmetric modulo endomorphisms.*

We will cite the two dichotomies for reducts of $(\mathbb{L}; C)$ obtained in [1].

5.10 Theorem ([1, Theorem 8.8])**.** *Let Γ be a reduct of $(\mathbb{L}; C)$ such that $C \in \langle\Gamma\rangle$. Then the following statements are equivalent:*

(1) $N \notin \langle\Gamma\rangle$.

(2) *All relations in $\langle\Gamma\rangle$ are affine Horn.*

(3) $\mathrm{tx} \curvearrowright \Gamma$.

(4) *There are $f \in \mathrm{Pol}^{(2)}(\Gamma)$ and $e_1, e_2 \in \mathrm{End}(\Gamma)$ such that $e_1(f(x, y)) = e_2(f(y, x))$ for all $x, y \in \Gamma$.*

(5) *No expansion of Γ by finitely many constants pp-interprets $(\{0, 1\}; \mathrm{NAE})$.*

5.11 Theorem ([1, Theorem 3.1]). *Let Γ be a reduct of $(\mathbb{L}; C)$ with finite signature and let Δ be its model-complete core. If there are $f \in \mathrm{Pol}^{(2)}(\Delta)$ and $e_1, e_2 \in \mathrm{End}(\Delta)$ such that $e_1(f(x,y)) = e_2(f(y,x))$ for all $x, y \in \Delta$, then $\mathrm{CSP}(\Gamma)$ is in P. Otherwise, it is NP-complete.*

The polymorphism tx was a central tool for showing that, for reducts Γ of $(\mathbb{L}; C)$ with $C \in \langle\Gamma\rangle$, unless P = NP, $\mathrm{CSP}(\Gamma)$ is in P if and only if all relations in Γ have an affine Horn definition. Using the polymorphism tb, we will show that if all relations in a structure Γ have a Boolean Horn definition, then $\mathrm{CSP}(\Gamma)$ is even expressible in FP.

5.12 Definition (Split vector). Let $x \in \mathbb{L}^n$. Then $s \in \{0,1\}^n$ is a *split vector* of x if, for all $i, j, k \in \{1, \dots, n\}$, $s_i \neq s_j = s_k$ implies $x_i | x_j x_k$. s is called *nontrivial* if $s \notin \{(0, \dots, 0), (1, \dots, 1)\}$. For $a, b \in \mathbb{L}^k$, we write $a \sim_{\text{split}} b$ if either a and b are both constant tuples or they have a common nontrivial split vector. For a phylogeny relation $R \subseteq \mathbb{L}^n$, the *split relation* of R is the set $S(R) \coloneqq \{s \in \{0,1\}^n \mid s \text{ is the split vector of some } x \in R\}$.

Note that every nonconstant tuple has a nontrivial split vector and that if s is a split vector of x, then also the Boolean complement of s is.

5.13 Lemma (cf. [1, Lemma 7.4]). *Let $B \subseteq \{0,1\}^n$ be such that $B \cup \{(0, \dots, 0), (1, \dots, 1)\}$ is a Boolean algebra with respect to* min, max *and the Boolean complementation. Then* tb *preserves φ_B.*

Proof. Let $a, b \in \mathbb{L}^n$ such that $\varphi_B(a)$ and $\varphi_B(b)$ hold and let $A \coloneqq \{a_1, \dots, a_n\}$ and $B \coloneqq \{b_1, \dots, b_n\}$. We have to show $\varphi_B(\mathrm{tb}(a, b))$. Note that $\varphi_B(x) \iff \varphi_B(y)$ whenever $x \sim_{\text{split}} y$.

First, suppose that $|A| = 1$. If also $|B| = 1$, then all components of $\mathrm{tb}(a, b)$ are equal, which implies $\varphi_B(\mathrm{tb}(a, b))$. So let $|B| > 1$. If $A | B$, then $\mathrm{tb}(a, b) \sim_{\text{split}} b$, since $|A| = 1$ and tb is perfectly dominated by one argument on $A \times B$. The other possibility is the existence of a partition of B into two nonempty subsets B_0 and B_1 such that $(A \cup B_0) | B_1$. Since tb is unbalanced semidominated on $A \cup B$, it holds that $\mathrm{tb}((A \cup B_0)^2) \mid \mathrm{tb}((A \cup B_0) \times B_1)$; in particular, $\mathrm{tb}(A \times B_0) \mid \mathrm{tb}(A \times B_1)$, which implies $\mathrm{tb}(a, b) \sim_{\text{split}} b$.

The case that $|B| = 1$ is symmetric; so suppose that $|A| \geq 2$ and $|B| \geq 2$. Let $X \coloneqq A \cup B = \{x_1, \dots, x_m\}$ and let s be a nontrivial split vector of $(x_1, \dots, x_m)$. We will view s as a function $X \to \{0,1\}$. We distinguish the following cases:

- s is constant on A and on B: Then $A | B$, so tb is perfectly dominated by the first or by the second argument on $A \times B$. Thus, $\mathrm{tb}(a, b) \sim_{\text{split}} a$ or $\mathrm{tb}(a, b) \sim_{\text{split}} b$, respectively.

- s is constant on A, but not on B: Then there is a partition of B into two nonempty subsets B_0 and B_1 such that $(A \cup B_0) | B_1$. We obtain $\mathrm{tb}((A \cup B_0)^2) \mid \mathrm{tb}((A \cup B_0) \times B_1)$, hence $\mathrm{tb}(A \times B_0) \mid \mathrm{tb}(A \times B_1)$ and therefore $\mathrm{tb}(a, b) \sim_{\text{split}} b$. The case that s is constant on B, but not on A, is symmetric.

- s is neither constant on A nor on B: Let X_0 and X_1 be two nonempty sets that form a partition of X such that $X_0 | X_1$. Without loss of generality, assume that $\mathrm{tb}(X_1 \times X_1) \mid \mathrm{tb}((X_0 \times X_0) \cup (X_1 \times X_0) \cup (X_0 \times X_1))$. If $a \sim_{\text{split}} b$, then $\mathrm{tb}(X_0 \times X_0) \mid \mathrm{tb}(X_1 \times X_1)$ implies that $a \sim_{\text{split}} b \sim_{\text{split}} \mathrm{tb}(a, b)$.

So assume that $a \not\sim_{\text{split}} b$. Let $s' \in B$ be a nontrivial split vector of a and let $s'' \in B$ be a nontrivial split vector of b. Since s is not constant on a nor on b, we know that s' and s'' are both constant on X_0 as well as on X_1. Since B also contains the Boolean complements of s' and s'', we can assume without loss of generality that $s'\big|_{X_0 \cap A} = s''\big|_{X_0 \cap B} \equiv 0$ and $s'\big|_{X_1 \cap A} = s''\big|_{X_1 \cap B} \equiv 1$.

Consider the sets $U \coloneqq \{\mathrm{tb}(a,b)_i \mid s'_i = s''_i = 1\}$ and $V \coloneqq \{\mathrm{tb}(a,b)_i \mid s'_i = 0 \text{ or } s''_i = 0\}$. By the previous assumption, $U \subseteq \mathrm{tb}(X_1 \times X_1)$ and $V \subseteq \mathrm{tb}((X_0 \times X_0) \cup (X_1 \times X_0) \cup (X_0 \times X_1))$; thus, $U|V$. We know that the binary minimum operation preserves B, so $\min(s', s'') \in B$. By the definitions of U and V, we get $\mathrm{tb}(a,b)_i \in U \iff \min(s', s'') = 1$ and $\mathrm{tb}(a,b)_i \in V \iff \min(s', s'') = 0$, so $\min(s', s'')$ is a split vector of $\mathrm{tb}(a,b)$. □

5.14 Proposition. tb *preserves all Boolean Horn formulas.*

Proof. It suffices to show the claim for Boolean Horn clauses. Let a, b be two tuples which satisfy the clause $x_1 \neq y_1 \vee \cdots \vee x_m \neq y_m \vee \varphi_B(z_1, \ldots, z_n)$, where φ_B is Boolean. If either a or b satisfies one of the inequality disjuncts, then also $\mathrm{tb}(a,b)$ does since tb is injective. If, however, both a and b satisfy φ_B, then so does $\mathrm{tb}(a,b)$ by Lemma 5.13. □

5.15 Lemma. $\mathrm{tb} \curvearrowright C$, *but* $\mathrm{tb} \not\curvearrowright N$.

Proof. Let $x, y, z, x', y', z' \in \mathbb{L}$ such that $x|yz$ and $x'|y'z'$ and let $B \coloneqq \{(0,1,1),(1,0,0)\}$. By Lemma 5.13, tb preserves the formula $\varphi_B(x,y,z) \equiv x|yz \vee (x = y = z)$; thus, $\mathrm{tb}(x,x')|\mathrm{tb}(y,y')\mathrm{tb}(z,z') \vee (\mathrm{tb}(x,x') = \mathrm{tb}(y,y') = \mathrm{tb}(z,z'))$. Since $x \neq y$ and tb is injective, we obtain $\mathrm{tb}(x,x') \neq \mathrm{tb}(y,y')$, so $\mathrm{tb}(x,x')|\mathrm{tb}(y,y')\mathrm{tb}(z,z')$.

To show that tb does not preserve N, consider the following tree:

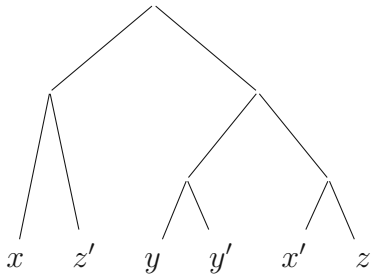


Figure 8.

Clearly, we have $N(x,y,z)$ and $N(x',y',z')$. By the semidomination property of tb (note that $\{x,z'\}|\{y,y',x',z\}$), we have $\mathrm{tb}(y,y')|\mathrm{tb}(x,x')\mathrm{tb}(z,z')$; thus, $N(\mathrm{tb}(x,x'),\mathrm{tb}(y,y'),\mathrm{tb}(z,z'))$ does not hold. □

5.16 Lemma. *Let R be a phylogeny relation such that* $\mathrm{tb} \curvearrowright R$. *Then $S(R)$ is closed under* min.

Proof. Let $s, s' \in S(R)$ be split vectors of $t, t' \in R$, respectively. We show the existence of some $t'' \in R$ with split vector $\min(s,s')$. Assume without loss of generality that $s, s' \notin \{(0,\ldots,0),(1,\ldots,1)\}$. Let u, v be arbitrary distinct elements of $\mathbb{L}$ and set $X_u \coloneqq \{x \in \mathbb{L} \colon ux|v\}, X_v \coloneqq \{x \in \mathbb{L} \colon u|xv\}$. tb is unbalanced semidominated on $\{u,v\}^2$; we can assume without loss of generality that $\mathrm{tb}(v,v)|\mathrm{tb}(u,u)\mathrm{tb}(u,v)\mathrm{tb}(v,u)$:

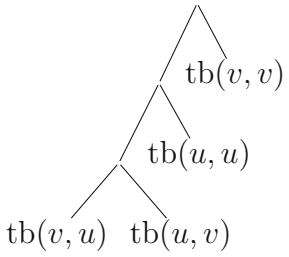


Figure 9.

By the homogeneity of $(\mathbb{L}; C)$, there are $\alpha, \beta \in \mathrm{Aut}(\mathbb{L}; C)$ such $\alpha(\{t_i : s_i = 0\}) \cup \beta(\{t'_i : s'_i = 0\}) \subseteq X_u$ and $\alpha(\{t_i : s_i = 1\}) \cup \beta(\{t'_i : s'_i = 1\}) \subseteq X_v$. For some $i \in \{1, \dots, \mathrm{ar}(R)\}$, we distinguish four cases:

- If $s_i = 0$ and $s'_i = 1$, then $v|\alpha(t_i)u$ and $u|\beta(t'_i)v$, hence $\mathrm{tb}(v,u)|\mathrm{tb}(\alpha(t_i), \beta(t'_i))\mathrm{tb}(u,v)$.
- If $s_i = 1$ and $s'_i = 0$, then $\mathrm{tb}(u,v)|\mathrm{tb}(\alpha(t_i), \beta(t'_i))\mathrm{tb}(v,u)$.
- If $s_i = 0$ and $s'_i = 0$, then $\mathrm{tb}(v,v)|\mathrm{tb}(\alpha(t_i), \beta(t'_i))\mathrm{tb}(u,u)$.
- If $s_i = 1$ and $s'_i = 1$, then $\mathrm{tb}(u,u)|\mathrm{tb}(\alpha(t_i), \beta(t'_i))\mathrm{tb}(v,v)$.

Hence, $\mathrm{tb}(v,v)\mathrm{tb}(\alpha(t_i), \beta(t'_i))|\mathrm{tb}(u,u)$ if and only if $\min(s_i, s'_i) = 1$. Therefore, $\min(s, s')$ is a split vector of $t'' \coloneqq \mathrm{tb}(\alpha(t), \beta(t')) \in R$. □

5.17 Proposition. *Let Γ be a reduct of $(\mathbb{L}; C)$ such that $C \in \langle\Gamma\rangle$ and $\mathrm{tb} \curvearrowright \Gamma$. Then every relation $R \in \langle\Gamma\rangle$ has a Boolean Horn definition.*

Proof. Since $\mathrm{tb} \curvearrowright C$ and $\mathrm{tb} \not\curvearrowright N$ by Lemma 5.15, all results from [1, § 5–6] remain valid for R. A closer inspection of the proofs there shows that it suffices to prove that $\varphi_{S(R)}$ is a Boolean Horn formula, i. e. that $S(R)$ is a Boolean algebra (cf. the statement of [1, Lemma 6.6] and the definition of ψ_R there). It in turn suffices to show that $S(R)$ is closed under min, since $S(R)$ is clearly closed under Boolean complementation and $\max(a,b) = (\min(a^c, b^c))^c$. Hence, the statement follows from Lemma 5.16. □

5.2 An FP Algorithm for Boolean Phylogeny CSPs

This section is about the relation

$$R_{\mathrm{tb}} \coloneqq \{(u,v,x,y,z) \in \mathbb{L}^5 : (u \neq v) \vee (xy|z) \vee (x = y = z)\}.$$

We will show that $\mathrm{CSP}(\mathbb{L}; R_{\mathrm{tb}}, \neq)$ is expressible in FP. Moreover, $(\mathbb{L}; R_{\mathrm{tb}}, \neq)$ pp-defines all phylogeny relations defined by a Boolean Horn formula. Hence, all CSPs with a Boolean Horn template are expressible in FP.

Consider the following algorithm:

Algorithm SolveB
Input: An instance Γ of $\mathrm{CSP}(\mathbb{L}; R_{\mathrm{tb}}, \neq)$
Output: $\top$ or $\bot$

$\widetilde{\Theta} \leftarrow \{(x,x) : x \in \Gamma\}$
while $\widetilde{\Theta}$ changes:
  $\Theta \leftarrow A^2$
  while Θ changes:
    $\Theta' \leftarrow \widetilde{\Theta}$
    for $(u,v,x,y,z) \in R_{\mathrm{tb}}^{\Gamma}$:
      if $(u,v) \in \widetilde{\Theta}$ and $(x,y),(y,z) \in \Theta$:
        $\Theta' \leftarrow \Theta' \cup \{(x,y),(y,x)\}$
    $\Theta \leftarrow \mathrm{tcl}(\Theta')$ // tcl is the transitive closure
  $\widetilde{\Theta} \leftarrow \Theta$
if there are $a, b \in \Gamma$ such that $(a,b) \in \widetilde{\Theta}$ and $(a,b) \in \neq^{\Gamma}$:
  return $\bot$
else:
  return $\top$

Figure 10. An FP algorithm for a generic phylogeny CSP preserved by tb.

5.18 Proposition. *The algorithm in Figure 10 is sound and complete for* $\mathrm{CSP}(\mathbb{L}; R_{\mathrm{tb}}, \neq)$, *i. e.*

$$\Gamma \to (\mathbb{L}; R_{\mathrm{tb}}, \neq) \iff \textsc{SolveB}(\Gamma) = \top.$$

Proof. For some $i \geq 0$, we write $\widetilde{\Theta}^i$ for the state of the program variable $\widetilde{\Theta}$ after the i-th iteration of the outermost loop. For some $i, j \geq 0$, we denote by $\Theta^{i,j}$ the state of Θ after the j-th iteration of the inner while-loop within the i-th iteration of the outermost loop. Since Γ is finite and both $\widetilde{\Theta}$ and Θ are increasing, they will eventually reach a limit; let ∞ denote the respective final iteration. Note that $\widetilde{\Theta}^i = \Theta^{i,\infty}$ for all $i \geq 1$.

»⇒«: Let $h : \Gamma \to (\mathbb{L}; R_{\mathrm{tb}}, \neq)$ be a solution to Γ. We will show that for every $i \geq 1$, $\widetilde{\Theta}^i(a,b)$ implies $\neg(\neq^{\Gamma}(a,b))$ for all $a, b \in \Gamma$. It suffices to show that $h(a) = h(b)$, since this implies $\neg(\neq^{\Gamma}(a,b))$ because h is a homomorphism. We will proceed by induction on i. If $i = 0$, then $a = b$ and hence $h(a) = h(b)$. So let $i \geq 1$ and let $(a,b) \in \widetilde{\Theta}^i$. Consider the last iteration of the inner loop within the i-th iteration of the outer loop: We have $(a,b) \in \Theta^{i,\infty} = \mathrm{tcl}(\Theta')$. Hence, there is a path $(x_1,x_2),(x_2,x_3),\ldots,(x_{n-1},x_n)$ in Θ' with $x_1 = a$ and $x_n = b$. We will show that $h(x_j) = h(x_{j+1})$ for all $j \in \{1,\ldots,n\}$. At some point, (x_j, x_{j+1}) is added to Θ'. If $(x_j, x_{j+1}) \in \widetilde{\Theta}^{i-1}$, then $h(x_j) = h(x_{j+1})$ by the induction hypothesis. So assume that (x_j, x_{j+1}) is added to Θ' within the for-loop, i. e. there are $u, v, z \in \Gamma$ such that $(u,v) \in \widetilde{\Theta}^{i-1}, (x_j, x_{j+1}), (x_{j+1}, z) \in \Theta^{i,\infty}$ and $R_{\mathrm{tb}}^{\Gamma}(u,v,x_j,x_{j+1},z)$. Because h is a solution, we obtain $R_{\mathrm{tb}}^{\mathbb{L}}(h(u),h(v),h(x_j),h(x_{j+1}),h(z))$ and hence, because $h(u) = h(v)$ by the induction hypothesis, $h(x_j)h(x_{j+1})|h(z)$ or $h(x_j) = h(x_{j+1}) = h(z)$. If the latter disjunct is true, we are done, so assume towards contradiction that the first one holds.

Let C be the connected component of x_j, x_{j+1} and z with respect to $\Theta^{i,\infty}$ and observe that it has at least two elements, since $h(x_j)h(x_{j+1})|h(z)$. Thus, there is a partition of C into two nonempty sets C_1 and C_2 such that $h(C_1)|h(C_2)$. Since C is connected, there must exist a tuple $(x, y) \in \Theta^{i,\infty} = \mathrm{tcl}(\Theta')$ with $x \in C_1$ and $y \in C_2$. We can even assume that $(x, y) \in \Theta'$. It is impossible that $(x, y) \in \widetilde{\Theta}^{i-1}$, since this would imply $h(x) = h(y)$; thus, (x, y) is added to Θ' within the for-loop. Subsequently, there are elements $\widetilde{u}, \widetilde{v}, \widetilde{z} \in \Gamma$ such that $(\widetilde{u}, \widetilde{v}) \in \widetilde{\Theta}^{i-1}, (y, \widetilde{z}) \in \Theta^{i,\infty}$ and $R^{\Gamma}_{\mathrm{tb}}(\widetilde{u}, \widetilde{v}, x, y, \widetilde{z})$. By the induction hypothesis, $h(\widetilde{u}) = h(\widetilde{v})$, thus, similiary to before, we obtain $h(x)h(y)|h(\widetilde{z})$ or $h(x) = h(y) = h(\widetilde{z})$. Both cases contradict the assumption that $x \in C_1, y \in C_2$ and $h(C_1)|h(C_2)$.

»⇐«: We construct a rooted tree T whose leaves are the equivalence classes $[\cdot]$ of $\widetilde{\Theta}^{\infty}$ such that $R^{\Gamma}_{\mathrm{tb}}(u, v, a, b, c)$ implies $([u] \neq [v]) \vee ([a][b]||[c]) \vee ([a] = [b] = [c])$ for all $u, v, a, b, c \in \Gamma$. The tree T which we construct is in general not binary, but from T we can build a binary tree T' with the same property by Lemma 3.4. T' can be embedded into $(\mathbb{L}; C)$ and the respective embedding composed with the canonical projection $x \mapsto [x]$ is a solution to Γ.

For $i \geq 0$, the vertices on the i-th level of T shall be the sets of equivalence classes of $\widetilde{\Theta}^{\infty}$ whose representatives form a connected component of $\Theta^{\infty,i}$. This is well-defined: Assume that $(a, b) \in \Theta^{\infty,i}$ and $(a, a'), (b, b') \in \widetilde{\Theta}^{\infty}$; since $\widetilde{\Theta}^{\infty} \subseteq \Theta^{\infty,i}$ and $\Theta^{\infty,i}$ is transitively closed, we can infer $(a', b') \in \Theta^{\infty,i}$. Since $\widetilde{\Theta}^{\infty} = \Theta^{\infty,\infty}$, the leaves of T exactly correspond to the equivalence classes of $\widetilde{\Theta}^{\infty}$.

Let $u, v, a, b, c \in \Gamma$ such that $R^{\Gamma}_{\mathrm{tb}}(u, v, a, b, c)$. If $[u] \neq [v]$ or $[a] = [b] = [c]$, there is nothing to show. So assume that $[u] = [v]$ and $\neg([a] = [b] = [c])$. We distinguish two cases: If $[a] = [b] \neq [c]$, we can trivially conclude $[a][b]||[c]$. If, however, $[a] \neq [b]$, then there is some i such that $(a, b) \in \Theta^{\infty,i-1} \setminus \Theta^{\infty,i}$; in other words: $[a]$ and $[b]$ are in the same vertex on the $(i-1)$-th level, but in different vertices on the i-th level. Assume towards contradiction that $(a, c) \in \Theta^{\infty,i-1}$ (or, equivalently, $(b, c) \in \Theta^{\infty,i-1}$). Since $(u, v) \in \widetilde{\Theta}^{\infty}$ and $R^{\Gamma}_{\mathrm{tb}}(u, v, a, b, c)$ hold, (a, b) is added to $\Theta^{\infty,i}$ in the for-loop, contradiction. Hence, $[a][b]||[c]$, as required. □

As a rule of thumb, an algorithm can be translated into an FP formula if it does not contain choices of arbitrary elements; we will perform the translation for SolveB:

5.19 Corollary. $\mathrm{CSP}(\mathbb{L}; R_{\mathrm{tb}}, \neq)$ *is expressible in* FP.

Proof. We will translate the algorithm SolveB into an FP formula φ, i. e. $\Gamma \models \varphi \iff \textsc{SolveB}(\Gamma) = \bot$. Then $\Gamma \not\to (\mathbb{L}; R_{\mathrm{tb}}, \neq) \iff \Gamma \models \varphi$ by Proposition 5.18. It is easy to see that the formula

$$\varphi \equiv \exists a, b\big(\big[\mathsf{ifp}_{\widetilde{\Theta},(x,y)}\psi(x, y)\big](a, b) \wedge {\neq}(a, b)\big)$$

is a suitable translation, where

$$\psi(x, y) \equiv x = y \vee \big[\mathsf{dfp}_{\Theta,(\widetilde{x},\widetilde{y})}\big[\mathrm{tcl}\, \varrho\big](\widetilde{x}, \widetilde{y})\big](x, y)$$

and

$$\varrho(\widetilde{x}, \widetilde{y}) \equiv \widetilde{\Theta}(\widetilde{x}, \widetilde{y}) \vee \big(\exists u, v, z\big(R_{\mathrm{tb}}(u, v, \widetilde{x}, \widetilde{y}, z) \wedge \widetilde{\Theta}(u, v) \wedge \Theta(\widetilde{x}, \widetilde{y}) \wedge \Theta(\widetilde{y}, z)\big)\big).$$

$\big[\mathrm{tcl}\,\varrho\big](\widetilde{x},\widetilde{y})$ in turn is an abbreviation for the formula

$$\big[\mathsf{ifp}_{Z,(s,t)}\,\varrho(s,t)\vee\exists z(\varrho(s,z)\wedge Z(z,t))\big](\widetilde{x},\widetilde{y});$$

it computes the transitive closure of ϱ, cf. Example 2.31. □

5.20 Lemma. *Let $B\subseteq\{0,1\}^n$. Then B is a nontrivial Boolean algebra with respect to* $\min,\max$ *and the Boolean complementation if and only if B is characterized by a set of equations of the form $x_i+x_j=0$ (where $+$ is the addition modulo 2).*

Proof. One implication is trivial: If $a,b\in\{0,1\}^n$ satisfy the equation $x_i+x_j=0$, then it is also satisfied by $\max(a,b)$, $\min(a,b)$ and the Boolean complement a^{c} of a. B is nontrivial since it contains the two elements $(0,\dots,0)$ and $(1,\dots,1)$.

So assume that B is a nontrivial Boolean algebra. Let E be the set of all equations of the form $x_i+x_j=0$ which hold for all elements of B. Let a be a tuple from $\{0,1\}^n$ that satisfies all equations from E; we have to show that $a\in B$.

We define an equivalence relation on $\{x_1,\dots,x_n\}$ by $x_i\sim x_j$ if and only if E contains the equation $x_i+x_j=0$. Without loss of generality, assume that $x_1,\dots,x_k$ are representatives of the classes of $\sim$. Let $B':=\{(x_1,\dots,x_k)\mid(x_1,\dots,x_n)\in B\}$ and note that B' is a Boolean algebra which only satisfies the trivial equations $x_i+x_i=0$ of length 2. We claim that $B'=\{0,1\}^k$. It suffices to show that all k-ary unit vectors are contained in B' since all vectors from $\{0,1\}^k$ can be built from them using max. It in turn suffices to show that for all $y\in B'$ which have at least two entries containing 1, there is a tuple $w\in B'$ such that $(0,\dots,0)<w<y$, where $u<v$ if and only if $u=\min(u,v)$ and $u\neq v$. Assume without loss of generality that $y_1=y_2=1$. Since B' only satisfies trivial equations of length 2, there is some $z\in B'$ with $z_1=0$ and $z_2=1$. Now $w:=\min(y,z)$ is a suitable choice.

Thus, we have proven that $B'=\{0,1\}^k$. In particular, $(a_1,\dots,a_k)\in B'$, which means that there are $\widetilde{a}_{k+1},\dots,\widetilde{a}_n$ such that $(a_1,\dots,a_k,\widetilde{a}_{k+1},\dots,\widetilde{a}_n)\in B$. But $\widetilde{a}_{k+1},\dots,\widetilde{a}_n$ are uniquely determined by $a_1,\dots,a_k$ and E; so $\widetilde{a}_i=a_i$ for all $i\in\{k+1,\dots,n\}$, which concludes the proof. □

5.21 Lemma. $(\mathbb{L};R_{\mathrm{tb}},\neq)$ *pp-defines every phylogeny relation defined by a Boolean Horn formula.*

Proof. It suffices to show the claim for Boolean Horn clauses.

We will first show by induction on m that the $2m$-ary relation defined by the clause $\varrho_m(x_1,y_1,\dots,x_m,y_m)\equiv x_1\neq y_1\vee\dots\vee x_m\neq y_m$ has a pp-definition in $(\mathbb{L};R_{\mathrm{tb}},\neq)$. This is trivial for $m=1$. For $m\geq1$, we claim that

$$\begin{aligned}&\varrho_{m+1}(x_1,y_1,\dots,x_{m+1},y_{m+1})\iff\\&\quad\exists a\big(\varrho_m(x_1,y_1,\dots,x_{m-1},y_{m-1},x_m,a)\wedge R_{\mathrm{tb}}(x_{m+1},y_{m+1},x_m,a,y_m)\big).\end{aligned}\tag{4}$$

»⇒«: First, assume that $\varrho_{m-1}(x_1,y_1,\dots,x_{m-1},y_{m-1})$ holds. If $x_m=y_m$, we set $a:=x_m$; if $x_m\neq y_m$, then there is some $a\in\mathbb{L}$ with $a\neq x_m$ such that $x_ma|y_m$. In both cases, $R_{\mathrm{tb}}(x_{m+1},y_{m+1},x_m,a,y_m)$ is fulfilled. Next, assume that $\varrho_{m-1}(x_1,y_1,\dots,x_{m-1},y_{m-1})$ is not true; this implies $x_m\neq y_m$ or $x_{m+1}\neq y_{m+1}$. If the former holds, we obtain the existence of some $a\in\mathbb{L}$ with $a\neq x_m$ and $x_ma|y_m$; if $x_{m+1}\neq y_{m+1}$, choose an

arbitrary $a \in \mathbb{L}$ such that $a \neq x_m$. Either way, $\varrho_m(x_1, y_1, \ldots, x_{m-1}, y_{m-1}, x_m, a)$ as well as $R_{\mathrm{tb}}(x_{m+1}, y_{m+1}, x_m, a, y_m)$ hold.

»⇐«: If $\varrho_{m-1}(x_1, y_1, \ldots, x_{m-1}, y_{m-1})$ holds, then we are done. Otherwise, $x_m \neq a$. Since $R_{\mathrm{tb}}(x_{m+1}, y_{m+1}, x_m, a, y_m)$, we obtain $x_{m+1} \neq y_{m+1}$ or $x_m a | y_m$, where the latter of course implies $x_m \neq y_m$.

Now, for some Boolean algebra $B \subseteq \{0,1\}^n$ and $m \geq 0$, consider the clause

$$\varphi^B_{m,n}(x_1, y_1, \ldots, x_m, y_m, z_1, \ldots, z_n) \equiv x_1 \neq y_1 \vee \cdots \vee x_m \neq y_m \vee \varphi_B(z_1, \ldots, z_n).$$

We will again use induction on m to show that the relation defined by $\varphi^B_{m,n}$ has a pp-definition $\psi^B_{m,n}$ in $(\mathbb{L}; R_{\mathrm{tb}}, \neq)$. By Lemma 5.20, B is characterized by a set of equations of the form $z_{i_k} + z_{j_k} = 0$, $k \in \{1, \ldots, \ell\}$. For every k, let $B_k \subseteq \{0,1\}^n$ be the Boolean algebra characterized by the single equation $z_{i_k} + z_{j_k} = 0$. It suffices to show the claim for each B_k since $\varphi^B_{m,n} \iff \varphi^{B_1}_{m,n} \wedge \cdots \wedge \varphi^{B_\ell}_{m,n}$. Hence, we can assume without loss of generality that $B = \{z \in \{0,1\}^n \colon z_1 + z_2 = 0\}$.

In light of Lemma 3.3 (3), it is easy to see that

$$\varphi^B_{0,n} \iff \bigwedge_{i,j \in \{3,\ldots,n\}} R_{\mathrm{tb}}(z_1, z_1, z_1, z_2, z_i) \wedge R_{\mathrm{tb}}(z_1, z_1, z_i, z_j, z_1) \wedge R_{\mathrm{tb}}(z_1, z_1, z_i, z_j, z_2)$$

and

$$\varphi^B_{1,n} \iff \bigwedge_{i,j \in \{3,\ldots,n\}} R_{\mathrm{tb}}(x_1, y_1, z_1, z_2, z_i) \wedge R_{\mathrm{tb}}(x_1, y_1, z_i, z_j, z_1) \wedge R_{\mathrm{tb}}(x_1, y_1, z_i, z_j, z_2).$$

For $m \geq 1$, one can show very similiarly to (4) that

$$\varphi^B_{m+1,n}(x_1, y_1, \ldots, x_{m+1}, y_{m+1}, z_1, \ldots, z_n) \iff$$
$$\exists a \big(\varphi^B_{m,n}(x_1, y_1, \ldots, x_{m-1}, y_{m-1}, x_m, a, z_1, \ldots, z_n) \wedge R_{\mathrm{tb}}(x_{m+1}, y_{m+1}, x_m, a, y_m)\big). \qquad \square$$

5.22 Corollary. *Let Γ be a reduct of $(\mathbb{L}; C)$ such that all relations in Γ have a Boolean Horn definition. Then* $\mathrm{CSP}(\Gamma)$ *is expressible in* FP.

Proof. By Lemma 5.21, $(\mathbb{L}; R_{\mathrm{tb}}, \neq)$ pp-defines Γ. By Corollary 5.19, $\mathrm{CSP}(\mathbb{L}; R_{\mathrm{tb}}, \neq)$ is expressible in FP. Thus, $\mathrm{CSP}(\Gamma)$ is expressible in FP as well by Theorem 2.33. $\square$

References

[1] Manuel Bodirsky, Peter Jonsson, Trung Van Pham (2016): *The Complexity of Phylogeny Constraint Satisfaction.* In: 33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016). Leibniz International Proceedings in Informatics (LIPIcs), Vol. 47, 20:1–20:13, Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[2] Manuel Bodirsky, David Bradley-Williams, Michael Pinsker, András Pongrácz (2018): *The universal homogeneous binary tree.* Journal of Logic and Computation, 28(1), 133–163.

[3] Manuel Bodirsky, Jakub Rydval (2022): *On the Descriptive Complexity of Temporal Constraint Satisfaction Problems.* J. ACM 70, 1, Article 2 (February 2023).

[4] Libor Barto, Jakub Opršal, Michael Pinsker (2015): *The wonderland of reflections.* Israel Journal of Mathematics, Vol. 223.

[5] Manuel Bodirsky, Peter Jonsson, Trung Van Pham (2016): *The Reducts of the Homogeneous Binary Branching C-relation.* J. Log. Comp. 81, 4 (2016), 1255–1297.

[6] Libor Barto, Andrei Krokhin, Ross Willard (2017): *Polymorphisms, and How to Use Them.* In: The Constraint Satisfaction Problem: Complexity and Approximability. Dagstuhl Follow-Ups, Vol. 7, 1–44, Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[7] Manuel Bodirsky, Jan Kára (2006): *The Complexity of Equality Constraint Languages.* In: D. Grigoriev, J. Harrison, E. A. Hirsch (eds.): Computer Science – Theory and Applications. CSR 2006. Lecture Notes in Computer Science, Vol. 3967. Springer, Berlin, Heidelberg.

[8] Manuel Bodirsky, Hubie Chen, Michael Pinsker (2010): *The reducts of equality up to primitive positive interdefinability.* J. Symb. Log. 75, 4 (2010), 1249–1292.

[9] Manuel Bodirsky, Jan Kára (2010): *The complexity of temporal constraint satisfaction problems.* J. ACM 57, 2, Article 9 (January 2010).

[10] Manuel Bodirsky (2007): *Cores of Countably Categorical Structures.* In: Logical Methods in Computer Science, Vol. 3, Issue 1 (January 2007), Centre pour la Communication Scientifique Directe (CCSD).

[11] Manuel Bodirsky, Michael Pinsker (2014): *Minimal functions on the random graph.* Israel Journal of Mathematics, Vol. 200.

[12] Daoud Nasri Siniora (2017): *Automorphism Groups of Homogeneous Structures.* PhD Thesis, University of Leeds.

[13] Manuel Bodirsky, Jaroslav Nešetřil (2003): *Constraint Satisfaction with Countable Homogeneous Templates.* In: M. Baaz, J. Makowsky (eds.): Computer Science Logic. CSL 2003. Lecture Notes in Computer Science, Vol. 2803. Springer, Berlin, Heidelberg.

[14] Martin Grohe (2008): *The Quest for a Logic Capturing PTIME.* 23rd Annual IEEE Symposium on Logic in Computer Science, Pittsburgh, PA, USA, 2008, 267–271.

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

[16] Lauri Hella (1996): *Logical Hierarchies in PTIME.* Information And Computation 129 (1):1–19.

[17] Anuj Dawar (2015): *The nature and power of fixed-point logic with counting.* ACM SIGLOG News 2, 1 (January 2015), 8–21.

[18] Manuel Bodirsky (2021): *Complexity of Infinite-Domain Constraint Satisfaction.* Cambridge University Press.

[19] Wilfrid Hodges (1993): *Model Theory.* Cambridge University Press.

[20] Leonid Libkin (2004): *Elements of Finite Model Theory.* Springer.