

DICHOTOMY RESULTS FOR CLASSES OF COUNTABLE GRAPHS

VITTORIO CIPRIANI, EKATERINA FOKINA, MATTHEW HARRISON-TRAINOR, LILING KO,
AND DINO ROSSEGGER

ABSTRACT. We study classes of countable graphs where every member does not contain a given finite graph as an induced subgraph—denoted by $\text{Free}(\mathcal{G})$ for a given finite graph \mathcal{G} . Our main results establish a structural dichotomy for such classes: If \mathcal{G} is not an induced subgraph of \mathcal{P}_4 , then $\text{Free}(\mathcal{G})$ is on top under effective bi-interpretability, implying that the members of $\text{Free}(\mathcal{G})$ exhibit the full range of structural and computational behaviors. In contrast, if \mathcal{G} is an induced subgraph of \mathcal{P}_4 , then $\text{Free}(\mathcal{G})$ is structurally simple, as witnessed by the fact that every member satisfies the computable embeddability condition.

This dichotomy is mirrored in the finite setting when one considers combinatorial and complexity-theoretic properties. Specifically, it is known that $\text{Free}(\mathcal{G})^{fin}$ is complete for graph isomorphism and not a well-quasi-order under embeddability whenever \mathcal{G} is not an induced subgraph of \mathcal{P}_4 , while in all other cases $\text{Free}(\mathcal{G})^{fin}$ forms a well-quasi-order and the isomorphism problem for $\text{Free}(\mathcal{G})^{fin}$ is solvable in polynomial time.

1. INTRODUCTION

Given a collection \mathcal{S} of finite graphs, let

$$\text{Free}(\mathcal{S}) = \{\mathcal{H} : \text{no } \mathcal{G} \in \mathcal{S} \text{ is an induced subgraph of } \mathcal{H}\}$$

be the set of \mathcal{S} -free graphs. Such classes of graphs are well-studied in combinatorics and many natural graph classes arise in this way. For example, the bipartite graphs are the \mathcal{S} -free graphs where \mathcal{S} is the collection of odd cycles and the chordal graphs are the \mathcal{S} -free graphs where \mathcal{S} is the collection of cycles of length four or more. Other classes of graphs defined in such a way include the perfect graphs, line graphs of graphs, and split graphs. In combinatorics, graphs are usually assumed to be finite. Given a collection \mathcal{S} , a general paradigm to solving questions about the \mathcal{S} -free graphs is to show that the \mathcal{S} -free graphs are structured, often by admitting some kind of decomposition. Given such structure we can for example count the number of \mathcal{S} -free graphs on n vertices or develop efficient algorithms for solving problems such as the colourability or isomorphism problems. On the other hand, sometimes we can show that the \mathcal{S} -free graphs have no more structure than graphs in general. The bipartite graphs are an example of an unstructured class of graphs, and this can be shown by the fact that any graph \mathcal{G} can be transformed into a bipartite graph \mathcal{H} by bisecting each edge.

In this paper we will restrict to the case of a single forbidden induced subgraph \mathcal{G} , and write $\text{Free}(\mathcal{G})$ for the \mathcal{G} -free graphs. For finite graphs, this case is well-understood in combinatorics with a structural dichotomy centered around the graph \mathcal{P}_4 , the path consisting of four vertices. $\text{Free}(\mathcal{P}_4)$ enjoys several equivalent characterizations; the one most relevant to our purposes is its equivalence with the class of complement-reducible graphs, in short, *cographs*, introduced in [CLB81]. Such graphs have natural applications, e.g., in bioinformatics they naturally model certain binary relations between genes [Hel+13]. In the finite case, cographs are defined as the smallest class of graphs that contain a single-vertex graph and are closed under disjoint union and complementation. This allows to give a recursive definition, yielding a natural representation of any cograph as a labeled tree known as its *decomposition tree* (see Section 2).

As shown in [CLB81], this tree-based decomposition makes it possible to solve, in polynomial time, a number of problems for finite graphs that are hard or even NP-complete in general. These include, for example, problems such as graph isomorphism, Hamiltonicity, clique detection, and graph coloring.

2020 *Mathematics Subject Classification*. 03C57, 03E15, 05C63, 05C60.

Key words and phrases. Forbidden subgraphs, Bi-interpretability, Borel reducibility, Σ -small, computable embeddability condition.

The first, second and fourth author were supported by the Austrian Science Fund (FWF) 10.55776/P36781. The third author was supported by a Sloan Research Fellowship and by NSF grants DMS-2452105 and DMS-2419591. The last author was supported by the Austrian Science Fund (FWF) 10.55776/PIN1878224.

Structural dichotomies for classes of graphs of the form $\text{Free}(\mathcal{G})$ tend to center around $\text{Free}(\mathcal{P}_4)$. For example, if \mathcal{G} is an induced subgraph of \mathcal{P}_4 (so that $\text{Free}(\mathcal{G}) \subseteq \text{Free}(\mathcal{P}_4)$) then graph isomorphism for $\text{Free}(\mathcal{G})$ is solvable in polynomial time, and otherwise if \mathcal{G} is not an induced subgraph of \mathcal{P}_4 (so that $\text{Free}(\mathcal{G}) \not\subseteq \text{Free}(\mathcal{P}_4)$) then graph isomorphism for $\text{Free}(\mathcal{G})$ is as hard as graph isomorphism in general¹ (GI -complete, see [BC79]). Similarly, $\text{Free}(\mathcal{G})$ is a well-quasi-order under the induced subgraph relation if and only if \mathcal{G} is an induced subgraph of \mathcal{P}_4 [Dam90]. Thus, the cographs are the dividing line between structure and non-structure: $\text{Free}(\mathcal{G})$ is tractable if and only if $\text{Free}(\mathcal{G})$ consists entirely of cographs.

In this paper we study classes of countably infinite graphs. Our perspective is that of computable structure theory and descriptive set theory, branches of mathematical logic which provide a framework—based on algorithms, respectively, topology—for analyzing the complexity of countably infinite structures. When considering classes of structures, we typically observe two extremes. On one side are classes subject to strong global structural constraints, which tightly restrict the behavior of their structures and are thus structurally simple. On the other side lie classes that are on top with respect to some notion of reducibility. These classes accommodate the full spectrum of behaviors and serve as universal benchmarks in complexity comparisons. These classes are thus structurally complex.

As in the finite case, in the infinite case we establish $\text{Free}(\mathcal{P}_4)$ as the main character of structural dichotomies for classes of graphs of the form $\text{Free}(\mathcal{G})$. We show the following results:

- When \mathcal{G} is an induced subgraph of \mathcal{P}_4 , $\text{Free}(\mathcal{G})$ is structurally simple:
 - $\text{Free}(\mathcal{G})$ is Σ -small, which means that there are at most countably many existential types.
 - $\text{Free}(\mathcal{G})$ satisfies the computable embeddability condition, which means that each existential type is computable.
 - $\text{Free}(\mathcal{G})$ is not on top for bi-interpretability, which means that there is a graph which is not bi-interpretable with a \mathcal{G} -free graph. (In particular, there is an automorphism group of a graph which is not the automorphism group of a \mathcal{G} -free graph.)
- When \mathcal{G} is not an induced subgraph of \mathcal{P}_4 , $\text{Free}(\mathcal{G})$ is structurally complex:
 - $\text{Free}(\mathcal{G})$ is not Σ -small.
 - $\text{Free}(\mathcal{G})$ does not satisfy the computable embeddability condition.
 - $\text{Free}(\mathcal{G})$ is on top for bi-interpretability: Every graph is (computably, uniformly, Σ_1)-bi-interpretable with a \mathcal{G} -free graph.

This lines up with the dichotomies for finite graphs, summarized by the following table.

| | Complexity theory | Combinatorics | Computable structure theory |
|---|--|---|--|
| $\text{Free}(\mathcal{G}) \not\subseteq \text{Free}(\mathcal{P}_4)$ | $\text{Free}(\mathcal{G})^{fin}$ is GI -complete | $\text{Free}(\mathcal{G})^{fin}$ is not a wqo | $\text{Free}(\mathcal{G})$ is on top for eff. bi-int. |
| $\text{Free}(\mathcal{G}) \subseteq \text{Free}(\mathcal{P}_4)$ | GI on $\text{Free}(\mathcal{G})^{fin}$ is in P | $\text{Free}(\mathcal{G})^{fin}$ is a wqo | $\text{Free}(\mathcal{G})$ is Σ -small every \mathcal{H} in $\text{Free}(\mathcal{G})$ has the c.e.c. $\text{Free}(\mathcal{G})$ is not on top for inf. bi-int. |

TABLE 1. Summary of the dichotomies for classes of graphs obtained by forbidding a finite induced subgraph. The first row indicates the areas of study from which the complexity notions under consideration originate. The last two rows represent mutually exclusive alternatives: given a finite graph \mathcal{G} , either $\text{Free}(\mathcal{G})$ satisfies all conditions in the second row, or all conditions in the third row.

We do find a new phenomenon for infinite graphs which does not appear for finite graphs. A corresponding notion to the graph-isomorphism problem, and GI -completeness, for infinite structures comes from Borel reducibility theory. We say that a class of structures \mathbb{C} is on top for Borel reducibility if there is a Borel map Φ such that for all graphs \mathcal{G}, \mathcal{H} ,

$$\mathcal{G} \cong \mathcal{H} \iff \Phi(\mathcal{G}) \cong \Phi(\mathcal{H}).$$

The reader unfamiliar with descriptive set theory should think of a Borel map as a “reasonably definable” map. If \mathbb{C} is on top for Borel reducibility, this means that the isomorphism problem for \mathbb{C} is at least as

¹Note that the graph isomorphism problem is neither known to be NP -complete nor known to lie in P . Babai [Bab16] has shown that graph isomorphism is solvable in quasipolynomial time (see also the expository article [19] and its translation [HBD17] which contains corrections by Babai to the original proof). Consequently, GI is widely believed to be an intermediate class between the two. In this equivalence, and in all subsequent ones concerning GI -completeness, we assume that GI is not in P .

hard as the isomorphism problem for graphs (which is as hard as an isomorphism problem can be). If a class of structures is on top for bi-interpretability, then it is on top for Borel reducibility, but not vice versa. Well-known examples of classes which are on top for Borel reducibility, but not for bi-interpretability, are trees and linear orders. We show that cographs can be added to this list.

Theorem 4.14. *Free(\mathcal{P}_4) is on top for Borel reducibility but not for effective bi-interpretability.*

In particular, out of all classes Free(\mathcal{G}), only Free(\mathcal{P}_4) is on top for Borel reducibility but not on top for effective bi-interpretability.

This phenomenon shows up in the infinite case because deciding whether two infinite trees are isomorphic is very hard (trees are Borel complete) while deciding whether two finite trees are isomorphic is very easy (in linear time by the AHU algorithm). Given a finite cograph \mathcal{G} , the size of its tree \mathcal{T} is polynomial in the number of vertices, and so we get this difference of behaviour. (We note that Friedman and Stanley’s argument in [FS89] that trees are Borel complete is by associating to some graph \mathcal{G} the tree of all finite tuples from that graph. While for a countable graph this yields a countable tree, for a finite graph the size of the tree is exponential in the size of the graph, and so even though determining isomorphism of the trees is linear in the size of the tree it is exponential in the size of the graphs.)

For finite graphs, as far as the authors are aware, whenever the isomorphism problem for a class of graphs is shown to be maximally complicated it is always because there is an interpretation of arbitrary graphs into that class. Is this necessarily the case, or is there, as for infinite graphs, some other strategy for showing that isomorphism on a class is maximally complicated?

Question 1.1. *Is there a collection \mathcal{S} of finite graphs such that Free(\mathcal{S}) is GI-complete, but there is no interpretation from the class of all graphs to Free(\mathcal{S})?*

More generally, we could ask a similar question for more general classes, e.g., for any hereditarily finite class in a relational language.

The paper is organized as follows. In Section 2 we give all the necessary preliminaries and notations regarding (classes of) structures, in particular for labeled trees and graphs. Section 3, after giving the necessary preliminaries around the notion of Σ -smallness, is devoted to prove Theorem 3.12. In addition we will prove that Free(\mathcal{P}_4) is not Σ_2^{inf} -small, i.e., it has uncountably many Σ_2^{inf} -types (Proposition 3.10). We mention that in natural classes of structures, the property of being Σ -small but not Σ_2^{inf} -small has been observed so far only for equivalence structures. Section 4 is devoted to prove Theorem 4.12 and Theorem 4.13.

2. NOTATION AND BACKGROUND

Structures will be denoted by calligraphic letters such as $\mathcal{A}, \mathcal{B}, \mathcal{C}, \dots$, with their corresponding domains being A, B, C, \dots . Embeddings between structures will be denoted via “ \hookrightarrow ”. All of our structures have domain ω and, given a signature τ , we denote by $Mod(\tau)$ the space of τ -structures with domain ω . Notice that such a space is **Polish**, i.e., a separable completely metrizable topological space. The basic clopen sets are determined by atomic and negated atomic formulas.

The structures we will be interested in are labeled trees and graphs. Graphs \mathcal{G} will be irreflexive and undirected. Our notion of subgraph will always be an induced subgraph. We write \mathcal{P}_n for the *path graph* with n vertices (i.e., a graph forming a linear chain of size n), and \mathcal{C}_n for the *cycle graph* with n vertices. We denote by $\overline{\mathcal{G}}$ the complement of \mathcal{G} , i.e., the graph having the same set of vertices and such that two vertices have an edge in $\overline{\mathcal{G}}$ if and only if they do not have an edge in \mathcal{G} . Given two graphs \mathcal{G} and \mathcal{H} the **disconnected union** of \mathcal{G} and \mathcal{H} is defined as the graph consisting of \mathcal{G} and \mathcal{H} with no edge connecting a vertex of \mathcal{G} and a vertex of \mathcal{H} . The **connected union** of \mathcal{G} and \mathcal{H} is defined as the graph consisting of \mathcal{G} and \mathcal{H} and in which every vertex of \mathcal{G} is connected to every vertex of \mathcal{H} . Note that the disconnected union of \mathcal{G} and \mathcal{H} is the complement of the connected union of $\overline{\mathcal{G}}$ and $\overline{\mathcal{H}}$.

As mentioned in the introduction, the \mathcal{P}_4 -free graphs are crucial to this paper. The finite \mathcal{P}_4 -free graphs are exactly the *complement reducible graphs (cographs)*. These are the smallest class of graphs which may be constructed by starting with the single-vertex graphs and closing under connected unions and disconnected unions. For countably infinite \mathcal{P}_4 -free graphs we must also use an operation called by [HPW24] a **sum over a labeled chain** which is defined as follows. Let $(\mathcal{G}_i)_{i \in I}$ for some countable set I be a sequence of graphs. Let \leq be a linear order defined on I and let ℓ be a labelling function from I to $\{0, 1\}$. The labeled sum of $(\mathcal{G}_i)_{i \in I}$ is the graph obtained first performing the disconnected union of $(\mathcal{G}_i)_{i \in I}$ and then, adding an edge

between $v \in \mathcal{G}_i$ and $w \in \mathcal{G}_j$ for $i < j$ if and only if $r(i) = 1$. Following standard convention, we continue to call the infinite \mathcal{P}_4 -free graphs cographs.

Since cographs are constructed recursively from smaller building blocks, we can understand them by a corresponding *decomposition tree* which records the construction. A **tree** $\mathcal{T} = (T, \preceq)$ is a partially ordered set such that for every $v \in T$, $\text{pred}(v) = \{x : x \preceq_{\mathcal{T}} v\}$ is linearly ordered and any two elements have a lower bound. We will call elements of the tree **nodes** and a maximal element of the tree a **leaf**. A tree is a **simple tree** if it has a least element (the **root**) and if every element has an immediate predecessor, in which case we can think of the tree as an acyclic graph with a distinguished element. Every finite tree is a simple tree, and so for finite cographs only simple trees will be required. For infinite cographs we will make use of arbitrary trees.

Given a set of labels L , a *labeled tree* is a tree with a labelling $\ell : T \rightarrow L$ where $L \subseteq \omega$. Given two labeled trees $\mathcal{T}_1, \mathcal{T}_2$ with functions ℓ_1 and ℓ_2 an embedding from \mathcal{T}_1 into \mathcal{T}_2 is given by an injective function f from T_1 to T_2 such that:

- for every $v \in T_1$, $\ell_1(v) \leq \ell_2(f(v))$;
- if $v \preceq_{\mathcal{T}_1} w$ then $f(v) \preceq_{\mathcal{T}_2} f(w)$;
- if $v \preceq_{\mathcal{T}_1} w_1, w_2$ with $w_1 \neq w_2$ then $f(v) \preceq_{\mathcal{T}_2} f(w_1), f(w_2)$ with $f(w_1) \neq f(w_2)$.

In this paper, we will always assume that the order on the labels induces an antichain and hence the first condition becomes “for every $v \in T_1$, $\ell_1(v) = \ell_2(f(v))$ ”.

We begin by describing the decomposition trees of finite cographs since this is the simpler case. Finite cographs are those graphs that can be obtained from the one-vertex graph iterating the operations of connected and disconnected union. This leads to the representation of a cograph \mathcal{G} as a finite $\{0, 1, 2\}$ -labeled tree $\mathcal{T}_{\mathcal{G}}$ describing how \mathcal{G} was constructed. All leaves of $\mathcal{T}_{\mathcal{G}}$ will be labeled by 2 and correspond to the vertices of \mathcal{G} . The nodes of $\mathcal{T}_{\mathcal{G}}$ that are not leaves are labeled by 0 or 1, denoting respectively the operations of disconnected and connected union (of the cographs represented by the subtrees of that node). An immediate consequence of this is that two vertices are connected in \mathcal{G} if and only if their least upper bound, or *meet*, in $\mathcal{T}_{\mathcal{G}}$ is labeled by 1. To guarantee that every cograph \mathcal{G} is determined by a unique decomposition tree $\mathcal{T}_{\mathcal{G}}$ we require that if a vertex $t \in T_{\mathcal{G}}$ is labeled by 1 (0) then all immediate successors of t not being leaves are labeled 0 (1) (see [Dam90, Section 3]). Any cograph has a tree of this form, and given any tree of this form it is the decomposition tree of a graph. It is worth noticing, as it will come in handy later, that every node in a finite decomposition tree that is not a leaf has at least two immediate successors.

For infinite cographs we must account for the operation of sums of labeled chains described previously. We follow [HPW24] which in turn follows [CD08]. We make the following definitions some of which trivialize for simple trees. We say that a tree \mathcal{T} is a **meet-tree** if every pair of elements $x, y \in \mathcal{T}$ have a **meet** $x \wedge y$ which is the closest common ancestor of x and y . Note that the meet of any finite number of elements in a meet-tree is actually the meet of exactly two elements. A meet-tree \mathcal{T} is **ramified** if every element of \mathcal{T} is the meet of a finite set of maximal elements of \mathcal{T} . Given a ramified meet-tree \mathcal{T} , a labelling of \mathcal{T} is a map $\ell : \mathcal{T} \rightarrow \{0, 1, 2\}$ such that the nodes of \mathcal{T} labeled 2 are exactly the leaves. The labelling is **dense** if for any $x \succ y$ non-leaves there is some z with $x \succ z \succeq y$ and $\ell(z) \neq \ell(x)$. Then the decomposition trees are exactly the densely labeled ramified meet-trees. In particular:

- Given a densely labeled ramified meet-tree \mathcal{T} , let $\mathcal{G}(\mathcal{T})$ be the graph whose vertices are the leaves of \mathcal{T} , with two vertices u and v being joined by an edge if and only if $u \wedge v$ is labeled 1.
- Given a cograph \mathcal{G} , the **decomposition tree** $\mathcal{T}(\mathcal{G})$ of \mathcal{G} is the tree whose nodes are $R(\mathcal{G})$, the **robust modules** of \mathcal{G} .² The trivial modules—which consist of a single vertex—get the label 2 and are the leaves of \mathcal{T} . If $A \subseteq G$ is the least robust module of \mathcal{G} containing u and v , then we give A the label 1 if there is an edge between u and v , and 0 if there is no such edge; this does not depend on the choice of u and v .

²We refer the reader to [HPW24] for more details here, but recall briefly the definitions. A **module** of a graph \mathcal{G} is any subset $A \subseteq G$ such that for all $u, v \in A$ and $w \notin A$, there is an edge between u and w if and only if there is one between v and w . A module is **strong** if it is either comparable to or disjoint from every module. A module is **robust** if it is the least strong module containing two vertices of \mathcal{G} . For this paper it mostly suffices to note that robust modules are graph-theoretically defined and the property of being a robust module is maintained by isomorphisms.

Theorem 6.19 of [HPW24] proves that these constructions give a one-to-one correspondence between cographs and ramified meet-trees densely labeled by $\{0, 1, 2\}$. In Theorem 4.10 we show that they are in fact bi-interpretable in infinitary logic. This is somewhat implicit in [CD08] though Courcelle and Delhommé use second-order logic. In particular, an automorphism of the decomposition tree decomposition induces an automorphism of the cograph by restricting to the leaves, and an automorphism of the cograph induces an automorphism of its tree by acting on the leaves and extending to the whole tree in a unique way.

3. $\text{Free}(\mathcal{G})$ THAT ARE STRUCTURALLY SIMPLE

Recall that the **existential type** of a tuple \bar{a} in a structure \mathcal{A} is the set

$$\exists\text{-tp}_{\mathcal{A}}(\bar{a}) = \{\varphi : \varphi(\bar{x}) \text{ is a first-order existential formula with } \mathcal{A} \models \varphi(\bar{a})\}.$$

A first-order type p is realized in a structure \mathcal{A} if there is a tuple \bar{a} such that for all $\varphi \in p$, $\mathcal{A} \models \varphi(\bar{a})$, and we say that p is **realized** in a class of structures \mathbb{K} if there is a structure in \mathbb{K} realizing p . Furthermore, an existential type p is **sharply realized** in \mathbb{K} if there exists a tuple \bar{a} in some structure $\mathcal{A} \in \mathbb{K}$ so that $p = \exists\text{-tp}_{\mathcal{A}}(\bar{a})$, and it is **sharply realized** in \mathcal{A} if it is sharply realized in $\mathbb{K} = \{\mathcal{A}\}$.

The following definition first appeared under the name recursive embeddability condition in [Ric81].

Definition 3.1. A structure \mathcal{A} satisfies the **computable embeddability condition (c.e.c)** if each sharply realized existential type realized in \mathcal{A} is computable. A class \mathbb{K} satisfies the computable embeddability condition if every structure \mathbb{K} satisfies it.

The following classes have the computable embeddability condition: linear orderings, trees ([Ric81]), Boolean algebras ([HM12]), \mathbb{Q} -vector spaces, algebraically closed fields, differentially closed fields, abelian p -groups, and equivalence structures.

Montalbán [Mon12] defined the following non-effective version of the c.e.c. Though his definition was in terms of universal types, we state it here in terms of existential types. The two definitions are equivalent.

Definition 3.2. A class \mathbb{K} of structures is **Σ -small** if there are at most countably many existential types sharply realized in \mathbb{K} .

It is an immediate observation that a class of structures is Σ -small if and only if there is an $X \in 2^\omega$ so that all structures in \mathbb{K} have the computable embeddability condition relative to X .

Any three element connected induced subgraph of a graph in $\text{Free}(\mathcal{P}_3)$ must be complete and thus the edge relation on such a graph must be transitive. From this it follows that graphs in $\text{Free}(\mathcal{P}_3)$ are just disconnected unions of cliques. It also shows that graphs in $\text{Free}(\mathcal{P}_3)$ are definitionally equivalent with equivalence structures (i.e., we can define an equivalence relation agreeing with the cliques in the graphs using quantifier-free formulas, and vice versa, define graphs from equivalence relations). Using this equivalence we get the following from results about equivalence structures in the literature [Mon21, Section X].

Proposition 3.3. *Every graph in $\text{Free}(\mathcal{P}_3)$ (i.e., equivalence structure) has the computable embeddability condition. Thus, $\text{Free}(\mathcal{P}_3)$ is Σ -small.*

We are going to prove the same result for $\text{Free}(\mathcal{P}_4)$. To do so, we use a technique that appeared in [Mon21, Chapter X.2] and is based on [Ric81]. Let τ be a finite vocabulary without function symbols, and let \mathbb{K} be a class of τ -structures. Given a set A , define τ_A by increasing the vocabulary τ with new constant symbols, one for each element of A . Then, a τ_A -structure is determined by a τ -structure \mathcal{B} and a map $f : A \rightarrow \mathcal{B}$ describing the assignments of the new constants. We denote by \mathbb{K}^{fin} the set of finite substructures in \mathbb{K} and, given some $\mathcal{A} \in \mathbb{K}^{fin}$, let

$$\mathbb{K}_{\mathcal{A}} = \{\mathcal{B} \in \mathbb{K} : \mathcal{B} \text{ is a } \tau_A\text{-structure and } \mathcal{A} \xrightarrow{g} \mathcal{B} \text{ for some } \tau_A\text{-embedding } g\}.$$

Recall that a **well quasi-order** (in short, **wqo**) is a quasi-order without infinite antichains and infinite descending chains. We use the following sufficient condition for a class to be Σ -small.

Theorem 3.4 ([Mon21, Theorem 10.2.2]). *Let \mathbb{K} be a class of structures in finite vocabulary without function symbols. Suppose that for every finite substructure $\mathcal{A} \in \mathbb{K}^{fin}$, $\mathbb{K}_{\mathcal{A}}^{fin}$ is a wqo under the embeddability relation. Then, in \mathbb{K} , every Σ_1^{inf} formula is equivalent to a finitary existential formula. In particular, \mathbb{K} is Σ -small.*

To prove that such a class is Σ -small we want to apply Theorem 3.4. To do so, first notice that, since decomposition trees are just labeled trees, that they form a well quasi-order was already established by Kruskal [Kru60].

Theorem 3.5 (Kruskal's tree theorem). *Let L be a well quasi-ordered set of labels. The collection of finite L -labeled rooted trees ordered by embeddability is a wqo.*

With some further effort, the result above allows us to prove Theorem 3.4 for the class of decomposition trees, which we denote by \mathbb{DT} for the remainder of this section.

Lemma 3.6. *For any finite decomposition tree \mathcal{T} , $\mathbb{DT}_{\mathcal{T}}^{fin}$ is a wqo. Hence, the class of decomposition trees is Σ -small.*

Proof. Given a finite $\text{Free}(\mathcal{P}_4)$ graph \mathcal{G} with set of constants A , consider its decomposition tree \mathcal{T} with set of constants A and labelling function ℓ .

Given $\mathcal{S} \in \mathbb{DT}_{\mathcal{T}}^{fin}$, consider the tree $\mathcal{S}^A = (S^A, \ell^A)$ defined as follows. The tree \mathcal{S}^A is \mathcal{S} but without constants and with a different labelling function. Namely, we define $\text{range}(\ell^A) = \{0, 1, 2\} \cup \{a + 2 : a \in A\}$ and we let $\ell^A(v) = a + 2$ if $a^{\mathcal{T}} = v$ and $\ell^A(v) = \ell(v)$ otherwise. Notice that \mathcal{S}^A is just a finite labeled tree, hence, by Theorem 3.5, the set $\{\mathcal{S}^A : \mathcal{S} \in \mathbb{DT}_{\mathcal{T}}^{fin}\}$ ordered by embeddability is a wqo.

By definition of embedding between labeled trees and the definition of \mathcal{S}^A (and in particular of ℓ^A) we have that for $\mathcal{S}_0, \mathcal{S}_1 \in \mathbb{DT}_{\mathcal{T}}^{fin}$, \mathcal{S}_0 embeds into \mathcal{S}_1 if and only if \mathcal{S}_0^A embeds into \mathcal{S}_1^A . Hence, also $\mathbb{DT}_{\mathcal{T}}^{fin}$ ordered by embeddability is a wqo. \square

To apply Theorem 3.4 and get the result for $\text{Free}(\mathcal{P}_4)$, we need to use the announced interplay between decomposition trees and $\text{Free}(\mathcal{P}_4)$.

Lemma 3.7. *If \mathcal{G} and \mathcal{G}' are finite cographs with decomposition trees \mathcal{T} and \mathcal{T}' respectively, then \mathcal{G} is an induced subgraph of \mathcal{G}' if and only if \mathcal{T} embeds as a partial order into \mathcal{T}' preserving labels³ and preserving labels of meets⁴. The same holds even if we assume that graphs possibly have constants and decomposition trees possibly have constants on leaves.*

Proof. The right-to-left direction is [Dam90, Lemma 1]. Notice that the author assumes that the labels in a decomposition tree are such that $0, 1 < 2$ and 0 and 1 are incomparable. On the other hand, after [Dam90, Theorem 4], the author also notices that the proof of [Dam90, Lemma 1] still works (and actually is even simpler) if one assumes (like we do) that $0, 1$ and 2 are incomparable each other. Indeed, assume that \mathcal{T} embeds into \mathcal{T}' preserving meets via some function ι , and let ℓ and ℓ' be corresponding labelling functions. Notice that ι is an embedding between labeled trees and $0, 1$ and 2 are incomparable each other. Hence for every $x \in \mathcal{T}$ and for every $i \leq 2$, $\ell(v) = \ell'(\iota(v))$. Combining this with the hypothesis that it also preserve meets we obtain that, for every $x, y \in \mathcal{T}$ such that $\ell(x) = \ell(y) = 2$ (i.e., x and y are vertices of $V(\mathcal{G})$), $\ell(x \wedge y) = \ell'(\iota(x) \wedge \iota(y))$. Hence, (x, y) is an edge in \mathcal{G} if and only if $(\iota(x), \iota(y))$ is an edge in \mathcal{G}' , i.e., \mathcal{G} is an induced subgraph of \mathcal{G}' .

For the left-to-right direction, we generalize proof ideas from [CLB81, Lemma 1]. Observe that any induced subgraph of a graph can be obtained by successively removing vertices from the larger graph. Therefore, it is enough for us to prove the direction for when $\mathcal{G} = \mathcal{G}' - \{v\}$, where v is a vertex in \mathcal{G}' . If $|\mathcal{G}'| \leq 1$ the claim follows immediately. Otherwise, let $p(v)$ denote the parent of the node of v in \mathcal{T}' , which must exist since $|\mathcal{G}'| \geq 2$. \mathcal{T} can be obtained from \mathcal{T}' as follows:

- (i) If $p(v)$ has more than two children just remove v from \mathcal{T}' .
- (ii) Otherwise, if $p(v)$ has exactly two children and $p(v)$ is the root, remove $v, p(v)$ from \mathcal{T}' , making the unique sibling of v the new root.
- (iii) Otherwise, $p(v)$ has exactly two children and is not the root. Then $p(v)$ has a parent $p(p(v))$ and v has a unique sibling v' . We first remove $v, p(v)$ from \mathcal{T}' , then we *lift* v' up to $p(p(v))$ as follows:
 - (a) if v' is a leaf let its new parent be $p(p(v))$;
 - (b) if v' is not a leaf, remove v' and for every child of v' , let $p(p(v))$ be its new parent.

³ $(\forall \sigma \in \mathcal{T}) [\ell(\sigma) = \ell'(\sigma)]$.

⁴ $(\forall v, w \in \mathcal{G}) [\ell(v \wedge w) = \ell'(v \wedge w)]$.

Letting ℓ' denote the labelling function of \mathcal{T}' , we define the labelling function ℓ of \mathcal{T} by $\ell'(w) = \ell(w)$ for $w \in \mathcal{T}$. Then the identity map embeds \mathcal{T} in \mathcal{T}' as a partial order. Furthermore, in all cases, labels and meet labels are preserved, because in the hardest case (iii) where an internal node $p(v)$ is removed, the affected node v' will be lifted to its grandparent, preserving its label since the labels of internal nodes must alternate between 0 and 1 by label density. Therefore, \mathcal{T} is the decomposition tree of \mathcal{G} .

In case we assume that graphs have constants, we apply the same strategy of Lemma 3.6. Namely, assume that \mathcal{G} and \mathcal{G}' have sets of constants A and B respectively. Instead of considering \mathcal{T} and \mathcal{T}' , we consider the labeled trees \mathcal{S} and \mathcal{S}' isomorphic respectively to \mathcal{T} and \mathcal{T}' but with labelling functions ℓ^A and ℓ^B such that $\text{range}(\ell^A) = \{0, 1, 2\} \cup \{a + 2 : a \in A\}$ and $\text{range}(\ell^B) = \{0, 1, 2\} \cup \{b + 2 : b \in B\}$ and in which all labels are incomparable each other. The labelling functions are defined letting $\ell^A(v) = a + 2$ if $a^{\mathcal{G}} = v$ and $\ell^A(v) = \ell(v)$ otherwise and $\ell^B(v) = b + 2$ if $b^{\mathcal{G}'} = v$ and $\ell^B(v) = \ell'(v)$ otherwise. With the same proofs as above, one can verify that \mathcal{G} is an induced subgraph of \mathcal{G}' if and only if \mathcal{S} embeds into \mathcal{S}' preserving meets. \square

Theorem 3.8. *The class $\text{Free}(\mathcal{P}_4)$ is Σ -small.*

Proof. Let $\mathcal{G} \in \text{Free}(\mathcal{P}_4)$ and consider $\mathcal{T}_{\mathcal{G}}$. By Lemma 3.6, $\mathbb{DT}_{\mathcal{T}_{\mathcal{G}}}^{\text{fin}}$ is a wqo. Since $\mathcal{T}_{\mathcal{G}}$ comes from \mathcal{G} , it only has constants on leaves and hence we can apply Lemma 3.7 obtaining that $\text{Free}(\mathcal{P}_4)_{\mathcal{G}}^{\text{fin}}$ is a wqo as well. \square

The proof of the theorem above nicely exploits the relationships between \mathcal{P}_4 -free graphs and decomposition trees. On the other hand, the same result could have been obtained using a more general fact without exploiting these relationships. Indeed, [AL15, Theorem 2] proves that certain classes of graphs (included $\text{Free}(\mathcal{P}_4)$) are a wqo under the *labeled induced subgraph relation*. To define this notion, we consider an arbitrary quasiorder (W, \leq) . Every vertex v of a graph \mathcal{G} has a label $l(v) \in W$ and \mathcal{G} is a labeled induced subgraph of \mathcal{G}' if \mathcal{G} is isomorphic to an induced subgraph of \mathcal{G}' and the isomorphism maps each $v \in V(\mathcal{G})$ to some $w \in V(\mathcal{G}')$ with $l(v) \leq l(w)$. What we have noted both in the proof of Lemma 3.6 and Lemma 3.7 boils down to say that a graph with constants can be regarded as a labeled graph in which all labels are incomparable. Hence, one could have used [AL15, Theorem 2] to prove Theorem 3.8, showing directly that for any $\mathcal{G} \in \text{Free}(\mathcal{P}_4)$, $\text{Free}(\mathcal{P}_4)_{\mathcal{G}}^{\text{fin}}$ is a wqo.

From the lemmas above we also obtain that any graph in $\text{Free}(\mathcal{P}_4)$ has the computable embeddability condition.

Theorem 3.9. *Let \mathbb{K} be such that for every $\mathcal{A} \in \mathbb{K}$, $\mathbb{K}_{\mathcal{A}}^{\text{fin}}$ is a wqo. Then, every graph in $\mathcal{A} \in \mathbb{K}$ has the computable embeddability condition.*

In particular every $\mathcal{G} \in \text{Free}(\mathcal{P}_4)$ has the computable embeddability condition.

Proof. Let $\mathcal{A} \in \mathbb{K}$ and $\bar{a} \in A^{<\omega}$ and let \mathcal{B} be such that $\mathcal{B} \hookrightarrow \mathcal{A}$ on elements \bar{a} . We have to show that $\exists\text{-tp}_{\mathcal{A}}(\bar{a})$ is computable. Without loss of generality and for readability we can assume that the vocabulary of the structures in \mathbb{K} consists of a binary relational symbol E . We begin by showing that the existential type describing all possible finite extensions of \mathcal{B} in \mathbb{K} , i.e., $\mathbb{K}_{\mathcal{B}}^{\text{fin}}$. The quasi-ordering $\mathbb{K}_{\mathcal{F}}$ contains graphs on arbitrary finite domains and thus neither membership nor embeddability is decidable in $\mathbb{K}_{\mathcal{B}}$. However, if we restrict ourselves to structures whose universes are—besides elements of \mathcal{B} —initial segments of ω , then indeed the two relations become decidable, i.e., for two finite structures \mathcal{H} and \mathcal{J} with such universes, we can decide whether $\mathcal{H} \in \mathbb{K}_{\mathcal{B}}$ and \mathcal{H} embeds into \mathcal{J} . As we will see, considering this restriction is sufficient to show that $\exists\text{-tp}_{\mathcal{A}}(\bar{a})$ is computable. Let $N_{\mathcal{B}}$ be the set of such structures, and for $\mathcal{H} \in \mathbb{K}_{\mathcal{B}}$ let i_0, \dots enumerate the elements of $H \setminus B$ in ascending order, define

$$\varphi_{\mathcal{H}} = \exists x_0 \dots \exists x_{|H \setminus B|} \bigwedge_{v \in B} \bigwedge_{k: v E i_k} v E x_k \wedge \bigwedge_{k: v \not E i_k} v \not E x_k \wedge \bigwedge_{(k,l): i_k E i_l} x_k E x_l \wedge \bigwedge_{(k,l): i_k \not E i_l} x_k \not E x_l$$

and let $X_N = \{\varphi_{\mathcal{H}} : \mathcal{H} \in N_{\mathcal{B}}\}$. The set X_N is decidable and equivalent to $X = \{\varphi_{\mathcal{H}} : \mathcal{H} \in \text{Free}(\mathcal{P}_4)_{\mathcal{B}}^{\text{fin}}\}$ up to renaming of variables. In particular, we can now see that X is computable. For $\psi \in X$ in prenex normal form with variable symbols x_{i_0}, \dots we have that $\psi \in X$ if and only if $\psi[x_{i_0}/x_0][x_{i_1}/x_1] \dots \in X_N$. Now, X is the existential type describing all possible finite extensions of \bar{a} in \mathbb{K} , and we can in similar fashion to the argument above obtain the existential type of \bar{a} in \mathcal{A} . To see that it is computable, first note that $\mathcal{A}_{\mathcal{B}}^{\text{fin}} \subseteq \mathbb{K}_{\mathcal{B}}$, the quasi-ordering consisting of all finite graphs that embed into \mathcal{A} and extend \mathcal{B} , is closed

downwards. Let S be the basis of its compliment, i.e., $\mathcal{H} \in \mathbb{K}_{\mathcal{B}} \setminus \mathcal{A}_{\mathcal{B}}^{fin}$ such that $\mathcal{H} \hookrightarrow \mathcal{J}$ implies $\mathcal{J} \not\hookrightarrow \mathcal{A}$. As $\mathbb{K}_{\mathcal{B}}$ is a wqo, S contains only a finite number of equivalence classes; let J_0, \dots, J_n enumerate representatives of these classes. Then the set

$$\{\varphi_{\mathcal{H}} : \mathcal{H} \in N_{\mathcal{B}} \wedge (\forall k \leq n) J_k \not\hookrightarrow \mathcal{H}\} \subseteq X_N$$

is computable and corresponds to $\exists\text{-tp}_{\mathcal{A}}(\bar{a})$ up to renaming of variables. \square

A natural question is whether the class of cographs is not only Σ -small but also small for types of higher complexities. For example, say that a class \mathbb{K} is Σ_2^{inf} -**small**, if there are only countably many different complete Σ_2^{inf} -types sharply realized in \mathbb{K} . Most examples of classes that are Σ -small, such as Boolean algebras and linear orderings [HM12; Mon12], are also Σ_2^{inf} -small. The only natural class in the literature that is known to be Σ -small but not Σ_2^{inf} -small are equivalence structures [Mon12]. They are joined by the class of cographs.

Proposition 3.10. *Free(\mathcal{P}_3) is not Σ_2^{inf} -small. In particular, Free(\mathcal{P}_4) is not Σ_2^{inf} -small as well.*

Proof. Recall that Free(\mathcal{P}_3) can be viewed as equivalence structures and the latter is not Σ_2^{inf} -small [Mon12, Section 4.2]. Since \mathcal{P}_3 is an induced subgraph of \mathcal{P}_4 we have Free(\mathcal{P}_3) \subseteq Free(\mathcal{P}_4) and hence, by the fact that the property of being Σ_2^{inf} -small is closed under subsets, Free(\mathcal{P}_4) cannot be Σ_2^{inf} -small. \square

The main result of this section is that Free(\mathcal{P}_4) is a dividing line for being Σ -small and having the computable embeddability condition.

The following observation can be proved by chasing definitions.

Lemma 3.11. *A finite graph \mathcal{G} is an induced subgraph of finite graph \mathcal{H} if and only if Free(\mathcal{G}) \subseteq Free(\mathcal{H}).*

Theorem 3.12. *For a finite graph \mathcal{G} , the following are equivalent:*

- (i) every graph in Free(\mathcal{G}) has the computable embeddability condition;
- (ii) Free(\mathcal{G}) is Σ -small;
- (iii) Free(\mathcal{G}) \subseteq Free(\mathcal{P}_4).

Proof. That (i) implies (ii) is trivial.

The fact that (iii) implies (i) follows from Lemma 3.11 and the fact that the property of being Σ -small is closed under subsets. To conclude the proof, we show that (ii) implies (iii) proving its contrapositive. By Lemma 3.11 we have that \mathcal{G} is not an induced subgraph of \mathcal{P}_4 . It is well-known in graph theory that if \mathcal{P}_4 is not an induced subgraph of \mathcal{G} then either \mathcal{G} or $\bar{\mathcal{G}}$ contain a cycle (see e.g., the first theorem of Section 4.7 in [BC79]). Hence, take the maximum m such that \mathcal{C}_m is an induced subgraph of \mathcal{G} or $\bar{\mathcal{C}}_m$ is an induced subgraph of \mathcal{G} .

Damaschke [Dam90] showed that the classes Free(\mathcal{G}) for such graphs \mathcal{G} are not wqos. Indeed, $\{\mathcal{C}_{m+1+n} : n \in \omega\}$ is an antichain in Free(\mathcal{G}) if \mathcal{C}_m is an induced subgraph of \mathcal{G} and $\{\bar{\mathcal{C}}_{m+n+1} : n \in \omega\}$ is an antichain if $\bar{\mathcal{C}}_m$ is an induced subgraph of \mathcal{G} . For every i , let φ_i (respectively ψ_i) be the existential formula saying that there is an induced subgraph isomorphic to \mathcal{C}_{m+1+i} (respectively $\bar{\mathcal{C}}_{m+1+i}$).

For every $I \subseteq \omega$:

- if \mathcal{C}_m is an induced subgraph of \mathcal{G} , let \mathcal{A}_I be the graph in Free(\mathcal{G}) defined as the disconnected union of all the \mathcal{C}_{m+1+i} 's for $i \in I$.
- if $\bar{\mathcal{C}}_m$ is an induced subgraph of \mathcal{G} , let \mathcal{B}_I be the graph in Free(\mathcal{G}) defined as the connected union of all the $\bar{\mathcal{C}}_{m+1+i}$'s for $i \in I$.

Notice that $\mathcal{A}_I \models \varphi_i$ if and only if $i \in I$ (similarly for \mathcal{B}_I). Also, $\exists\text{-tp}_{\mathcal{A}_I}(\langle \rangle) \supseteq \{\varphi_i : i \in I\}$ and $\exists\text{-tp}_{\mathcal{B}_I}(\langle \rangle) \supseteq \{\psi_i : i \in I\}$ where $\langle \rangle$ denotes the empty tuple. To conclude, notice that

- if \mathcal{C}_m is an induced subgraph of \mathcal{G} , $\{\exists\text{-tp}_{\mathcal{A}_I}(\bar{a}) : \mathcal{A} \in \text{Free}(\mathcal{G}) \wedge \bar{a} \in \mathcal{A}^{<\omega}\} \supseteq \{\exists\text{-tp}_{\mathcal{A}_I}(\langle \rangle) : I \subseteq \omega\}$ and
- if $\bar{\mathcal{C}}_m$ is an induced subgraph of \mathcal{G} , $\{\exists\text{-tp}_{\mathcal{B}_I}(\bar{a}) : \mathcal{B} \in \text{Free}(\mathcal{G}) \wedge \bar{a} \in \mathcal{B}^{<\omega}\} \supseteq \{\exists\text{-tp}_{\mathcal{B}_I}(\langle \rangle) : I \subseteq \omega\}$.

Since there are uncountably many I 's, it follows that there are uncountably many existential types, proving that Free(\mathcal{G}) is not Σ -small. \square

Most natural classes that are known to be Σ -small allow for a finite set of Σ_1^{inf} or Π_1^{inf} relations so that after adding these relations to the structures, every Π_1^{inf} -definable relation becomes Σ_1^{inf} -definable. These augmented structures are often called the ‘‘jump of the structure’’. For example, for linear orderings one

obtains the jump by adding the successor relation and for Boolean algebras the jump is obtained by adding the atom relation.

Question 3.13. *Which canonical set of relations needs to be added to obtain the jump of $\text{Free}(\mathcal{P}_4)$?*

4. $\text{Free}(\mathcal{G})$ THAT ARE STRUCTURALLY RICH

The following definitions are standard in descriptive set theory, see e.g., [Gao09; Kec95]. For a Polish space X , the class of **Borel sets** is given by the smallest σ -algebra containing all open subsets of X . Given two Polish spaces X, Y , a function $f : X \rightarrow Y$ is **Borel** if for every open set $U \subseteq Y$, $f^{-1}(U)$ is Borel.

Definition 4.1 ([FS89]). Given two classes closed under isomorphism $\mathbb{C} \subseteq \text{Mod}(\tau)$ and $\mathbb{D} \subseteq \text{Mod}(\tau')$ for two languages τ and τ' we will say that \mathbb{C} is **Borel reducible** to \mathbb{D} if the isomorphism relation on \mathbb{C} is Borel reducible to the isomorphism relation on \mathbb{D} .

Definition 4.2. A class of structures \mathbb{C} is **on top for Borel reducibility** if for every other class of structures \mathbb{D} in some vocabulary τ , \mathbb{D} Borel reduces to \mathbb{C} .

Classes on top for Borel reducibility are often called S_∞ -complete in the literature as every equivalence relation induced by a Borel action of a closed subgroup of S_∞ reduces to it. Another, more unfortunate name, is simply Borel complete.

Theorem 4.3. *$\text{Free}(\mathcal{P}_4)$ is on top for Borel reducibility.*

Proof. To prove the claim we reduce the class of trees (which is well-known to be on top for Borel reducibility, see [FS89]) to the subclass of infinite *simple* decomposition trees (recalling that these are the rooted decomposition trees such that every node has an immediate predecessor). Then we will easily see that it is Borel to produce from a decomposition tree the corresponding cograph.

Given a tree \mathcal{T} , we produce a tree \mathcal{T}_ℓ adding to every node of \mathcal{T} infinitely many new children each of which is a leaf. The root of \mathcal{T}_ℓ is labeled by 0 and for every $v \in \mathcal{T}_\ell$, if v is a leaf, the label of v is 2 while if v is not a leaf and has odd (even) distance from the root, the label of v is 0 (1). The construction implies that \mathcal{T}_ℓ is an infinite labeled tree. Because each non-leaf node of \mathcal{T}_ℓ has at least two leaf children, \mathcal{T}_ℓ is ramified. The definition of the labels ensures that the labelling is dense. Thus \mathcal{T}_ℓ is the decomposition tree of an infinite cograph.

We argue that $\mathcal{T} \cong \mathcal{T}'$ if and only if $\mathcal{T}_\ell \cong \mathcal{T}'_\ell$. If $\mathcal{T} \cong \mathcal{T}'$, we can extend the isomorphism witnessing $\mathcal{T} \cong \mathcal{T}'$ to take care of the new leaves that have been added in the obvious way. If $\mathcal{T}_\ell \cong \mathcal{T}'_\ell$ we can just remove all of the leaves to obtain an isomorphism $\mathcal{T} \cong \mathcal{T}'$.

The standard way to produce from a tree the corresponding cograph is Borel. Recall that this was done as follows. We first identify the leaves which we take as the vertices of the cograph. Given vertices u, v , we can find in the tree the meet $u \wedge v$ of u and v , and set an edge between u and v if and only if $u \wedge v$ is labeled 1. \square

Countable trees and graphs are both on top for Borel reducibility. However, as can be seen from work of Gao [Gao01], graphs are structurally much richer than trees. The Borel reduction of some class \mathbb{C} into the class of graphs can be facilitated by the classical first-order interpretation of arbitrary structures in graphs, see e.g. [Mar02]. However, the reduction given in [FS89] that shows that the class of trees is on top for Borel reducibility is much weaker. Indeed, Gao showed that trees are not “faithfully Borel complete” [Gao01]. Faithful Borel reductions can be seen as a weak form of model-theoretic interpretation, and as indeed every model-theoretic interpretation induces a faithful Borel reduction, this highlights a difference in the structural richness of the two classes.

The strongest notion of reducibility is that of reductions by effective bi-interpretations. If a class \mathbb{C} is reducible by effective bi-interpretations to a class \mathbb{D} then this means that essentially any computability-theoretic behaviour that can be found in the class \mathbb{C} can also be found in \mathbb{D} . It was shown in [Har+17] that a structure \mathcal{A} being effectively interpretable in \mathcal{B} is equivalent to there being a computable functor from the isomorphism class of \mathcal{B} to \mathcal{A} (and a similar characterization of bi-interpretability in terms of a computable equivalence of categories). We will use this semantic approach in this paper. For an extensive treatment of bi-interpretability, including a proof of the equivalence mentioned above we refer the reader to [Mon21; Mon25].

The following definitions first appeared in [Har+17]. Some definitions are taken from [Ros22] but are equivalent to definitions that can be found in [Har+17].

Definition 4.4. A **computable functor** $F : \mathbb{C} \rightarrow \mathbb{D}$ is witnessed by two computable operators Φ and Φ_* such that

- for every $\hat{\mathcal{A}} \in \mathbb{C}$, $\Phi^{D(\hat{\mathcal{A}})}$ is the atomic diagram of $F(\hat{\mathcal{A}}) \in \mathbb{D}$
- for every isomorphism $f : \hat{\mathcal{A}} \rightarrow \tilde{\mathcal{A}}$ in \mathbb{C} , $\Phi_*^{\hat{\mathcal{A}} \oplus f \oplus \tilde{\mathcal{A}}} = F(f)$.

In order to define a semantic notion of reduction equivalent to effective bi-interpretability we need a few more category theoretic definitions.

Definition 4.5 ([Har+17]). A functor $F : \mathbb{C} \rightarrow \mathbb{D}$ is **effectively isomorphic** to $G : \mathbb{C} \rightarrow \mathbb{D}$ if there is a Turing operator Λ such that for every $\mathcal{A} \in \mathbb{C}$, $\Lambda^{\mathcal{A}}$ is an isomorphism from $F(\mathcal{A})$ to $G(\mathcal{A})$, and the following diagram commutes for all $\mathcal{A}, \mathcal{B} \in \mathbb{C}$ and every $\gamma \in \text{Hom}(\mathcal{A}, \mathcal{B})$.

$$\begin{array}{ccc} F(\mathcal{A}) & \xrightarrow{\Lambda^{\mathcal{A}}} & G(\mathcal{A}) \\ \downarrow F(\gamma) & & \downarrow G(\gamma) \\ F(\mathcal{B}) & \xrightarrow{\Lambda^{\mathcal{B}}} & G(\mathcal{B}) \end{array}$$

The following definition appeared for arbitrary categories in [Ros22]. Here we only deal with classes of structures where the arrows are given by the isomorphism relation.

Definition 4.6. We say that \mathbb{C} is **CBF-reducible** to \mathbb{D} if

- (1) there is a computable functor $F : \mathbb{C} \rightarrow \mathbb{D}$ and a computable functor $G : \mathbb{D} \supseteq \hat{\mathbb{D}} \rightarrow \mathbb{C}$ where $\hat{\mathbb{D}}$ is the isomorphism closure of $F(\mathbb{C})$,
- (2) $F \circ G$ is effectively isomorphic to $Id_{\hat{\mathbb{D}}}$ and $G \circ F$ is effectively isomorphic to $Id_{\mathbb{C}}$,
- (3) and, if $\Lambda_{\mathbb{C}}, \Lambda_{\mathbb{D}}$ are the operators witnessing the effective isomorphism between $G \circ F$ and $Id_{\mathbb{C}}$, respectively, $F \circ G$ and $Id_{\hat{\mathbb{D}}}$, then for every $\mathcal{A} \in \mathbb{C}$, $F(\Lambda_{\mathbb{C}}^{\mathcal{A}}) = \Lambda_{\mathbb{D}}^{F(\mathcal{A})} : F(\mathcal{A}) \rightarrow F(G(F(\mathcal{A})))$ and every $\mathcal{B} \in \hat{\mathbb{D}}$, $G(\Lambda_{\mathbb{D}}^{\mathcal{B}}) = \Lambda_{\mathbb{C}}^{G(\mathcal{B})} : G(\mathcal{B}) \rightarrow G(F(G(\mathcal{B})))$.

Functors satisfying Items 2 and 3 are said to be **effective pseudo-inverses**.

The main result of [Har+17] shows that CBF-reducibility is equivalent to reduction via effective bi-interpretability. We obtain the following notion of universality for this notion.

Definition 4.7. A class of structures \mathbb{C} is **on top for effective bi-interpretability** if every Borel class of countable structures CBF-reduces to \mathbb{C} .

If we replace the Turing operators in Definitions 4.4 to 4.6 with Borel operators we obtain the notions of **Borel functors** and **BBF-reducibility**. In [HMM18] it was shown that BBF-reducibility is equivalent to reductions via infinitary bi-interpretability, and thus we can define the notion of being **on top for infinitary bi-interpretability** analogous to Definition 4.7. If two structures are infinitary bi-interpretable, then their automorphism groups must be isomorphic [HMM18, Theorem 6]. Thus, if class \mathbb{C} is on top for infinitary bi-interpretability, then every automorphism group of countable structures must be found among the automorphism groups of structures in \mathbb{C} . We will use this fact to show that the class of cographs is not on top for infinitary bi-interpretability (Theorem 4.11).

But first, we show that the class of cographs and cotrees are BBF-equivalent (Theorem 4.10). Using Lemmas 4.8 and 4.9, we show that a cograph and its decomposition tree are not only infinitary bi-interpretable but also bi-interpretable in the standard model-theoretic sense (see [Hod93, Chapter 5]) which is more restrictive. This suggests that cographs are not even on top for faithful Borel reducibility. (Gao [Gao01] has only shown that simple trees and linear orders are not on top for faithful Borel reducibility, but we expect that this extends to all trees and, we expect, to densely labeled ramified meet-trees.)

To begin, we must show that certain relations on cographs are definable. Given distinct vertices u, v in a graph, let $M(u, v)$ denote the least module containing u and v . We can define $M(u, v)$ by a closure operation: It is the smallest set M containing u and v and such that if $x, y \in M$ and there is an edge from w to x and no edge from w to y then $w \in M$. We say that w **witnesses that $\{u, v\}$ is not a module** if there is an edge from w to one of u or v and no edge from w to the other. Lemma 4.8 shows that in a cograph the closure can be done in two steps, allowing us to express $w \in M(u, v)$ in first-order logic.

Lemma 4.8. *Let \mathcal{G} be a cograph and let u, v be distinct vertices in \mathcal{G} . Then $w \in M(u, v)$ if and only if either:*

- (1) $w = u$ or $w = v$,
- (2) w witnesses that $\{u, v\}$ is not a module, or
- (3) there is $w' \neq u, v$ such that w' witnesses that $\{u, v\}$ is not a module and w witnesses that either $\{u, w'\}$ or $\{v, w'\}$ is not a module.

Therefore $w \in M(u, v)$ can be expressed using an \exists_1 formula of finitary first-order logic. Moreover, we will show that

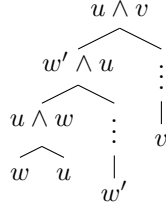
$$(*) \quad M(u, v) = \{w : w \wedge u \succ u \wedge v\} \cup \{w : w \wedge v \succ u \wedge v\}.$$

Proof. One direction—that if w satisfies (a), (b), or (c) then $w \in M(u, v)$ —is clear. We must prove the other direction. One way to argue this is to verify by case analysis that if w enters $M(u, v)$ in three steps, but not in two steps, then there must be an induced subgraph isomorphic to \mathcal{P}_4 . We will take a more structural approach via the decomposition tree.

Now we will show that

$$\{w : w \wedge u \succ u \wedge v\} \cup \{w : w \wedge v \succ u \wedge v\} \subseteq M(u, v).$$

If $w = u$ or $w = v$ then by (1) we have $w \in M(u, v)$. Otherwise, suppose that $w \wedge u \succ u \wedge v$. Then $w \wedge v = u \wedge v$. Either the labels of $w \wedge u$ and $w \wedge v$ are different—in which case there is an edge from w to u and none from w to v , or vice versa, and so $w \in M(u, v)$ by (2)—or they are the same. If they have the same label—say, without loss of generality, label 0—then there is some node σ with label 1 in between them, that is, $w \wedge u \prec \sigma \prec w \wedge v$. Since \mathcal{T} is ramified, σ is the meet of two leaves extending it, and at least one of those leaves, say w' , does not extend $w \wedge u$. So we have $w \wedge w' = w' \wedge u = \sigma$. We also have $w' \wedge v = w' \wedge v = u \wedge v$.



Thus: There is an edge from w' to u and no edge from w' to v , and there is an edge from w to w' and no edge from w to u or to v . Then $w \in M(u, v)$ by (3).

Note that every w on the right-hand-side of (*) satisfies one of (1), (2), or (3). So if we can show that

$$M(u, v) \subseteq \{w : w \wedge u \succ u \wedge v\} \cup \{w : w \wedge v \succ u \wedge v\}$$

then we are done. The right-hand-side of (*) contains u and v so it suffices to show that it is a module. Suppose that w is on the right-hand-side and that w' is not. Suppose without loss of generality that $w \wedge u \prec u \wedge v$. We also have $w' \wedge u = w' \wedge v \succeq u \wedge v$. This implies that $w \wedge w' = w' \wedge u = w' \wedge v$. So whether or not there is an edge between w and w' depends on the label of $w \wedge w'$, and this node is independent of the choice of w . \square

Given distinct vertices u, v in a cograph, let $S(u, v)$ be the least strong module containing u and v . Note that $S(u, v)$ is robust. The definition of a strong module forces the set $S(u, v)$ to not only contain $M(u, v)$, but also be closed under vertices w not yet in the set that can form an *overlapping* set $M(u', w)$ where u' is in the set and some $v' \notin M(u', w)$. (We define two sets as **overlapping** if they are not disjoint and neither is contained in the other.) Lemma 4.9(c) shows that it is enough to consider the situation where $u' = u$ and $v' = v$. In particular, the closure can be done in one step.

Lemma 4.9. *Let \mathcal{G} be a cograph with tree \mathcal{T} , and let u, v be distinct vertices in \mathcal{G} . Then $w \in S(u, v)$ if and only if either $w \in M(u, v)$ or $v \notin M(u, w)$. Therefore $w \in S(u, v)$ can be expressed using an \exists_2 formula of finitary first-order logic.*

Proof. In one direction, if $w \in M(u, v) \subseteq S(u, v)$ then $w \in S(u, v)$. And if $v \notin M(u, w)$ then since $M(u, v)$ intersects $S(u, v)$ at u , and $S(u, v) \not\subseteq M(u, w)$, we must have $M(u, w) \subseteq S(u, v)$ as otherwise $S(u, v)$ would not be strong, and so $w \in S(u, v)$.

Suppose that \mathcal{G} is a graph and let a, b, c be three of its vertices. Suppose that f is an automorphism of order 3 such that $f(a) = b$, $f(b) = c$, and $f(c) = a$. We reason within the decomposition tree \mathcal{T} of \mathcal{G} , recalling that \mathcal{G} corresponds to the leaves of \mathcal{T} . (Since \mathcal{G} and \mathcal{T} are infinitary bi-interpretable (Theorem 4.10), they have the same automorphism group. An automorphism of \mathcal{G} induces an automorphism of \mathcal{T} by mapping robust modules of \mathcal{G} to other robust modules.)

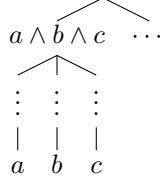
Hence f acts on \mathcal{T} as follows:

$$f(a \wedge b) = f(a) \wedge f(b) = b \wedge c, \quad f(b \wedge c) = c \wedge a, \quad f(a \wedge c) = a \wedge b$$

and

$$f(a \wedge b \wedge c) = a \wedge b \wedge c.$$

Reasoning about trees, at least two of $a \wedge b$, $b \wedge c$, and $c \wedge a$ are equal to $a \wedge b \wedge c$. Since this is fixed by f , and f maps $a \wedge b \mapsto b \wedge c \mapsto c \wedge a \mapsto a \wedge b$, we must have that $a \wedge b = b \wedge c = c \wedge a = a \wedge b \wedge c$.



Now consider the follows sets:

$$\begin{aligned} A &= \{u \in G : u \succeq a \wedge b \wedge c \text{ and } u \wedge a \succ a \wedge b \wedge c\} \\ B &= \{u \in G : u \succeq a \wedge b \wedge c \text{ and } u \wedge b \succ a \wedge b \wedge c\} \\ C &= \{u \in G : u \succeq a \wedge b \wedge c \text{ and } u \wedge c \succ a \wedge b \wedge c\} \\ D &= G - A - B - C \end{aligned}$$

The f maps, setwise, A to B , B to C , and C to A , and fixed D setwise. Then define

$$g(u) = \begin{cases} f(u) & u \in A \\ f^{-1}(u) & u \in B \\ u & u \in C \cup D \end{cases}.$$

This is an automorphism of \mathcal{G} which is its own inverse. Thus if \mathcal{G} has an automorphism of order three, it must have an automorphism of order two, and so cannot have automorphism group \mathbb{Z}_3 . \square

The following proof construction relies on ideas from [BC79] proving that $\text{Free}(\mathcal{P}_4)^{fin}$ is not GI -complete, i.e., not every graph isomorphism problem polynomial-time reduces to it.

Theorem 4.12. *For a finite graph \mathcal{G} , the following are equivalent:*

- (i) $\text{Free}(\mathcal{G})$ is on top for effective and infinitary bi-interpretability;
- (ii) $\text{Free}(\mathcal{G}) \not\subseteq \text{Free}(\mathcal{P}_4)$.

Proof. To show that (i) implies (ii) notice that, by Theorem 4.11, $\text{Free}(\mathcal{P}_4)$ is not on top for infinitary bi-interpretability. Hence, since the property of being on top for effective bi-interpretability is closed under subset we have that $\text{Free}(\mathcal{G})$ is not on top for infinitary bi-interpretability. Since effective bi-interpretability implies infinitary intepretability we also have that $\text{Free}(\mathcal{G})$ is not on top for effective bi-interpretability as well.

For the other direction, suppose that \mathcal{G} is not an induced subgraph of \mathcal{P}_4 . To establish that $\text{Free}(\mathcal{G})$ is on top for effective bi-interpretability—and thus also for infinitary bi-interpretability—it suffices to consider the case where \mathcal{G} contains a cycle as an induced subgraph. This is justified by the following two facts:

- a) a graph \mathcal{G} is an induced subgraph of \mathcal{P}_4 if and only if \mathcal{G} and $\bar{\mathcal{G}}$ do not have a cycle as an induced subgraph. This fact is folklore mentioned in [BC79] Section 4.7. The left-to-right direction is immediate combining the facts that $\overline{\mathcal{P}_4} = \mathcal{P}_4$ and that any induced subgraph of \mathcal{P}_4 cannot contain a cycle. The opposite direction is immediate for graphs of more than 6 vertices (this follows from the fact that the $R(3, 3) = 6$), and it can be proved for vertices of 5 or less vertices exhausting all the possible cases.
- b) The classes $\text{Free}(\mathcal{G})$ and $\text{Free}(\bar{\mathcal{G}})$ are trivially CBF-reducible to each other.

We will show that the class of graphs is CBF-reducible to $\text{Free}(\mathcal{G})$. We will define computable functors $F = (\Phi, \Phi_*) : \text{Graphs} \rightarrow \text{Free}(\mathcal{G})$ and $G = (\Psi, \Psi_*) : F(\text{Graphs}) \rightarrow \text{Graphs}$ satisfying the conditions in Definition 4.6. Given a graph \mathcal{H} , let $n = \min\{k : (\forall n \geq k)(\mathcal{C}_k \text{ is not an induced subgraph of } \mathcal{H})\}$.

Definition of Φ . Given a graph \mathcal{H} , let $\Phi(\mathcal{H})$ be the graph obtained as follows, for an example see Fig. 1.

- (1) For every $v \in H$, add to $\Phi(\mathcal{H})$ a point $\Phi(\mathcal{H})(v)$ and make it part of a copy of \mathcal{C}_{n+1} .
- (2) Then, for every $v, w \in H$ so that $v E w$, connect them by a copy of $\mathcal{P}_{|G|}$ and make each vertex in this path a member of disjoint copies of \mathcal{C}_{n+1} . If $v \not E w$, then still connect them by a copy of $\mathcal{P}_{|G|}$ but make each vertex in this path a member of a disjoint copy of \mathcal{C}_{n+2} .

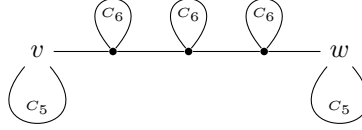


FIGURE 1. The operator Φ on elements v, w connected by an edge to reduce $\text{Free}(\mathcal{P}_5)$. The complement of \mathcal{P}_5 contains a cycle of length 4, thus we use \mathcal{C}_5 and \mathcal{C}_6 .

We need to show that given a graph \mathcal{H} , $\Phi(\mathcal{H})$ is $\text{Free}(\mathcal{G})$ and to do so, it suffices to show that \mathcal{C}_{n-1} is not an induced subgraph of $\Phi(\mathcal{H})$. To check this notice that clearly a copy of \mathcal{C}_{n-1} cannot be in any of the cycles that have been added to $\Phi(\mathcal{H})$. Even more, no copy of \mathcal{C}_{n-1} can share a vertex with such cycles. It remains to exclude the case in which the copy of \mathcal{C}_{n-1} consists of vertices coming from the copies of $\mathcal{P}_{|G|}$. By definition, $|G| \geq n - 1$ and since all paths between vertices of H are of length $|G|$ it follows that any cycle is of length greater than $n - 1$.

Furthermore, the function Φ we defined is computable as it is defined on finite portions of \mathcal{H} and we also have that $\mathcal{H} \cong \mathcal{H}'$ if and only if $\Phi(\mathcal{H}) \cong \Phi(\mathcal{H}')$.

Definition of Φ_ .* Let f be an isomorphism between two graphs \mathcal{H} and \mathcal{H}' . We define the isomorphism $\Phi_*(f)$ from $\Phi(\mathcal{H})$ to $\Phi(\mathcal{H}')$ as follows.

- if $v \in H$, $\Phi_*(f)$ maps $\Phi(\mathcal{H})(v)$ to $\Phi(\mathcal{H}')(f(v))$. Furthermore, $\Phi_*(f)$ maps the copy of \mathcal{C}_n attached to $\Phi(\mathcal{H})(v)$ to the one attached to $\Phi(\mathcal{H}')(f(v))$ (the existence of \mathcal{C}_n is guaranteed by the definition of Φ);
- for every $v, w \in H$, consider $\Phi(\mathcal{H})(v)$ and $\Phi(\mathcal{H})(w)$. By definition of Φ , in $\Phi(\mathcal{H})$ there is a copy of $\mathcal{P}_{|G|+2n+2}$ with endpoints $\Phi(\mathcal{H})(v)$ and $\Phi(\mathcal{H})(w)$ and in which every vertex that is not $\Phi(\mathcal{H})(v)$ or $\Phi(\mathcal{H})(w)$ has attached a copy of \mathcal{C}_{n+1} or \mathcal{C}_{n+2} , depending on whether (v, w) is an edge in \mathcal{H} or not. The isomorphism $\Phi_*(f)$ maps this induced subgraph to the one having endpoints $\Phi(\mathcal{H})(f(v))$ and $\Phi(\mathcal{H})(f(w))$ (which exists, by definition of $\Phi(\mathcal{H}')$).

Notice that $\Phi_*(f)$ is an isomorphism between $\Phi(\mathcal{H})$ and $\Phi(\mathcal{H}')$ which is computable relative to \mathcal{H} , Φ and \mathcal{H}' .

Definition of Ψ . Given $\mathcal{H} \in \mathbb{C}$ let $\Psi(\mathcal{H})$ be the graph having as vertex set the set of vertices $v \in H$ with a copy of \mathcal{C}_n attached to them and with $\text{deg}(v) > 2$. The condition $\text{deg}(v) > 2$ just ensures that we choose the correct vertex in a copy of \mathcal{C}_n : such a condition can be checked computably, as for every copy of \mathcal{C}_n in \mathcal{H} only one vertex satisfies it. Then for every v, w with a copy of \mathcal{C}_n attached to them, let (v, w) be an edge in \mathcal{H} if and only if all vertices in the copy of $\mathcal{P}_{|G|+2n+2}$ with v and w as endpoints have a copy of \mathcal{C}_{n+1} attached to them. The existence of such cycles can be checked computably as every vertex in such a path has attached either a copy of \mathcal{C}_{n+1} or \mathcal{C}_{n+2} . Furthermore, $\mathcal{H} \cong \mathcal{H}'$ if and only if $\Psi(\mathcal{H}) \cong \Psi(\mathcal{H}')$

Definition of Ψ_ .* Let f be an isomorphism between two graphs $\mathcal{H}, \mathcal{H}' \in \mathbb{C} \subseteq \text{Free}(\mathcal{P}_4)$. We define the isomorphism $\Psi_*(f)$ from $\Psi(\mathcal{H})$ to $\Psi(\mathcal{H}')$ as follows: for every $v \in H$ with a copy of \mathcal{C}_n attached to it, $\Psi_*(f)$ maps $\Psi(\mathcal{H})(v)$ to $\Psi(\mathcal{H}')(f(v))$. The fact that it is an isomorphism computable relative to \mathcal{H} , Ψ and \mathcal{H}' follows from the definition of Ψ .

It remains to show that $F = (\Phi, \Phi_*)$ and $G = (\Psi, \Psi_*)$ are effective pseudo-inverses. We first have to construct Turing operators Λ and Γ witnessing that GF and FG are naturally isomorphic to the identity functor. To construct Λ , note that for a vertex v in a graph \mathcal{H} , Φ builds a designated vertex $\Phi(\mathcal{H})(v)$ and similarly Ψ , identifies the vertex set of the original graph inside $\Phi(\mathcal{H})$ and builds a vertex, call it x_v . We

let $\Lambda(\mathcal{H}) : v \mapsto x_v$. By construction Λ is computable and one can check that for any two graphs $\mathcal{H}_1, \mathcal{H}_2$ and isomorphism $f : \mathcal{H}_1 \cong \mathcal{H}_2$, the associated diagram commutes, showing that GF is effectively naturally isomorphic to the identity functor. The operator Γ can be constructed in similar fashion to Λ and checking that Γ induces an effective natural isomorphism of FG and the identity is routine. At last, it remains to check that for every graph \mathcal{H} , $F(\Lambda^{\mathcal{H}}) = \Gamma^{F(\mathcal{H})}$ and for every graph $\mathcal{H} \in \mathbb{C}$, $G(\Gamma^{\mathcal{H}}) = \Lambda^{G(\mathcal{H})}$. Both facts are again guaranteed by the uniformity of the construction. \square

The following theorem is the analogous of Theorem 4.12 for Borel reducibility. Notice that in Theorem 4.12 we did not have to consider the case $\text{Free}(\mathcal{G}) \subseteq \text{Free}(\overline{\mathcal{P}_4})$ as $\mathcal{P}_4 = \overline{\mathcal{P}_4}$ (this fact was used in part a) of the proof).

Theorem 4.13. *For a finite graph \mathcal{G} , The following are equivalent:*

- (i) $\text{Free}(\mathcal{G})$ is on top for Borel reducibility;
- (ii) $\text{Free}(\mathcal{G}) \not\subseteq \text{Free}(\mathcal{P}_3)$ and $\text{Free}(\mathcal{G}) \not\subseteq \text{Free}(\overline{\mathcal{P}_3})$.

Proof. To prove (i) implies (ii) we prove the contrapositive. To do so it suffices to combine the following two facts: (a) neither $\text{Free}(\mathcal{P}_3)$ nor $\text{Free}(\overline{\mathcal{P}_3})$ are not on top for Borel reducibility, and (b) the property of not being on top for Borel reducibility is closed under subset. Property (b) just follows by definition of Borel reducibility. Property (a) follows from the following two facts:

- a graph in $\text{Free}(\mathcal{P}_3)$ is just an equivalence structure, and it is well-known that deciding whether two equivalence structures are isomorphic is Borel (in particular it is Π_4^0). Indeed, to determine whether two equivalence structures are isomorphic, one has to compare the numbers of equivalence classes of each size (see e.g. [Mon21, page 92]);
- $\text{Free}(\overline{\mathcal{P}_3})$ Borel reduces to $\text{Free}(\mathcal{P}_3)$ via the (Borel) function that, given in input a graph outputs its complement.

For the opposite direction, by Theorem 4.12, we know that if $\text{Free}(\mathcal{G}) \not\subseteq \text{Free}(\mathcal{P}_4)$, then $\text{Free}(\mathcal{G})$ is on top for effective bi-interpretability and hence on top for Borel reducibility. Theorem 4.3 tells us that $\text{Free}(\mathcal{P}_4)$ is Borel complete. Notice that every graph \mathcal{G} such that \mathcal{G} is a proper induced subgraph of \mathcal{P}_4 is either an induced subgraph of \mathcal{P}_3 or $\overline{\mathcal{P}_3}$. The fact that the notion being not Borel complete is closed under subset combined with what we just proved, namely that both $\text{Free}(\mathcal{P}_3)$ and $\text{Free}(\overline{\mathcal{P}_3})$ are not Borel complete, allows us to conclude the proof. \square

As a corollary of Theorems 4.13 and 4.12 and Lemma 3.11, we get that $\text{Free}(\mathcal{P}_4)$ is the only $\text{Free}(\mathcal{G})$ class that is on top for Borel reducibility but not for effective bi-interpretability. In particular, we get the following as announced in the introduction.

Theorem 4.14. *$\text{Free}(\mathcal{P}_4)$ is on top for Borel reducibility but not for effective bi-interpretability.*

REFERENCES

- [19] “Séminaire Bourbaki : volume 2016/2017, exposés 1120-1135.” fre;eng. In: *Astérisque*, 407. Séminaire Bourbaki (2016-2017). Paris: Société mathématique de France, 2019. ISBN: 2856298974.
- [AL15] Aistis Atminas and Vadim V. Lozin. “Labelled induced subgraphs and well-quasi-ordering”. In: *Order* 32.3 (2015), pp. 313–328. ISSN: 0167-8094,1572-9273. DOI: 10.1007/s11083-014-9333-9.
- [Bab16] László Babai. “Graph isomorphism in quasipolynomial time [extended abstract]”. In: *STOC’16—Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, New York, 2016, pp. 684–697. ISBN: 978-1-4503-4132-5. DOI: 10.1145/2897518.2897542.
- [BC79] Kellog S Booth and Charles J Colbourn. *Problems polynomially equivalent to graph isomorphism*. Computer Science Department, Univ., 1979.
- [CD08] Bruno Courcelle and Christian Delhommé. “The modular decomposition of countable graphs. Definition and construction in monadic second-order logic”. In: *Theoret. Comput. Sci.* 394.1-2 (2008), pp. 1–38. ISSN: 0304-3975,1879-2294. DOI: 10.1016/j.tcs.2007.10.046.
- [CLB81] D. G. Corneil, H. Lerchs, and L. Stewart Burlingham. “Complement reducible graphs”. In: *Discrete Appl. Math.* 3.3 (1981), pp. 163–174. ISSN: 0166-218X,1872-6771. DOI: 10.1016/0166-218X(81)90013-5.
- [Dam90] Peter Damaschke. “Induced subgraphs and well-quasi-ordering”. In: *J. Graph Theory* 14.4 (1990), pp. 427–435. ISSN: 0364-9024,1097-0118. DOI: 10.1002/jgt.3190140406.

- [FS89] Harvey Friedman and Lee Stanley. “A Borel reducibility theory for classes of countable structures”. In: *J. Symbolic Logic* 54.3 (1989), pp. 894–914. ISSN: 0022-4812,1943-5886. DOI: 10.2307/2274750.
- [Gao01] Su Gao. “Some Dichotomy Theorems for Isomorphism Relations of Countable Models”. In: *The Journal of Symbolic Logic* 66.2 (2001), pp. 902–922.
- [Gao09] Su Gao. *Invariant descriptive set theory*. Boca Raton, FL: CRC Press, 2009.
- [Har+17] Matthew Harrison-Trainer, Alexander Melnikov, Russell Miller, and Antonio Montalbán. “Computable functors and effective interpretability”. In: *J. Symb. Log.* 82.1 (2017), pp. 77–97. ISSN: 0022-4812,1943-5886. DOI: 10.1017/jsl.2016.12.
- [HBD17] Harald Andrés Helfgott, Jitendra Bajpai, and Daniele Dona. “Graph isomorphisms in quasi-polynomial time”. In: (2017). Translation from the French original with additions. eprint: arXiv:1710.04574.
- [Hel+13] Marc Hellmuth, Maribel Hernandez-Rosales, Katharina T. Huber, Vincent Moulton, Peter F. Stadler, and Nicolas Wieseke. “Orthology relations, symbolic ultrametrics, and cographs”. In: *J. Math. Biol.* 66.1-2 (2013), pp. 399–420. ISSN: 0303-6812,1432-1416. DOI: 10.1007/s00285-012-0525-x.
- [HM12] Kenneth Harris and Antonio Montalbán. “On the n -Back-and-Forth Types of Boolean Algebras”. In: *Transactions of the American Mathematical Society* 364.2 (2012), pp. 827–866.
- [HMM18] Matthew Harrison-Trainer, Russell Miller, and Antonio Montalbán. “Borel functors and infinitary interpretations”. In: *J. Symb. Log.* 83.4 (2018), pp. 1434–1456. ISSN: 0022-4812,1943-5886. DOI: 10.1017/jsl.2017.81.
- [Hod93] Wilfrid Hodges. *Model Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1993.
- [HPW24] Geña Hahn, Maurice Pouzet, and Robert Woodrow. “Siblings of countable cographs”. In: *J. Mult.-Valued Logic Soft Comput.* 42.4 (2024), pp. 299–331. ISSN: 1542-3980,1542-3999.
- [Kec95] Alexander S. Kechris. *Classical Descriptive Set Theory*. 1st ed. Vol. 156. Graduate Texts in Mathematics. Springer-Verlag, New York, 1995, pp. xviii+402. ISBN: 0387943749. DOI: 10.1007/978-1-4612-4190-4.
- [Kru60] Joseph B. Kruskal. “Well-Quasi-Ordering, the Tree Theorem, and Vazsonyi’s Conjecture”. In: *Transactions of the American mathematical society* 95.2 (1960), pp. 210–225. JSTOR: 1993287.
- [Mar02] David Marker. *Model Theory: An Introduction*. Springer Science & Business Media, 2002. ISBN: 978-0-387-98760-6. DOI: 10.1007/b98860.
- [Mon12] Antonio Montalbán. “Counting the back-and-forth types”. In: *J. Logic Comput.* 22.4 (2012), pp. 857–876. ISSN: 0955-792X,1465-363X. DOI: 10.1093/logcom/exq048.
- [Mon21] Antonio Montalbán. *Computable structure theory—within the arithmetic*. Perspectives in Logic. Cambridge University Press, Cambridge; Association for Symbolic Logic, Ithaca, NY, 2021, pp. xxii+190. ISBN: 978-1-108-42329-8. DOI: 10.1017/9781108525749.
- [Mon25] Antonio Montalbán. *Computable Structure Theory: Beyond the Arithmetic*. to appear, 2025.
- [Ric81] Linda Jean Richter. “Degrees of structures”. In: *J. Symbolic Logic* 46.4 (1981), pp. 723–731. ISSN: 0022-4812,1943-5886. DOI: 10.2307/2273222.
- [Ros22] Dino Rossegger. “Degree Spectra of Analytic Complete Equivalence Relations”. In: *The Journal of Symbolic Logic* 87.4 (2022), pp. 1663–1676. ISSN: 0022-4812, 1943-5886. DOI: 10.1017/jsl.2021.82. arXiv: 2010.00755.

(Harrison-Trainor) DEPARTMENT OF MATHEMATICS, STATISTICS, AND COMPUTER SCIENCE, UNIVERSITY OF ILLINOIS, CHICAGO, IL, USA 60607

Email address: vittorio.cipriani@tuwien.ac.at

URL: <https://vittoriocipriani.github.io/>

Email address: ekaterina.fokina@tuwien.ac.at

Email address: mht@uic.edu

URL: <https://homepages.math.uic.edu/>

Email address: li.ko@tuwien.ac.at

Email address: dino.rossegger@tuwien.ac.at

URL: <https://drossegger.github.io/>