Model-Checking for First-Order Logic with Disjoint Paths Predicates in Proper Minor-Closed Graph Classes[1]

Petr A. Golovach[2] Giannos Stamoulis[3] Dimitrios M. Thilikos[3]

Abstract

The *disjoint paths logic*, $\mathsf{FOL{+}DP}$, is an extension of First-Order Logic (FOL) with the extra atomic predicate $\mathsf{dp}_k(x_1, y_1, \ldots, x_k, y_k)$, expressing the existence of internally vertex-disjoint paths between x_i and y_i, for $i \in \{1, \ldots, k\}$. This logic can express a wide variety of problems that escape the expressibility potential of FOL. We prove that for every proper minor-closed graph class, model-checking for $\mathsf{FOL{+}DP}$ can be done in quadratic time. We also introduce an extension of $\mathsf{FOL{+}DP}$, namely the *scattered disjoint paths logic*, $\mathsf{FOL{+}SDP}$, where we further consider the atomic predicate $s\text{-}\mathsf{sdp}_k(x_1, y_1, \ldots, x_k, y_k)$, demanding that the disjoint paths are within distance bigger than some fixed value s. Using the same technique we prove that model-checking for $\mathsf{FOL{+}SDP}$ can be done in quadratic time on classes of graphs with bounded Euler genus.

Keywords: Algorithmic meta-theorems, Model-checking, First-order logic, Disjoint paths, Hadwiger number, Graph minors, Irrelevant vertex technique.

---

[1]An extended abstract of this paper appeared in the *Proceedings of the 34th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2023).*

[2]Department of Informatics, University of Bergen, Norway. Supported by the Research Council of Norway via the project BWCA (314528). Email: `petr.golovach@uib.no`.

[3]LIRMM, Univ Montpellier, CNRS, Montpellier, France. Supported by the ANR projects DEMOGRAPH (ANR-16-CE40-0028), ESIGMA (ANR-17-CE23-0010), and the French-German Collaboration ANR/DFG Project UTMA (ANR-20-CE92-0027). The third author was also supported by the *Ministère de l'Europe et des Affaires étrangères* (MEAE) and the *Ministère de l'Enseignement supérieur et de la Recherche* (MESR), via the Franco-Norwegian project PHC Aurora projet N° 51260WL (2024). Emails: `giannos.stamoulis@lirmm.fr`, `sedthilk@thilikos.info`.

Contents

1 Introduction

Logic plays a fundamental role in algorithmic research. It provides a universal language for formally describing computational problems and is important for the investigation of their computational complexity. In many cases, the accumulation of knowledge on algorithm design revealed that several algorithmic techniques have conceptual similarities that result from some common logical description of the problems where they apply. These similarities become effective when the inputs of the corresponding problems have certain structural characteristics. In some cases, this empirical evidence has been materialized in the so called Algorithmic Meta-Theorems (AMTs), a term introduced by Martin Grohe in [109]. Such theorems typically provide two types of conditions, a logical one and a combinatorial one, such that every problem that is expressible by the logical condition can be solved efficiently when its inputs are restricted by the combinatorial condition. The importance of AMTs resides to the fact that they are able to unify wide families of computational problems (and also the algorithmic solutions for them) under a single model-theoretic/combinatorial framework (see [109, 111, 153]).

1.1 AMTs for MSOL and FOL

Probably, the most prototypical AMT is known as *Courcelle's Theorem* proved in [52] (see also [9, 38] and [68]), asserting that every problem on graphs that is expressible by a sentence φ in Monadic Second Order Logic (MSOL) can be solved in time[1] $\mathcal{O}_{|\varphi|,k}(n)$, when restricted to graphs of treewith at most k. Clearly, when φ and k are fixed, this readily implies a linear-time algorithm. However, we prefer to display the dependencies on φ and k, under the $\mathcal{O}_{|\varphi|,k}$ notation, so as to make clear that this theorem provides a linear-time parameterized[2] algorithm, for every problem expressible in MSOL, when it is parameterized by treewidth.

Clearly, the logical/combinatorial compromise of Courcelle's theorem is not the only possible one. In fact, each AMT constitutes a different compromise between the logical and the combinatorial condition and a considerable amount of research in the theory of algorithms has been dedicated to the conception of alternative such compromises. Also, for particular logics, research has been dedicated to the identification of their combinatorial horizon, i.e., the most general combinatorial conditions that can accompany them in an AMT. For instance, for MSOL, the meta-algorithmic horizon is, under certain assumptions, delimited by the graph classes of bounded treewidth (see [21, 94, 156]).

Meta-algorithmics of FOL. Given that the meta-algorithmic limits of MSOL are fairly well understood, research on AMTs has been largely oriented to the meta-algorithmics of First-Order Logic (FOL). The two most powerful results in this direction concern two different types of combinatorial conditions. The first was given by Grohe, Kreutzer, and Siebertz in [112] and is the graph class property of being *nowhere dense*. The second was given by Bonnet, Kim, Thomassé, and Watrigant in [34] and is the graph class property of having *bounded twin-width*. We should stress here that the notion of being nowhere dense originates from a long line of research on graph sparsity, initiated by Nešetřil and Ossona de Mendez in [177] (see also [69, 154, 155, 178, 179, 189]). On the other side, twin-width is a recently introduced graph parameter, defined in terms of sequences of vertex identifications (see [10, 17, 26, 28–36, 67, 93, 95, 121, 127, 152, 186, 190, 191, 200] for a sample of the vibrant

[1] Let $\mathbf{t} = (x_1, \ldots, x_l) \in \mathbb{N}^l$ and $f, g : \mathbb{N} \to \mathbb{N}$. We adopt the notation $f(n) = \mathcal{O}_{\mathbf{t}}(g(n))$ in order to denote that there exists a computable function $\ell : \mathbb{N}^l \to \mathbb{N}$ such that $f(n) = \mathcal{O}(\ell(\mathbf{t}) \cdot g(n))$.

[2] More generally, we say that a computational problem, parameterized by k, is FPT (Fixed Parameter Tractable) when it admits an algorithm running in $\mathcal{O}_k(n^{O(1)})$ time. We assume that the reader is familiar with the basic concepts of parameterized algorithms and parameterized complexity classes – see [55, 80, 184].

current research on the algorithmic and combinatorial properties of twin-width). Seminal results on the above two combinatorial conditions indicate that, under certain assumptions, they approach the combinatorial horizon of FOL (see [69] for nowhere density and [31] for bounded twin-width). Research on the meta-algorithmics of FOL is nowadays quite active and has moved to several directions such as the study of FOL-interpretability [27,92,180–182,187] or the enhancement of FOL with counting/numerical predicates [68, 113, 157, 158] (see also [72, 108, 114, 206] for other extensions).

AMTs between FOL and MSOL. A challenging direction is the introduction of new logics whose expressive power is between FOL and MSOL that can lead to AMTs under combinatorial conditions that are less general than those applicable for FOL and more general than those applicable for MSOL. Two approaches that have been initiated in this direction are the following. The first direction is the introduction of compound logics that can express problems whose description combines both FOL and MSOL queries. Such a compound logic has been recently introduced in [81] and yielded AMTs that are applicable on a wide family of graph modification problems. The second direction is the extension of FOL with additional predicates that are not expressible in FOL. An important step in this direction was the introduction of the separator logic FOL+conn. This extension of FOL was introduced independently by Schirrmacher, Siebertz, and Vigny in [201] and by Bojańczyk in [22] (under the name *separator logic*), who considered, for every $k \geq 1$, the general predicate $\mathsf{conn}_k(\mathsf{x}, \mathsf{y}, \mathsf{z}_1, ..., \mathsf{z}_k)$, that evaluates to true on a graph G if (the valuations of) x and y are joined in G by a path that avoids (the valuations) of the variables $\{\mathsf{z}_1, ..., \mathsf{z}_k\}$. According to the recent meta-algorithmic results of Pilipczuk, Schirrmacher, Siebertz, Toruńczyk, and Vigny in [188], every problem on graphs that is expressible by some formula $\varphi \in$ FOL+conn can be solved in time $\mathcal{O}_{|\varphi|,r}(n^3)$, where $r = \mathsf{hj}(G)$ is the Hajós number[3] of G. Notice that this AMT implies the existence of parameterized algorithms on graph classes with bounded Hajós number for problems whose definition uses connectivity queries and therefore are not expressible in FOL. The most indicative example of a (meta-) problem displaying the expressibility power of FOL+conn is Elimination Distance to φ, asking whether the elimination distance of a graph G from some model of $\varphi \in$ FOL is at most k. This problem is W[2]-hard (when parameterized by k) even for simple instantiations of φ [85], however, it admits a time $\mathcal{O}_{|\varphi|,k,r}(n^3)$ algorithm when restricted to graph classes with Hajós number at most r, because of the results in [188]. For more examples of the expressibility power of FOL+conn, see [201]. Also, in [188] it was proved that the tractability horizon of FOL+conn is, under certain assumptions, delimited by graph classes of bounded Hajós number.

1.2 AMTs for FOL+DP and extensions

As a next step towards a more expressive logic, Schirrmacher, Siebertz, and Vigny [201] defined the, more expressive, *disjoint-paths logic* FOL+DP by adding, for every k, the atomic predicates $\mathsf{dp}_k(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_k, \mathsf{y}_k)$ that evaluate to true if there are internally vertex-disjoint paths between (the valuations of) x_i and y_i, for all $i \in \{1, \ldots, k\}$. In the same paper they defined[4] the logical hierarchy FOL+dp_k that uses predicates for at most k disjoint paths and proved that these fragments of FOL+DP define a *strict* descriptive complexity hierarchy.

The challenging open question is whether an AMT exists for this logic, under some suitable combinatorial restriction that is more general than the one of having bounded treewidth (that is where Courcelle's theorem is applicable).

[3] The *Hajós number* is the maximum h for which G contains a subivision of K_h as a subgraph.

[4] In the definition of FOL+DP we insist that the paths are *disjoint* while in [201] paths are required to be *internally disjoint* (certainly, these two variants define equivalent formulas). We insist on the "complete disjointness" as this permits us to see FOL+DP as a special case of the more general FOL+SDP that we introduce in this paper.

Our results. Given a graph G, we define the *Hadwiger number* of G, denoted by $\mathsf{hw}(G)$, as the maximum r for which G contains K_r as a minor.[5] Our main result is the following AMT.

Theorem 1. *Every problem on graphs that is expressible by some formula φ in* FOL+DP *can be solved by an algorithm running in time $\mathcal{O}_{|\varphi|,r}(n^2)$, where r is the Hadwiger number of G.*

Some indicative (meta) problems whose standard parameterizations (i.e., those defined by the parameter k in their inputs) are automatically classified in FPT because of Theorem 1 are MINOR CONTAINMENT, TOPOLOGICAL MINOR CONTAINMENT, CYCLABILITY, UNORDERED LINKABILITY, ORDERED LINKABILITY, $\mathcal{F}$-MINOR-DELETION, $\mathcal{F}$-TOPOLOGICAL MINOR-DELETION, $\mathcal{F}$-CONTRACTION DELETION (for bounded genus graphs), ANNOTATED $\mathcal{F}$-$\preceq$-DELETION, SUBSET $\mathcal{F}$-$\preceq$-DELETION, φ-DELETION, φ-AMALGAMATION, $\mathcal{L}$-φ-REPLACEMENT, φ-ELIMINATION DISTANCE, φ-RECONFIGURATION. For the definitions, the complexity, and the FOL+DP-expressibility of all these problems we refer the reader to Appendix A.

Our next step towards a more expressive logic, is to extend FOL+DP by considering, for every $k \geq 1$, a more general form of predicate $s\text{-}\mathsf{dp}_k(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_k, \mathsf{y}_k)$ where we now demand that the disjoint paths in question are pairwise s-scattered, i.e., there are no two vertices of two distinct paths that are within distance at most s. We call the new logic FOL+SDP. As $0\text{-}\mathsf{dp}_k(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_k, \mathsf{y}_k) = \mathsf{dp}_k(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_k, \mathsf{y}_k)$, we readily have that FOL+SDP is an extension of FOL+DP. Our second result is the following AMT.

Theorem 2. *Every problem on graphs that is expressible by some formula φ in* FOL+SDP *can be solved by an algorithm running in time $\mathcal{O}_{|\varphi|,k}(n^2)$, where k is the Euler genus of G.*

Some indicative problems whose standard parameterizations are automatically classified in FPT because of Theorem 2 are INDUCED MINOR, INDUCED TOPOLOGICAL MINOR CONTAINMENT, CONTRACTION CONTAINMENT, INDUCED UNORDERED LINKABILITY, INDUCED ORDERED LINKABILITY, $\mathcal{F}$-INDUCED MINOR DELETION, $\mathcal{F}$-INDUCED TOPOLOGICAL MINOR DELETION, φ-DELETION, φ-AMALGAMATION, $\mathcal{L}$-φ-REPLACEMENT, φ-ELIMINATION DISTANCE, φ-RECONFIGURATION. For the definitions, the complexity, and the FOL+SDP-expressibility of all these problems we refer the reader to Appendix A.

1.3 The irrelevant vertex technique

In Volume XIII of their Graphs Minors series, Robertson and Seymour introduced the celebrated *irrelevant vertex technique* in order design a time $\mathcal{O}_k(n^3)$ algorithm for the following problem [193].

DISJOINT PATHS
Input: a graph G and pairs $(s_1, t_1), \ldots, (s_k, t_k)$ of vertices of G.
Question: Are there pairwise vertex-disjoint paths between s_i and t_i, for $i \in \{1, \ldots, k\}$?

Notice that the description of the above problem does not fit in FOL. It demands the existence of k pairwise disjoint sets of vertices each inducing a *connected* graph containing the terminals s_i and t_i. While connectivity is expressible in MSOL it is known that it cannot be expressed in FOL (see e.g., [70, 164]).

The general idea behind the irrelevant vertex technique is that if some part of the input graph is "sufficiently insulated" from the rest of the graph, then it may be *irrelevant* in the sense that the

[5]Given two graphs G and H, H is a *minor* of G if G contains a contraction of H as a subgraph.

solution can be "reconfigured" away from it. When this idea applies, then this irrelevant part can be safely deleted and produce an equivalent, and simpler, instance of the problem.

The original application of the irrelevant vertex technique for the Disjoint Paths problem have had two phases:

- **1st phase**: when the input graph G contains a big (as a function of k) clique minor
- **2nd phase**: when G minor-excludes a clique, that is G has small Hadwiger number.

Theorem 1 and Theorem 2 deal with the applicability of the 2nd phase. We next give a brief outline of how this phase was applied for the Disjoint Paths problem and, in particular, in the "non-trivial" situation where G has "big" treewidth. To deal with this situation, Robertson and Seymour proved in [193] the so called *Flat Wall Theorem*, asserting that if a graph has small Hadwiger number and big treewidth, then after the removal from G of "few" vertices, called *apex vertices*, the resulting graph contains a big wall that is "flat". Intuitively, by the term "flat wall" we refer to a wall W whose perimeter P contains a separator S of G where the inner part of the wall is inside one of the connected components C of $G \setminus V(S)$ and where no two disjoint (s_i, t_i)-paths, $i \in \{1, 2\}$, exist in the graph $G[V(P) \cup V(C)]$, called the *compass* of W, where s_1, s_2, t_1, and t_2 are vertices of S, appearing in this ordering in P. This flatness property implies that every set of homocentric cycles around the central part of the wall can act as a system of separators "insulating" the two central vertices of W from the part of the graph that lies outside the compass of the wall. With this structural result at hand, Robertson and Seymour proved that every set of k disjoint paths that may certify a yes-instance of the Disjoint Paths problem can be rerouted away from its central vertices, that is it they be declared irrelevant and be safely discarded from G. The proof of this rerouting argument is quite technical and was given in Volumes XXI and XXII of the Graph Minors series (this result is now known as the Unique Linkage Theorem – see [2,101,103,147,171] for later proofs and improvements). Moreover, Kawarabayashi, Kobayashi, and Reed proved in [139] that the flat wall W (and therefore its central vertices as well) can be found in linear time. Given now that a simpler equivalent instance of the Disjoint Paths problem is found, we may repeat the above procedure a linear number of times until the treewidth is "small" so that the problem can be solved by a dynamic programming algorithm (which exists because of Courcelle's theorem). This, taken into account the improvement of [139], takes a total of $\mathcal{O}_k(n^2)$ time.

The potential of irrelevant vertex technique. To adapt the above arguments for other problems has been a challenging enterprise for graph algorithm designers during the last 20 years. For a indicative (while not exhaustive) list of papers that made use of this technique, see [1, 14, 49, 81, 82, 84, 87, 89, 90, 100, 101, 110, 118, 126, 129, 132–135, 137, 141–145, 149, 166, 170, 196–199, 203]. Typically, for each problem, the challenge is to give an algorithm that is able to detect, in polynomial time, some vertex that can be declared irrelevant and then prove that this vertex is indeed irrelevant in the sense that discarding it from the input graph creates an equivalent instance. In some cases, apart from declaring a vertex irrelevant, "annotated versions" of problems have been considered and vertices may also be declared *annotation-irrelevant* in the sense that they can be safely excluded from the set of annotated vertices. This extended concept of irrelevancy was used in [99] for the Cyclability problem and in [101] for the $\mathcal{F}$-Topological Minor-Deletion problem (see also the meta-algorithmic results in [81, 82, 84]). In our proof of Theorem 1 and Theorem 2 we largely make use of the annotation technology. In fact we consider an annotated set for the variables quantified in the FOL+DP formula.

Most of the problems that are amenable to the application of the irrelevant vertex technique have a common denominator: they are not FOL-expressible and they deal with graph classes with unbounded treewidth, that go beyond the combinatorial applicability of MSOL. Thus, they escape the logical/combinatorial conditions of the known meta-algorithmic technology of FOL and MSOL.

The proof of Theorem 1 and Theorem 2 is abstracting the irrelevant vertex technology of all the aforementioned problems into two AMTs. Theorem 1 (resp. Theorem 2) essentially indicates that, for graphs of bounded Hadwiger number (Euler genus), the descriptive potential of the DISJOINT (INDUCED) PATHS problem can be "embedded" inside FOL in the form of the predicates $\mathsf{dp}_k(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_k, \mathsf{y}_k)$ (s-$\mathsf{dp}_k(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_k, \mathsf{y}_k)$).

What to do with a clique. Clearly Theorem 1 and Theorem 2 concern the 2nd phase of the irrelevant vertex technique where the Hadwiger number is bounded. At this point we wish to mention that the applicability of the 1st phase, concerning the question "*what to do with a clique*", may vary depending on the problem in question. We distinguish three main categories of parameterized problems.

A. The first category contains standard parameterizations of problems, such as MINOR CONTAINMENT, $\mathcal{F}$-MINOR AMALGAMATION, $\mathcal{F}$-MINOR DELETION, or $\mathcal{F}$-MINOR LOCAL REPLACEMENT, where Theorem 1 applies and, moreover, big enough Hadwiger number immediately certifies a yes- or a no-instance. Given that checking whether a clique K_r is a minor of a graph can be done in time $\mathcal{O}_r(n^2)$ [139], for such problems, just FO+DP-expressibility is enough for implying that a problem is FPT, even in general graphs.

B. The second category of parameterized problems contains those where the 1st phase is non-applicable in general, in the sense that they are already intractable. For instance, the standard parameterization of CYCLABILITY or SUBSET LINKABILITY, where Theorem 1 applies, is co-W[1]-hard and problems, where Theorem 2 applies, such as INDUCED DISJOINT PATHS, CONTRACTION CONTAINMENT, and INDUCED MINOR CONTAINMENT are NP-complete even for fixed values of the parameter (see [136], [44, 162, 163], and [76, 161] respectively).

C. The third category concerns standard parameterizations of problems, such as DISJOINT PATHS [193], TOPOLOGICAL MINOR CONTAINMENT [110] , $\mathcal{F}$-TOPOLOGICAL MINOR DELETION [87], $\mathcal{F}$-TM ELIMINATION DISTANCE [5], and $\mathcal{F}$-TM-TREEWIDTH [5], where extra algorithmic machinery is employed for dealing with the 1st phase (typically related with the recursive understanding technique [48, 50, 110, 169]). To our knowledge, there is no general treatment of the 1st phase, further than the results of [5, 87, 110, 193]. To investigate the meta-algorithmic conditions that may unify them, even for some combinatorial condition that is more general that having bounded Hadwiger number, is an interesting open challenge.

For more on the classification of the problems that are treated by Theorem 1 and Theorem 2 according to the above three categories, see Appendix A.

Organization of the paper. In Section 2, we provide an overview of our proof. In Section 3, we provide some basic definitions that will be used throughout the paper. In Section 4, we present a way to translate model-checking to (recursive) folio containment. Then, in Section 5, we give some additional definitions and results for dealing with collections of paths inside (partially planar) graphs. In Section 6, we present a trick to transform a disjoint paths query to one that can deal with the presence of some *apex* vertices that can "spoil" flatness and in Section 7 we present the combinatorial result that supports the correctness of the main subroutine of the algorithm of Theorem 1 and Theorem 2. Next, in Section 8, we present the proof of Theorem 1 and in Section 9 we present the proof of Theorem 2. We conclude the paper with Section 10. In Appendix A we present a list of problems expressible in $\mathsf{FOL+DP}$ and in $\mathsf{FOL+SDP}$ and in Appendix B we present the flat wall framework that we use in this paper, which was introduced in [197]. Appendix C contains missing complexity proofs of problems mentioned in Section 10 and Appendix A.

2 Overview of the proof

In this section we summarize the main ideas involved in the proof of Theorem 1 and Theorem 2. We describe our approach for graphs. However, our results are proven for *colored graphs* (i.e., graphs equipped with a sequence of subsets of their vertex set).

2.1 General scheme of the algorithm

For our algorithms we follow the typical motif of the irrelevant vertex technique: if the treewidth of the input graph G is upper-bounded by an appropriately chosen function, depending only on the sentence $\varphi \in \mathsf{FOL+DP}$ and $\mathsf{hw}(G)$, then because of Courcelle's Theorem [52–54] we can check whether G satisfies φ in linear time, using the fact that $\mathsf{FOL+DP}$ is a fragment of MSOL (see Subsection 3.3). Otherwise, if the treewidth is "large enough", we identify an irrelevant vertex in linear time, that is a vertex whose removal does not affect satisfiability of φ. This highly non-trivial procedure of finding an irrelevant vertex is our main goal and, in what follows, we describe in an intuitive level the main ingredients of this approach.

Irrelevant vertices for model-checking. Applying the irrelevant vertex technique in a model-checking setting demands the restriction of the part of the graph that is used to interpret variables and predicates of the sentence. To show that a vertex is irrelevant, one has to prove that whether the given sentence is satisfied or not does not depend on the presence of this vertex inside the graph. The "building blocks" of our sentences are either first-order variables (quantified by universal or existential quantifiers), or interpretations of relation symbols of the given vocabulary (in our case, these are edges and colors in the vertices), or disjoint-path predicates between some of the already quantified first-order variables. Towards building an irrelevant vertex argument, an essential step is to *reduce the scope* of the quantification of the variables, while preserving the satisfiability status of the sentence. The fact that a vertex can be discarded from the scope of a quantifier, permits to declare it *annotation-irrelevant* with respect to this quantifier. Moreover, even if some vertex is annotation-irrelevant with respect to all quantifiers, this vertex could still be important for the existence (or not) of disjoint paths between vertices (that are picked inside the annotation). Such a vertex can be removed from the graph, therefore be declared *problem-irrelevant*, only if we can guarantee that disjoint paths can be safely rerouted away from it. The pursue of such a problem-irrelevant vertex is executed inside a "big enough" bidimensional area of vertices that are annotation-irrelevant for *all* quantifiers. While this does not deviate from the "typical" rerouting arguments of the irrelevant vertex technique, dealing with the restriction of the annotation is the most demanding part. In the rest of this section, we aim to demonstrate a way to tackle this problem.

2.2 Translating model-checking to (recursive) folio containment

Our main tool in order to create equivalent (annotated) instances is to express model-checking in graph-theoretic terms. Throughout this section, we assume that all sentences are given in prenex-normal form, i.e., $\varphi = Q_1\mathsf{x}_1 \ldots Q_r\mathsf{x}_r\psi(\mathsf{x}_1,\ldots,\mathsf{x}_r)$, where $Q_1,\ldots,Q_r \in \{\forall,\exists\}$, $\mathsf{x}_1,\ldots,\mathsf{x}_r$ are first-order variables, and $\psi(\mathsf{x}_1,\ldots,\mathsf{x}_r)$ is a quantifier-free formula with $\mathsf{x}_1,\ldots,\mathsf{x}_r$ as free variables.

Assigning annotated graphs to rooted trees. Our first step (Section 4) is to interpret the satisfaction of a sentence φ from a graph in terms of the existence of a subtree inside a tree where the graph is embedded, such that the bifurcations of this subtree correspond to the quantifiers of φ and the vertices collected in each root-to-leaf path evaluate to true the quantifer-free "tail" of φ. These trees, known as *game trees*, appear with different names in the literature, like *evaluation trees* [91] or *morphism trees* [34] (see also [107]). These trees follow the recursive structure of a set $\mathsf{sig}^r(G, R_1, \ldots, R_r)$, where G is a graph and $R_1, \ldots, R_r \subseteq V(G)$. This set, called *signature* of $(G, R_1, \ldots, R_r)$, is defined as the recursive collection of the different types of tuples $v_1, \ldots, v_i$ $i \leq r$ of vertices of $V(G)$, where each v_j belongs to R_j for every $j \leq i$ (see the definition in Subsection 4.1). We can construct a rooted tree (T, t_0) following the recursive structure (of depth r) of this set and this naturally gives a mapping of each node of the tree to the vertex of G of a particular type (when considering the tuple of all ancestors of it). We call this mapping an *assignement* of $(G, R_1, \ldots, R_r)$ to (T, t_0). Intuitively, every root-to-leaf path is mapped to a tuple of (possibly repeating) vertices $v_1, \ldots, v_r$, where $v_i \in R_i$, for every $i \in \{1, \ldots, r\}$ (see Figure 1). Also, we define a *φ-spanning* subtree of a rooted tree to be the (sub)tree T' with the same root that is obtained following the quantifiers of φ, i.e., if $Q_i = \exists$, then every node of T' of depth i has only one child in T', while if $Q_i = \forall$, every node of T' of depth i bifurcates (in T') to all its children in T (see Figure 1 for an example). In this setting, an (annotated) formula $\varphi = Q_1\mathsf{x}_1 \in \mathsf{R}_1 \ldots Q_r\mathsf{x}_r \in \mathsf{R}_r\ \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ is satisfied from a tuple $(G, R_1, \ldots, R_r)$ iff there is an assignment of $(G, R_1, \ldots, R_r)$ to a rooted tree and a φ-spanning subtree of this tree such that for every root-to-leaf path of the φ-spanning subtree, the formula ψ is satisfied when interpreting its free variables as the vertices collected in this path (Observation 2).

Patterns and pattern-coloring. A crucial observation that is central to our approach is the following: for every quantifier-free formula $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r) \in \mathsf{FOL{+}DP}$, given a graph G and a tuple of vertices $(v_1, \ldots, v_r)$ of G, the question whether ψ is satisfied when $\mathsf{x}_1, \ldots, \mathsf{x}_r$ are interpreted as $v_1, \ldots, v_r$ boils down to checking whether edges and/or disjoint paths between particular pairs from $v_1, \ldots, v_r$ (indicated by the atomic formulas of ψ) are present in G. For this reason, we define the *pattern* of a boundaried graph $(G, v_1, \ldots, v_r)$ to be an encoding of the edges and the disjoint paths between all pairs in $v_1, \ldots, v_r$ in G. Note that, knowing the pattern of a boundaried graph, we can determine which quantifier-free formulas in $\mathsf{FOL{+}DP}$ are evaluated to true in this graph and which do not. Therefore, when considering a graph assigned in a rooted tree, we can "color" each leaf of the tree by the pattern of the corresponding boundaried graph $(G, v_1, \ldots, v_r)$, where $v_1, \ldots, v_r$ are the vertices collected in the corresponding root-to-leaf path. This, we call the *pattern-coloring* of the leafs. Also, we introduce encoding in terms of patterns for formulas, i.e., we define the pattern of every clause of a quantifier-free formula, that encodes the presence (or not) of predicates or their nagation. This allows to restate Observation 2 and formulate model-chacking as the search of a φ-spanning tree with particular colors in its leafs, inside a leaf-colored tree (to which the annotated graph is assigned) where colors are given by the pattern-coloring (Observation 6). See Figure 1.

Same (recursive) patterns imply satisfaction of the same formulas. Our intuition is that for every graph G, the information encoded by the patterns, organised in a recursive tree structure as described above, is sufficient to evaluate *every* sentence in $\mathsf{FOL{+}DP}$ of a given number of quantifiers. In Subsection 4.4, we formalize this intuition by defining equivalence of leaf-colored trees and proving that two annotated graphs $(G, R_1, \ldots, R_r), (G', R'_1, \ldots, R'_r)$ that have the same recursive patterns for depth r (i.e., $\mathsf{sig}^r(G, R_1, \ldots, R_r) = \mathsf{sig}^r(G', R'_1, \ldots, R'_r)$) satisfy the same sentences in $\mathsf{FOL{+}DP}$ (Lemma 2). The proof of Lemma 2 is an induction on the numbers of quantifiers of a sentence.

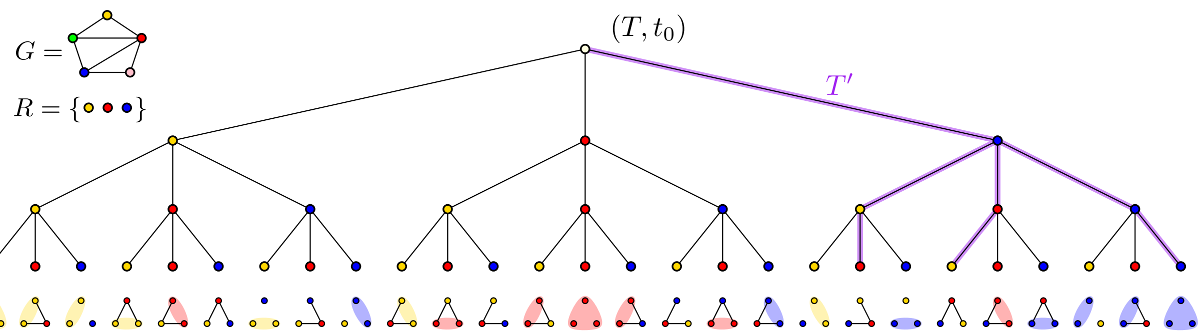

Figure 1: An example of an assignment of an annotated graph (G, R, R, R) (depicted on the upper left part of the figure) to a rooted tree (T, t_0) (in the center of the figure). We use colors for each vertex of G in order to distinguish them, without using indices and we use colors in the nodes of T to show where each vertex of G is mapped to. The small graphs below each leaf of T encode the pattern-coloring, i.e., are the (boundaried) graphs induced by the vertices collected in each root-to-leaf path of T and the linear orderings of their vertex sets (from left to right) follow the ordering of the corresponding path. Vertices that are picked twice in the same path are drawn inside a same-colored bag (respecting the ordering). The subtree T' is a φ-spanning subtree of (T, t_0) for the FOL-sentence $\varphi = \exists \mathsf{x}_1 \in \mathsf{R}_1\ \forall \mathsf{x}_2 \in \mathsf{R}_2\ \exists \mathsf{x}_3 \in \mathsf{R}_3\ (\mathsf{x}_1 = \mathsf{x}_2 \vee \mathsf{E}(\mathsf{x}_2, \mathsf{x}_3))$, that certifies that $(G, R, R, R) \models \varphi$.

We use Lemma 2 in the course of the proof of Theorem 3 in the following way: As long as we can find sets $R'_1, \ldots, R'_r \subseteq V(G)$ such that $R'_i \subseteq R_i$ and $(G, R_1, \ldots, R_r)$ and $(G, R'_1, \ldots, R'_r)$ have the same signature, we can safely reduce the former instance to the latter and report progress. Finding the sets $R'_1, \ldots, R'_r$ is not straightforward and in the next subsection we explain how to deal with this situation. Moreover, in our reduction, we will find a bidimensional annotation-irrelevant area and return a problem-irrelevant vertex v, i.e., a vertex v such that $(G, R_1, \ldots, R_r)$ and $(G \setminus v, R'_1, \ldots, R'_r)$ are also equivalent. Finally, we reduce to an instance $(G', R'_1, \ldots, R'_r)$ with the same r-signature as the input graph (i.e., $\mathsf{sig}^r(G, V(G), \ldots, V(G)) = \mathsf{sig}^r(G', R'_1, \ldots, R'_r)$, where the treewidth of G' depends only on r and the minor we exclude. Then, following the tree-like assignment given by the signature, we can evaluate any FOL+DP sentence of quantifier rank at most r. This latter is done by applying Courcelle's theorem.

2.3 Combinatorial trick to compute folios

Our main difficulty is to compute *recursive* folios of the given annotated graph and obtain a different annotation that gives the same recursive patterns. To tackle this, we need some further tools to handle disjoint paths. From a high-level point of view, our approach considers expressing "partial" patterns inside a bounded treewidth part of the given graph. Using Courcelle's Theorem, we can compute partial patterns and the boundary (tuples of) vertices that give rise to these patterns.

Flat walls with bounded treewidth compasses. In the introduction, we already sketched a definition of flat walls, originating in the work of Robertson and Seymour [193]. An alternative intuition for a flat wall is to see it as a structure made up of (not necessarily planar) pieces, called *flaps*, that are glued together with boundaries of size at most three, in a way that follows the bidimensional structure of a wall. In this article, we use the framework recently introduced in [197] that provides a more accurate view on some previously defined notions on flat walls, particularly in [146] (see Appendix B for formal definitions). In the course of our main algorithm, we use a variant of the Flat Wall Theorem proved in [146, 197] (see Proposition 9) that provides a guarantee that the *compass* of the flat wall has bounded treewidth. This permits us, because of Courcelle's Theorem, to answer in linear time MSOL-queries inside the compass.

In fact, for our arguments, instead of working with a flat wall, we work with a *flat railed annulus* (contained inside the flat wall). This is a structure similar to a flat wall, whose "underlying" structure is not a wall but a sequence of nested cycles transversed by some disjoint paths, called *rails*. See Figure 2.

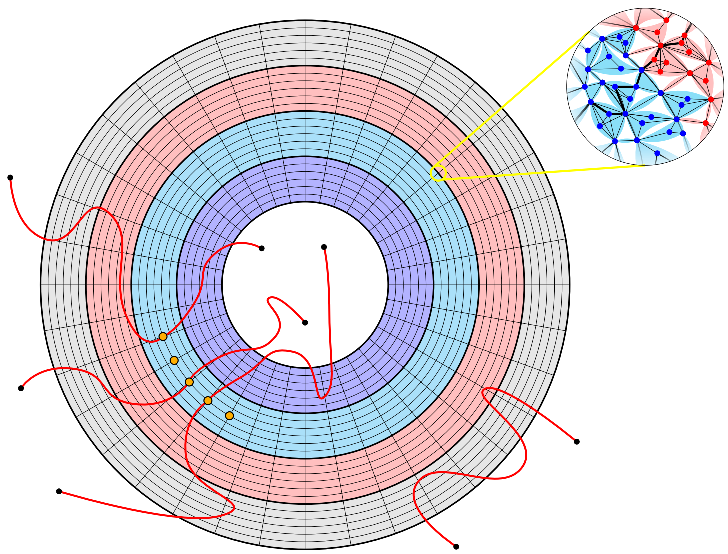

Figure 2: An illustration of a (flat) railed annulus and a linkage (depicted in red) that is combed through some prescribed vertices of the railed annulus (depicted in orange).

Routing linkages through railed annuli. Recall that, the pattern of a boundaried graph $(G, v_1, \ldots, v_r)$, apart from the edges between $v_1, \ldots, v_r$, also encodes the existence or not of disjoint paths between pairs of the boundary vertices $v_1, \ldots, v_r$. We are interested in the ways such a pattern may cross a cycle C of the railed annulus. To define a notion of "partial" pattern (i.e., the one "cropped" by a "central-enough" cycle C), we have to deal with the possibe ways that $v_1, \ldots, v_r$ can be connected through disjoint paths crossing C and this can be seen as a question about the variety of *linkages*, i.e., collections of disjoint paths, that can be routed between boundary vertices and the way they may cross C. Clearly, a linkage may transverse C in a quite entagled way, so that the partial pattern of $(G, v_1, \ldots, v_r)$ cropped by C may have an unbounded number of additional boundary vertices on C. In order to deal with this situation, we make use of a special result for handling linkages, that is the Linkage Combing Lemma (Proposition 2) proved in [103] (which has appeared before in [101]). This result is applied on partially annulus-embedded graphs, i.e., graphs that contain some subgraph K that is embedded in an annulus Δ and Δ separates the two other parts of the graph obtained after removing the vertices inside Δ. The Linkage Combing Lemma intuitively says that in the presence of an annulus-embedded railed annulus $\mathcal{A}$ inside a partially annulus-embedded graph G, every linkage L of G whose terminals are outside the annulus can be "combed", i.e., it can be routed throught the cycle C of $\mathcal{A}$ in a predefined number of vertices. This allows us to *represent* every linkage encoded in the partial pattern by a "combed" linkage with a few predefined additional terminals on C.

Applicability of the Linkage Combing Lemma. To encode the recursive patterns of a graph, we have to keep the pattern of the graph for every choice of the boundary tuple. Therefore, for each tuple of vertices, we have to encode all linkages between pairs of vertices in the tuple. The *trace* of each tuple indicates which (sub)annulus of the flat railed annulus remains "terminal-free" (see Figure 2). This railed annulus is flat but we can consider its *leveling* (see Subsection B.7), that is a *planar representation* of it, obtained by contracting the interior of the flaps to single vertices. We observe that for every linkage whose terminals are outside the compass of the flat railed annulus, there is an equivalent linkage in the graph obtained after replacing the flat railed annulus with its leveling (see Lemma 12). Therefore, the question for linkages between pairs of boundary vertices can be translated to an equivalent one in the partially annulus-embedded graph obtained after considering the leveling (this graph is different for every different trace). This allows for the application of the Linkage Combing Lemma.

Partial patterns. Using the Linkage Combing Lemma we can assume that, for each tuple of vertices, every linkage between these vertices has an appropriate part of the initial (flat) railed annulus where it is combed. This implies that the pattern of a boundaried graph can be "separated" to two parts, the one on the "inner" part and one on the "outer" part of the annulus corresponding to the trace of the boundary tuple. Therefore, we can encode partial patterns by considering boundaried graphs whose boundary vertices are the vertices of the rails where we comb the linkages and some vertices that are on the "inner side" of this annulus. Vertices can also be on the "outer side" of this annulus and thus we have to suitably encode their absence from the tuple (see Subsection 7.3).

Dealing with apices. Having sketched how to route linkages (that correspond to disjoint path queries of $\mathsf{FOL{+}DP}$) in flat walls, it remains to discuss how the presence of apices, that are few vertices that can have neighbors inside the wall in a completely uncontrolled way. The high-level idea here is to define a way to syntactically interpret these (few) apices by adding some extra colors in the structure for the neighborood of each apex vertex, remove the edges between apices and the rest of the graph and translate a disjoint path query in the original graph to some disjoint path queries in the new colored graph, encoding possible ways that the original paths could enter/exit the apex set. This idea originates in [79] for FOL (without annotation) and an analogous version it was given in [81].

Apex-projections of sentences. To define this translation, we first need to define the "projection" of a sentence with respect to an apex-tuple $\mathbf{a}$ (see Subsection 6.1). In fact, first we define the "projection" of the graph G with respect to an apex-tuple $\mathbf{a}$, by removing all edges between the vertices in $\mathbf{a}$ and the rest of the vertices of the graph and "coloring" the neighborhood of each $a \in \mathbf{a}$ by a specific color. This transformation of the graph allows us to "isolate" the apex-tuple and encode its adjacencies with the rest of the graph using colors. In terms of sentences, we define the *apex-projection* φ^l of a sentence $\varphi \in \mathsf{FOL{+}DP}$, where $l = |\mathbf{a}|$, as a sentence that uses these new colors in order to interpret the original sentence φ in the "projected" graph. We stress that the apex-projected sentence φ^l has larger quantifier rank, i.e., we quantify more variables, since one has to guess which (colored) vertices are the entry and exit points of the paths using vertices in the apex-tuple (see Observation 9).

Partial signatures. The next step is to build "meta-collections" of partial patterns. For this, in Subsection 7.4, we define the notion of *partial signatures*. This is a recursive definition, that in the base case is the pattern of a boundaried graph (whose boundary is a tuple of vertices together with the "combing-points" of the railed annulus corresponding to the trace of the given tuple) and every recursive step asks for all possible partial signatures that can be obtained after fixing another (boundary) vertex (inside the annotated set) or its absence. Intuitively, given an annotated graph $(G, R_1, \ldots, R_r)$ that contains a sufficiently big flat railed annulus, the partial signature of G encodes the recursive collection of (partial) patterns in the "inner" part of the railed annulus.

Exchangability of graphs with the same partial signature. After defining partial signatures, we prove that in the presence of a railed annulus flatness pair inside an (annotated) graph $(H, \hat{R}_1, \ldots, \hat{R}_r)$, two (annotated) graphs $(G, R_1, \ldots, R_r)$, $(G', R'_1, \ldots, R'_r)$ that yield the same partial signature when "glued in the inner part" of $(H, \hat{R}_1, \ldots, \hat{R}_r)$, have the same (global) signature when we additionally glue another (annotated) colored graph $(F, R^\star_1, \ldots, R^\star_r)$ in the outer part of $(H, \hat{R}_1, \ldots, \hat{R}_r)$ (see Lemma 5). The main idea of the proof of Lemma 5 is to show that equivalence of partial signatures implies equivalence of (global) signatures. Here, we have to demand equality of partial signatures of "larger depth" than the one that we aim for, in order to be able to deal with the extra quantifiers obtained after apex-projecting the quantifier-free formulas given by the pattern. This translation (to the apex-projection) is essential, in order to obtain a flat structure (without edges to the apiecs) where our rerouting arguments for the linkages can work. In particular, proving equivalency of signatures boils down to formalizing all arguments mentioned in the above paragraph that use the Linkage Combing Lemma, using a double inductive argument, in order to deal with the recursive nature of the definitions. The main importance of Lemma 5 is that it guarantees exchangeability (in terms of signatures) of graphs with the same partial signature and allows to shift our pursue of graphs of same signature to pursue of graphs of same partial signature.

Computing representatives. Since the partial signature of an annotated graph depends only on adjacency and linkage questions inside the "inner" part of a given flat railed annulus of the given graph, we can express the partial signature of the annotated graph in MSOL. Also, as we mentioned before, we can ask the part of the graph, where we want to compute partial signature, to have bounded treewidth. Therefore, we can define equivalence between vertices of the same partial signatures and using Courcelle's Theorem we can compute *representatives* of these equivalence relations (see Subsection 8.1). Therefore, given an annotated graph $(G, R_1, \ldots, R_r)$, an apex-tuple $\mathbf{a}$ of G, and a "big enough" flat railed annulus of $G \setminus V(\mathbf{a})$, where $V(\mathbf{a})$ is the set of vertices in $\mathbf{a}$, that "crops" a bounded treewidth graph, we can compute sets $R'_1, \ldots, R'_r$ such that $R'_i \subseteq R_i$ and the size of the intersection of R'_i with the "left cropped part" of the railed annulus is upper-bounded by a function of k, and moreover $(G, R_1, \ldots, R_r)$ and $(G, R'_1, \ldots, R'_r)$ have the same partial signature (see Lemma 6). Therefore, in the presence of a big enough flat wall (of bounded treewidth) inside an annotated graph, we can bound the number of annotated vertices in the inner part of the railed annulus, which, in turn, permits us to declare annotation-irrelevant the vertices of some "central" part of the flat wall that contains the railed annulus (Lemma 7).

Wrapping-up the proof of Theorem 3. As mentioned in the beginning of this section, the most demanding part of our proof is reducing the annotation set, i.e., to show Lemma 7. Then, given that the sets $R_1, \ldots, R_r$, in which we build the signatures, leave some big enough bidimensional area "annotation-irrelevant", it remains to compute a vertex inside this area whose removal does not affect the existence (or not) of any linkage between the interpretations of the variables in the definition of signature. This is done using known irrelevant vertex technique arguments (see Subsection 8.2 and in particular Lemma 8). Therefore, we can find a vertex that is irrelevant for the signature (Lemma 9). This produces an equivalent instance $(G', R'_1, \ldots, R'_r)$ as required for the application of the main procedure of our algorithm.

From disjoint to scattered paths. To deal with FOL+SDP and prove Theorem 4, we need a Linkage Combing Lemma for scattered linkages. Such kind of a result is known when we restrict ourselves to graphs of bounded Euler genus and is proved in [103] (Proposition 5; see also [138, 171]). Then, using this result, we follow the same approach as in the proof of Theorem 3, and we prove tractability of model-checking for FOL+SDP for graphs of embedded in some fixed surface (Theorem 4). We refer the reader to Section 9 for more details.

3 Preliminaries

In this section we present some basic definitions and we state our main result (Theorem 3). In Subsection 3.1, we start with some definitions on graphs and in Subsection 3.2 we define first-order and monadic second-order logic. Then, in Subsection 3.3 we define the extension of FOL with disjoint paths predicates and we state our main result (Theorem 3).

3.1 Basic definitions

Integers, sets, and tuples. We denote by $\mathbb{N}$ the set of non-negative integers. Given two integers p and q, the set $[p,q]$ refers to the set of every integer r such that $p \leq r \leq q$. For an integer $p \geq 1$, we set $[p] = [1,p]$ and $\mathbb{N}_{\geq p} = \mathbb{N} \setminus [0, p-1]$. Given a non-negative integer x, we denote by $\mathsf{odd}(x)$ the minimum odd number that is not smaller than x. For a set S, we denote by 2^S the set of all subsets of S and, given an integer $r \in [|S|]$, we denote by $\binom{S}{r}$ the set of all subsets of S of size r. Given two sets A, B and a function $f : A \to B$, for a subset $X \subseteq A$ we use $f(X)$ to denote the set $\{f(x) \mid x \in X\}$. Given a set X and an $r \in \mathbb{N}$, we use X^r to denote the product $X \times \cdots \times X$ of r copies of X.

Let $\mathcal{S}$ be a collection of objects where the operations $\cup$ and $\cap$ are defined. Given two tuples $\mathbf{x} = (x_1, \ldots, x_l)$ and $\mathbf{y} = (y_1, \ldots, y_l)$, where $x_i, y_i \in \mathcal{S}$, we denote $\mathbf{x} \cup \mathbf{y} = (x_1 \cup y_1, \ldots, x_l \cup y_l)$ and $\mathbf{x} \cap \mathbf{y} = (x_1 \cap y_1, \ldots, x_l \cap y_l)$. Also, we denote $\bigcup \mathcal{S} = \bigcup_{X \in \mathcal{S}} X$.

Basic concepts on graphs. All graphs considered in this paper are undirected, finite, and without loops or multiple edges. We use standard graph-theoretic notation and we refer the reader to [63] for any undefined terminology. Let G be a graph. Given a vertex $v \in V(G)$, we denote by $N_G(v)$ the set of vertices of G that are adjacent to v in G. For $S \subseteq V(G)$, we set $G[S] = (S, E \cap \binom{S}{2})$ and use the shortcut $G \setminus S$ to denote $G[V(G) \setminus S]$. The *length* of a path P of G is the number of edges of P.

Minors. The *contraction* of an edge $e = \{u, v\}$ of a simple graph G results in a simple graph G' obtained from $G \setminus \{u, v\}$ by adding a new vertex uv adjacent to all the vertices in the set $N_G(u) \cup N_G(v) \setminus \{u, v\}$. A graph G' is a *minor* of a graph G, denoted by $G' \preceq_{\mathsf{m}} G$, if G' can be obtained from G by a sequence of vertex removals, edge removals, and edge contractions. Given a finite collection of graphs $\mathcal{F}$ and a graph G, we use the notation $\mathcal{F} \preceq_{\mathsf{m}} G$ to denote that some graph in $\mathcal{F}$ is a minor of G. Given a set of graphs $\mathcal{F}$, we denote by $\mathsf{Excl}(\mathcal{F})$ the set containing every graph that excludes all graphs in $\mathcal{F}$ as minors. A graph class $\mathcal{G}$ is *minor-closed* if every minor of a graph in $\mathcal{G}$ is also a member of $\mathcal{G}$. The *Hadwiger number* of a graph G, denoted by $\mathsf{hw}(G)$, is the minimum k where $G \in \mathsf{Excl}(\{K_k\})$ and K_k is the complete graph on k vertices. A minor-closed graph class is called *proper* if it is not the class of all graphs.

Treewidth. A *tree decomposition* of a graph G is a pair (T, χ) where T is a tree and $\chi : V(T) \to 2^{V(G)}$ such that

- $\bigcup_{t \in V(T)} \chi(t) = V(G)$,
- for every edge e of G there is a $t \in V(T)$ such that $\chi(t)$ contains both endpoints of e, and
- for every $v \in V(G)$, the subgraph of T induced by $\{t \in V(T) \mid v \in \chi(t)\}$ is connected.

The *width* of (T, χ) is equal to $\max\left\{ |\chi(t)| - 1 \;\middle|\; t \in V(T)\right\}$ and the *treewidth* of G is the minimum width over all tree decompositions of G.

3.2 First-order logic and monadic second-order logic

In this subsection, we give the definition of first-order and monadic second-order logic. We define these logics on *relational* vocabularies with constant symbols and we work with structures of these vocabularies.

Structures. A *vocabulary* is a finite set of relation and constant symbols (we do not use function symbols). Every relation symbol R is associated with a positive integer that is called the *arity* of R, which we denote $\mathsf{ar}(\mathsf{R})$. A *structure* $\mathfrak{A}$ *of vocabulary* τ, in short a τ*-structure*, consists of a non-empty set $V(\mathfrak{A})$, called the *universe* of $\mathfrak{A}$, an r-ary relation $\mathsf{R}^{\mathfrak{A}} \subseteq V(\mathfrak{A})^r$ for each relation symbol $\mathsf{R} \in \tau$ of arity $r \geq 1$, and an element[6] $\mathsf{c}^{\mathfrak{A}} \in \{\text{\textvisiblespace}\} \cup V(\mathfrak{A})$ for each constant symbol $\mathsf{c} \in \tau$. We refer to $\mathsf{R}^{\mathfrak{A}}$ (resp. $\mathsf{c}^{\mathfrak{A}}$) as the *interpretation of the symbol* R *(resp.* c*) in the structure* $\mathfrak{A}$. A structure $\mathfrak{A}$ is *finite* if its universe $V(\mathfrak{A})$ is a finite set. We denote by $\mathbb{STR}[\tau]$ the set of all finite τ-structures.

We say that a τ-structure $\mathfrak{A}$ is *isomorphic* to a τ-structure $\mathfrak{B}$ if there is a bijection $V(\mathfrak{A}) \cup \{\text{\textvisiblespace}\}$ to $V(\mathfrak{B}) \cup \{\text{\textvisiblespace}\}$, such that $\pi(\text{\textvisiblespace}) = \text{\textvisiblespace}$ and for every $k \geq 1$, every relation symbol $\mathsf{R} \in \tau$ of arity k, and every $(a_1, \ldots, a_k) \in V(\mathfrak{A})^k$, it holds that $(a_1, \ldots, a_k) \in \mathsf{R}^{\mathfrak{A}} \iff (\pi(a_1), \ldots, \pi(a_k)) \in \mathsf{R}^{\mathfrak{B}}$ and for every constant symbol $\mathsf{c} \in \tau$, it holds that $\pi(\mathsf{c}^{\mathfrak{A}}) = \mathsf{c}^{\mathfrak{B}}$.

Given two τ-structures $\mathfrak{A}$ and $\mathfrak{B}$, where for every constant symbol $\mathsf{c} \in \tau$ either $\mathsf{c}^{\mathfrak{A}} = \mathsf{c}^{\mathfrak{B}}$ or $\mathsf{c}^{\mathfrak{A}} = \text{\textvisiblespace} \vee \mathsf{c}^{\mathfrak{B}} = \text{\textvisiblespace}$, we define the *disjoint union* of $\mathfrak{A}$ and $\mathfrak{B}$, and we denote it by $\mathfrak{A} \dot\cup \mathfrak{B}$, as the τ-structure where $V(\mathfrak{A} \dot\cup \mathfrak{B})$ is the disjoint union of $V(\mathfrak{A}) \setminus \{\text{\textvisiblespace}\}$, $V(\mathfrak{B}) \setminus \{\text{\textvisiblespace}\}$ and $\{\text{\textvisiblespace}\}$, for every relation symbol $\mathsf{R} \in \tau$, $\mathsf{R}^{\mathfrak{A} \dot\cup \mathfrak{B}} = \mathsf{R}^{\mathfrak{A}} \cup \mathsf{R}^{\mathfrak{B}}$, and for every constant symbol $\mathsf{c} \in \tau$, if $\mathsf{c}^{\mathfrak{A}} = \mathsf{c}^{\mathfrak{B}}$, then $\mathsf{c}^{\mathfrak{A} \dot\cup \mathfrak{B}} = \mathsf{c}^{\mathfrak{A}} = \mathsf{c}^{\mathfrak{B}}$, and if $\mathsf{c}^{\mathfrak{A}} = \text{\textvisiblespace}$ (resp. $\mathsf{c}^{\mathfrak{B}} = \text{\textvisiblespace}$), then $\mathsf{c}^{\mathfrak{A} \dot\cup \mathfrak{B}} = \mathsf{c}^{\mathfrak{B}}$ (resp. $\mathsf{c}^{\mathfrak{A} \dot\cup \mathfrak{B}} = \mathsf{c}^{\mathfrak{A}}$).

An undirected graph without loops can be seen as an $\{\mathsf{E}\}$-structure $\mathfrak{G} = (V(\mathfrak{G}), \mathsf{E}^{\mathfrak{G}})$, where $\mathsf{E}^{\mathfrak{G}}$ is a binary relation that is symmetric and anti-reflexive.

The *Gaifman graph* $G_{\mathfrak{A}}$ of a τ-structure $\mathfrak{A}$ is the graph whose vertex set is $V(\mathfrak{A})$ and two vertices x, y are adjacent if there is an $\mathsf{R} \in \tau$ and a $\bar{v} \in \mathsf{R}^{\mathfrak{A}}$ such that both x and y are elements of $\bar{v}$.

First-order logic and monadic second-order logic. We now define the syntax and the semantics of first-order logic and monadic second-order logic of a vocabulary τ. We assume the existence of a countable infinite set of *first-order variables*, usually denoted by lowercase symbols $\mathsf{x}_1, \mathsf{x}_2, \ldots$, and of a countable infinite set of *set variables*, usually denoted by uppercase symbols $\mathsf{X}_1, \mathsf{X}_2, \ldots$. A *first-order term* is either a first-order variable or a constant symbol. A *first-order logic formula*, in

[6]We stress that we allow constant symbols to be interpreted as the element $\text{\textvisiblespace}$, where $\text{\textvisiblespace}$ is an element that is not in $V(\mathfrak{A})$. Throughout this paper, we assume that the universe of every given structure is extended by adding the extra element $\text{\textvisiblespace}$, while all relation symbols are interpreted as tuples of elements of $V(\mathfrak{A})$, not containing $\text{\textvisiblespace}$. Moreover, we assume that for every formula that we consider, quantified first-order variables are interpreted as elements of the original universe of the structure (and not $\text{\textvisiblespace}$).

short FOL-*formula*, of vocabulary τ is built from atomic formulas $\mathsf{x} = \mathsf{y}$ and $(\mathsf{x}_1, \ldots, \mathsf{x}_r) \in \mathsf{R}$, where $\mathsf{R} \in \tau$ and has arity $r \geq 1$, on first-order terms $\mathsf{x}, \mathsf{y}, \mathsf{x}_1, \ldots, \mathsf{x}_r$, by using the logical connectives $\vee$, $\wedge$, $\neg$ and the quantifiers $\forall, \exists$ on first-order variables. We denote by $\mathsf{FOL}[\tau]$ the set of all FOL-formulas of vocabulary τ.

A *monadic second-order logic formula*, in short MSOL-*formula*, of vocabulary τ is obtained by enhancing the syntax of FOL-formulas by allowing the atomic formulas $\mathsf{x} \in \mathsf{X}$, for some first-order term x and some set variable X, and allowing quantification on both first-order and set variables. We denote by $\mathsf{MSOL}[\tau]$ the set of all MSOL-formulas of vocabulary τ. We make clear that what we call here MSOL is what is commonly referred in the literature as MSO_1, in which, for the vocabulary of graphs, first-order variables are interpreted as vertices and set variables are interpreted as sets of vertices.

The formulas in $\mathsf{FOL}[\tau]$ and $\mathsf{MSOL}[\tau]$ are evaluated on τ-structures by interpreting every symbol in τ as its interpretation in the structure and every first-order (resp. set) variable as an element (resp. set of elements) of the universe of the structure. Given a formula φ, the *free variables* of φ are its variables that are not in the scope of any quantifier. We write $\varphi(\mathsf{x}_1, \ldots, \mathsf{x}_k)$ to indicate that the free variables of the formula φ are $\mathsf{x}_1, \ldots, \mathsf{x}_k$. A *sentence* is a formula without free variables. Let $\varphi = Q_1\mathsf{x}_1 \ldots Q_r\mathsf{x}_r\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ be a sentence, where for each $i \in [r]$, $Q_i \in \{\forall, \exists\}$, $\mathsf{x}_1, \ldots, \mathsf{x}_r$ are first-order variables, and $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ is a quantifier-free formula with free variables $\mathsf{x}_1, \ldots, \mathsf{x}_r$. We call r the *quantifier rank* of φ.

Given a τ-structure $\mathfrak{A}$, a formula $\varphi(\mathsf{x}_1, \ldots, \mathsf{x}_k) \in \mathsf{FOL}[\tau]$, and $a_1, \ldots, a_k$ in $V(\mathfrak{A})$, we write $\mathfrak{A} \models \varphi(a_1, \ldots, a_k)$ to denote that $\varphi(\mathsf{x}_1, \ldots, \mathsf{x}_k)$ holds in $\mathfrak{A}$ if, for every $i \in [k]$, the variable x_i is interpreted as a_i. Given a $k \in \mathbb{N}$, two formulas $\varphi(\mathsf{x}_1, \ldots, \mathsf{x}_k)$, $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_k) \in \mathsf{FOL}[\tau]$ are *equivalent* if for every τ-structure $\mathfrak{A}$ and every $a_1, \ldots, a_k \in V(\mathfrak{A})$, we have $\mathfrak{A} \models \varphi(a_1, \ldots, a_k) \iff \mathfrak{A} \models \psi(a_1, \ldots, a_k)$. We call the set $\{\mathfrak{A} \in \mathbb{STR}[\tau] \mid \mathfrak{A} \models \varphi\}$ the set of *models of* φ and we denote it by $\mathrm{Mod}(\varphi)$.

3.3 Disjoint paths logic

For the rest of this paper, we deal with colored graphs, i.e., we fix a vocabulary τ that contains a binary relation symbol E that is always interpreted as a symmetric and anti-reflexive binary relation (corresponding to the edges of the graph) and a collection of unary relation symbols $\mathsf{Y}_1, \ldots, \mathsf{Y}_h$ (corresponding to colors on the vertices of the graph). We call such a vocabulary a *colored-graph vocabulary.* Also, we always assume that the interpretations of $\mathsf{Y}_1, \ldots, \mathsf{Y}_c$ are always pairwise disjoint (if not, then we introduce extra unary relations symbols for each intersection).

Disjoint-paths logic. We define the $2k$-ary predicate $\mathsf{dp}_k(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_k, \mathsf{y}_k)$, which evaluates true in a τ-structure $\mathfrak{G}$ if and only if there are paths $P_1, \ldots, P_k$ of $(V(\mathfrak{G}), \mathsf{E}^{\mathfrak{G}})$ of length at least 2 between (the interpretations of) x_i and y_i for all $i \in [k]$ such that for every $i, j \in [k]$, $i \neq j$, $V(P_i) \cap V(P_j) = \emptyset$. We let $\tau + \mathsf{dp} := \tau \cup \{\mathsf{dp}_k \mid k \geq 1\}$, where each dp_k is a $2k$-ary relation symbol. We use dp instead of dp_k when k is clear from the context. Our main result is the following (slightly more general) version of Theorem 1:

Theorem 3. *For every colored-graph vocabulary τ and every sentence $\varphi \in \mathsf{FOL}[\tau + \mathsf{dp}]$, there exists an algorithm that, given a τ-structure G of size n, outputs whether $G \models \varphi$ in time $\mathcal{O}_{|\varphi|, \mathbf{hw}(G)}(n^2)$.*

4 An alternative view to first-order logic model-checking

In this section we present an alternative way to interpret first-order logic model-checking, by embedding a given graph to a (rooted) tree. In Subsection 4.1, we translate the problem of whether a graph satisfies a formula φ to the search of a subtree of the tree in which the graph is embedded, such that the bifurcations of this subtree correspond to the quantifiers of φ and the vertices collected in each root-to-leaf path satisfy the quantifier-free "tail" of φ. We also formally define the notion of *signature* of a graph that encodes in a tree-like way how tuples of vertices satisfy quantifier-free formulas of $\mathsf{FOL}[\tau{+}\mathsf{dp}]$. In Subsection 4.2, we define the notion of *pattern* of a colored graph $\mathfrak{G}$ together with some vertices $v_1, \ldots, v_r \in V(\mathfrak{G})$. This notion is used to encode the way all quantifier-free $\mathsf{FOL}[\tau{+}\mathsf{dp}]$-formulas can be satisfied by $\mathfrak{G}$ when interpreting the variables as $v_1, \ldots, v_r$. This way, we also define the respective notion of patterns of quantifier-free $\mathsf{FOL}[\tau{+}\mathsf{dp}]$-formulas and, in Subsection 4.3, we formulate model-checking in these terms (Observation 6). Finally, in Subsection 4.4, we prove the main result of this section (Lemma 2), which intuitively states that two graphs that have the same signatures (or, in other words, the same recursive patterns) satisfy the same sentences.

4.1 Embedding model-checking to trees

In this subsection we present a way to embed graphs to trees and how to trace the satisfaction of a sentence from the given graph, in terms of subtrees of the original tree.

Rooted trees. Let (T, t_0) be a rooted tree. Given a node x of T, we denote by T_x the subtree of T rooted at x. We use $L(T)$ to denote the leaves of T. For every $i \in [0, r]$, where $r \in \mathbb{N}$ is the height of T, we use $D_i(T)$ to denote the set of nodes of T that are at distance i from t_0. We say that a node $t \in V(T)$ has *depth* i if $t \in D_i(T), i \in [0, r]$. In this paper, for every rooted tree (T, t_0) of height r that we consider, we assume that $L(T) = D_r(T)$. We use $\mathsf{Paths}(T)$ to denote the set of all root-to-leaf paths of T. We denote by $\mathsf{children}_T(t)$ the set of all children of t in T. Also, when it is clear from the context, we use $(t_0, \ldots, t_r)$ to denote a path $P \in \mathsf{Paths}(T)$ such that $V(P) = \{t_0, \ldots, t_r\}$ and $E(P) = \bigcup_{i \in [0, r-1]} \{\{t_i, t_{i+1}\}\}$. A rooted tree (T, t_0) of height r is called *n-ary*, if each node of depth at most $r - 1$ has n children.

Trees expressing quantification of formulas. We now express how, given a rooted tree (T, t_0) and a sentence φ, where the height of the tree and the quantifier rank of φ are the same, use the quantifier alternation of φ to construct a subtree T' of T. This is done in the following recursive way: the subtree starts from t_0 and, for every $t \in V(T')$, if the considered quantifier of φ is the universal one, then T' spans to *all* children of t, while if we have the existential one, T' arbitrarily choses *one* child of t. We proceed to formalize this idea.

Let τ be a colored-graph vocabulary. Let $\varphi = Q_1\mathsf{x}_1 \ldots Q_r\mathsf{x}_r\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ be an $\mathsf{FOL}[\tau + \mathsf{dp}]$-sentence and let (T, t_0) be a rooted tree of height r (where each root-to-leaf path has $r + 1$ nodes). A *φ-spanning triple* of (T, t_0) is a triple $(T, t_0, \mathcal{U}_\varphi)$ such that $\mathcal{U}_\varphi = \{U_0, U_1, \ldots, U_r\}$ where $U_0 = \{t_0\}$ and for every $i \in [r]$, U_i is a subset of $D_i(T)$ such that

- if $Q_i = \forall$, then for each node $v \in U_{i-1}$, we add in U_i **all** nodes $u \in \mathsf{children}_T(v)$ and
- if $Q_i = \exists$, then for each node $v \in U_{i-1}$, we add in U_i **one** node $u \in \mathsf{children}_T(v)$.

A *φ-spanning subtree* of T is a rooted subtree (T', t_0) of (T, t_0) where $T' = T[\bigcup_{i=0}^{r} U_i]$ for a φ-spanning triple $(T, t_0, \{U_0, U_1, \ldots, U_r\})$ of (T, t_0). See Figure 3. Note that, given a rooted tree (T, t_0) of height zero and a quantifier-free formula ψ, (T, t_0) is the unique ψ-spanning subtree of T.

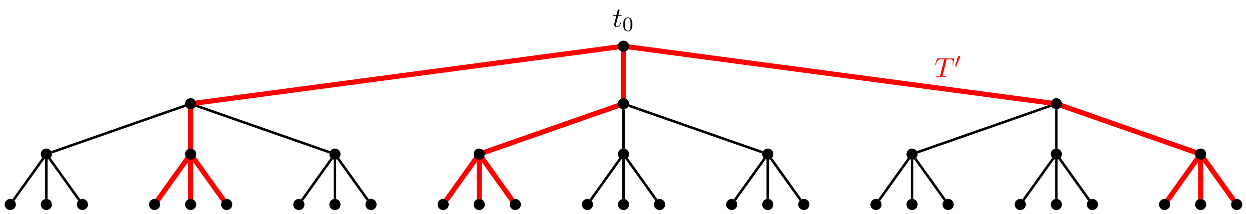

Figure 3: A rooted tree (T, t_0) and a φ-spanning subtree T' of T (depicted in red), for a sentence $\varphi = \forall \mathsf{x}_1 \exists \mathsf{x}_2 \forall \mathsf{x}_3 \psi(\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3)$.

Signatures of structures. Let τ be the vocabulary of graphs of h colors and ℓ roots (i.e., constants). We use $\Psi^{r,h,\ell}_{\mathsf{FOL}[\tau+\mathsf{dp}]}$ to denote the set of all quantifier-free $\mathsf{FOL}[\tau+\mathsf{dp}]$-formulas with r free variables. We treat equivalent formulas as equal (and choose one representative for each equivalence class, which is possible for quantifier-free formulas). Then the size of $\Psi^{r,h,\ell}_{\mathsf{FOL}[\tau+\mathsf{dp}]}$ is upper-bounded by some constant depending only on r, h, and ℓ. Let $r \in \mathbb{N}$, let $\mathfrak{G}$ be a τ-structure, let $R_1, \ldots, R_r \subseteq V(\mathfrak{G})$. We set $\bar{R} = (R_1, \ldots, R_r)$. The *atomic type* of a tuple $(v_1, \ldots, v_r) \in (V(G) \cup \{\_\})^r$ is the set of all atomic formulas that are true for $(v_1, \ldots, v_r)$ in $\mathfrak{G}$. Given $(v_1, \ldots, v_r) \in (V(G) \cup \{\_\})^r$, we define $\mathsf{sig}^0(\mathfrak{G}, \bar{R}, v_1, \ldots, v_r)$ to be the atomic type of $(v_1, \ldots, v_r)$. Also, for each $i \in [r-1]$ and every $v_1, \ldots, v_{r-i} \in V(G) \cup \{\_\}$, we define

$$\mathsf{sig}^i(\mathfrak{G}, \bar{R}, v_1, \ldots, v_{r-i}) = \big\{\mathsf{sig}^{i-1}(\mathfrak{G}, \bar{R}, v_1, \ldots, v_{r-i}, u) \mid u \in R_{r-i+1} \cup \{\_\}\big\}$$

Finally, we define

$$\mathsf{sig}^r(\mathfrak{G}, \bar{R}) = \big\{\mathsf{sig}^{r-1}(\mathfrak{G}, \bar{R}, v) \mid v \in R_1 \cup \{\_\}\big\}.$$

In the case where $\bar{R} = V(\mathfrak{G})^r$, we omit $\bar{R}$ from the notation, i.e., we write $\mathsf{sig}^i(\mathfrak{G}, v_1, \ldots, v_{r-i})$ and $\mathsf{sig}^r(\mathfrak{G})$ instead of $\mathsf{sig}^i(\mathfrak{G}, \bar{R}, v_1, \ldots, v_{r-i})$ and $\mathsf{sig}^r(\mathfrak{G}, \bar{R})$. Also, we set $\mathcal{B}^{r,h,\ell}_{\mathsf{sig}} := \{\mathsf{sig}^r(\mathfrak{G}) \mid \mathfrak{G} \in \mathbb{STR}[\tau]\}$.

Observation 1. *Let τ be the vocabulary of graphs of h colors and ℓ roots and let $r \in \mathbb{N}$. For every $\beta \in \mathcal{B}^{r,h,\ell}_{\mathsf{sig}}$, there is a $\varphi_\beta \in \mathsf{FOL}[\tau+\mathsf{dp}]$ such that for every τ-structure $\mathfrak{G}$, $\mathsf{sig}^r(\mathfrak{G}) = \beta$ if and only if $\mathfrak{G} \models \varphi_\beta$.*

Assigning graphs to rooted trees. Let $r \in \mathbb{N}$. Given a graph G, we construct a rooted tree (T, t_0) (starting from a single root t_0) and a function $\lambda : V(T) \setminus \{t_0\} \to V(G)$ as follows. First, for every $\alpha \in \mathsf{sig}^r(\mathfrak{G})$, we consider a vertex $v_\alpha \in V(G)$ such that $\mathsf{sig}^{r-1}(\mathfrak{G}, v) = \alpha$. We add $|\mathsf{sig}^r(\mathfrak{G})|$ children to t_0, while we define λ_{t_0} to be a bijection from $\mathsf{children}_T(t_0)$ to $\{v_\alpha \mid \alpha \in \mathsf{sig}^r(\mathfrak{G})\}$. Then, for every $t \in V(T)$, if $t_0, t_1, \ldots, t_d = t$ is a path of T and $(u_1, \ldots, u_d) = \lambda_{t_0}(t_1), \ldots, \lambda_{t_{d-1}}(t_d))$, then we add $|\mathsf{sig}^{r-d}(\mathfrak{G}, u_1, \ldots, u_d)|$-many children to t. Also, for every $\alpha \in \mathsf{sig}^{r-d}(\mathfrak{G}, u_1, \ldots, u_d)$, we consider a vertex $v_\alpha \in V(G)$ such that $\mathsf{sig}^{r-d-1}(\mathfrak{G}, u_1, \ldots, u_d, v) = \alpha$ and we define λ_t to be a bijection from $\mathsf{children}_T(t)$ to $\{v_\alpha \mid \alpha \in \mathsf{sig}^{r-d}(\mathfrak{G}, u_1, \ldots, u_d)\}$. We also define the function $\lambda : V(T) \setminus \{t_0\} \to V(G)$ such that for every $t \in V(T) \setminus \{t_0\}$ $\lambda(t) = \lambda_{t'}(t)$, where t' is the parent of t. We call λ an *assignment of G to (T, t_0)*.

We conclude this subsection by formulating model-checking of sentences in $\mathsf{FOL}[\tau+\mathsf{dp}]$ in terms of assignments and φ-spanning trees.

Observation 2. *Let $r \in \mathbb{N}$. Let τ be a colored rooted graph vocabulary. For every τ-structure $\mathfrak{G}$ and every sentence $\varphi = Q_1\mathsf{x}_1 \ \ldots Q_r\mathsf{x}_r \ \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ where $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r) \in \mathsf{FOL}[\tau+\mathsf{dp}]$, we have that $\mathfrak{G} \models \varphi$ if and only if there is*

- *an assignement λ of G to rooted tree (T, t_0) of height r and*

- *a φ-spanning subtree (T', t_0) of (T, t_0)*

such that for every $(t_0, t_1, \ldots, t_r) \in \mathsf{Paths}(T')$, it holds that $(\mathfrak{G}, \lambda(t_1), \ldots, \lambda(t_r)) \models \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$, where $\mathsf{x}_1, \ldots, \mathsf{x}_r$ are interpreted as $\lambda(t_1), \ldots, \lambda(t_r)$.

We say that a φ-spanning tree (T', t_0) as above *certifies* that $\mathfrak{G} \models \varphi$.

4.2 Patterns of boundaried colored graphs

In this subsection, we define the notion of a *pattern* of a colored rooted graph. The pattern aims to encode all information of this colored graph that concern the boundary vertices: 1) if some elements of the "boundary" tuple are the same, 2) which boundary vertices are root-vertices, 3) what are the colors of boundary vertices, 4) what is the graph induced by the boundary vertices, and 5) which sets of pairs of boundary vertices can be connected with internally vertex-disjoint paths of lenght at least two. In other words, the pattern is defined in a way that, having it in hand, we can check which quantifier-free formulas of $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$ are satisfied from $(\mathfrak{G}, \mathbf{a})$ if we interpret their free variables as the boundary vertices.

Apex-tuples of structures. Let τ be a vocabulary, let $\mathfrak{A}$ be a τ-structure, and let $l \in \mathbb{N}$. A tuple $\mathbf{a} = (a_1, \ldots, a_l)$ where each a_i is either an element of $V(\mathfrak{A})$ or $\_$, is called a *apex-tuple* of $\mathfrak{A}$ of size l. We use $V(\mathbf{a})$ for the set containing the non-$\_$ elements in $\mathbf{a}$. Also, if $S \subseteq V(\mathfrak{A})$, we define $\mathbf{a} \cap S = (a'_1, \ldots, a'_l)$ so that if $a_i \in S$, then $a'_i = a_i$, and otherwise $a'_i = \_$. We also define $\mathbf{a} \setminus S = \mathbf{a} \cap (V(\mathfrak{A}) \setminus S)$. For every apex-tuple $\mathbf{a}$, we always assume that all non-$\_$ elements in $\mathbf{a}$ are distinct.

Intuitively, an apex-tuple of a graph is a tuple consisting of vertices and empty entries and every choice of vertices (and empty entries) can be seen as an apex-tuple of appropriate size. All definitions and results in this subsection are stated for general apex-tuples, although in our proofs, we will consider apex-tuples of a particular type, i.e., arbitrary orderings of the apex sets A given by the algorithmic version of the Flat Wall Theorem (Proposition 9) presented in Appendix B.

Boundaried colored graphs. Let $t, h, l \in \mathbb{N}$. A *t-boundaried (h, l)-colored graph* is a tuple $\mathbf{G} = (G, X_1, \ldots, X_h, \mathbf{a}, v_1, \ldots, v_t)$ where $(G, X_1, \ldots, X_h)$ is a colored graph, $\mathbf{a}$ is an apex-tuple of G of size l, and $v_1, \ldots, v_t \in V(G) \cup \{\_\}$. Intuitively, we use $\_$ to encode the absence of a vertex of G of a certain index. The vertices $v_i, i \in [t]$ that belong to $V(G)$ are called *boundary vertices*.

Given two t-boundaried (h, l)-colored graphs $\mathbf{G}_1 = (G_1, X_1, \ldots, X_h, \mathbf{a}, v_1, \ldots, v_t)$ and $\mathbf{G}_2 = (G_2, X'_1, \ldots, X'_h, \mathbf{a}', u_1, \ldots, u_t)$, we say that $\mathbf{G}_1$ and $\mathbf{G}_2$ are *isomorphic* if G_1 is isomorphic to G_2 via a bijection $\eta : V(G_1) \cup \{\_\} \to V(G_2) \cup \{\_\}$ such that $\eta(\_) = \_$, for every $i \in [t]$, $\eta(v_i) = u_i$, for every $i \in [l]$, $\eta(a_i) = \eta(a'_i)$, and for every $i \in [h]$, $\eta(X_i) = X'_i$. We denote by $\mathcal{B}^{(t,h,l)}$ the set of all (pairwise non-isomorphic) t-boundaried (h, l)-colored graphs and we set $\mathcal{B}^{(h,l)} = \bigcup_{t \in \mathbb{N}} \mathcal{B}^{(t,h,l)}$.

Patterns. Let $(G, X_1, \ldots, X_h, \mathbf{a}, v_1, \ldots, v_r) \in \mathcal{B}^{(r,h,l)}$. Given a set $I \subseteq [r]$, we denote

$$\mathsf{Ind}_G(I) = (I, \{\{a, b\} \in I \times I \mid \{v_a, v_b\} \in E(G)\}).$$

We set $I = \{i \in [r] \mid v_i \neq \_\}$. We define the *pattern* of $(G, X_1, \ldots, X_h, \mathbf{a}, v_1, \ldots, v_r)$, denoted by $\mathsf{pattern}(G, X_1, \ldots, X_h, \mathbf{a}, v_1, \ldots, v_r)$, to be the quintuple $(V, \kappa, \delta, H^e, \mathcal{H}^P)$, where

- V is the partition of I into sets such that for every $A \in I$ and every $i, j \in A$, $v_i = v_j$,

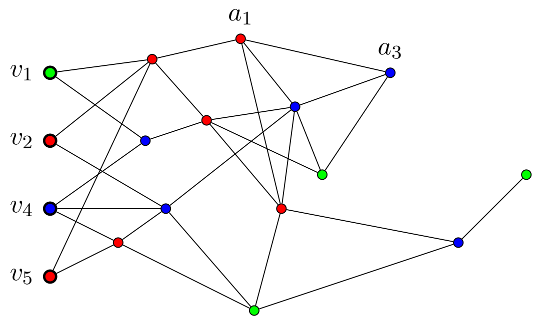

Figure 4: An example of a 5-boundaried $(3, 3)$-colored graph $(G, X_1, X_2, X_3, a_1, \llcorner\lrcorner, a_3, v_1, v_2, \llcorner\lrcorner, v_4, v_5)$, where X_1 is the set of vertices depicted in red, X_2 is the set of vertices depicted in blue, and X_3 is the set of vertices depicted in green.

- $\kappa : I \to [l]$ is the partial function mapping each $i \in I$ to the integer $j \in [l]$ such that $v_i = a_j$,
- $\delta : I \to [h]$ is the partial function mapping each $i \in I$ to the integer $j \in [h]$ such that $v_i \in X_j$,
- $H^e = \mathsf{Ind}_G(I)$, and
- $\mathcal{H}^P = \bigcup_{E \subseteq I \times I} \left\{ (I, E) \;\middle|\; \begin{array}{c} G \text{ contains vertex-disjoint paths of length at least} \\ \text{two between the vertices } v_i, v_j \text{ for all } \{i, j\} \in E \end{array} \right\}$.

Intuitively, the pattern of $(G, X_1, \ldots, X_h, v_1, \ldots, v_r)$ encodes a partition V of the set I formed by grouping indices that correspond to the same vertices, a partial function κ mapping each index in I to the index of the apex vertex to which the vertex indexed i corresponds, and a partial function δ mapping every index in I to the index of the color class among $X_1, \ldots, X_h$ in which the corresponding vertex belongs. Also, it encodes some graphs with I as vertex set. These are (1) the graph H^e that corresponds to the graph induced by the vertices $v_i, i \in I$ and (2) the collection $\mathcal{H}^P$ that has all graphs with vertex set I whose edge set corresponds to the existence of internally vertex-disjoint paths between the respective vertices. We set

$$\mathcal{G}_{\mathsf{pat}}^{(r,h,l)} = \{\mathsf{pattern}(G, X_1, \ldots, X_h, \mathbf{a}, v_1, \ldots, v_r) \mid (G, X_1, \ldots, X_h, \mathbf{a}, v_1, \ldots, v_r) \in \mathcal{B}^{(r,h,l)}\}.$$

Also, by the definition of a pattern, we observe the following.

Observation 3. *There is a function $f_1 : \mathbb{N}^3 \to \mathbb{N}$ such that, for every $r \in \mathbb{N}$, $|\mathcal{G}_{\mathsf{pat}}^{(r,h,l)}| \leq f_1(r, h, l)$.*

4.3 Expressing satisfiability of sentences using patterns

As mentioned in the previous subsection, patterns of boundaried colored graphs can be seen as encodings of the set of quantifier-free formulas of $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$ that the given boundaried colored graph satisfies. Following this line, in this subsection, we also encode quantifier-free formulas of $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$ in the setting of patterns. This will allow us to formulate model-checking questions in terms of pattern realization (see Observation 5 and Observation 6). Under this viewpoint, model-checking can be formulated in purely graph-theoretical terms (through the information encoded in patterns) of boundaried colored graphs, whose boundary is rescursively obtained, following the assignment of the given graph to a rooted tree.

We start with some additional definitions on formulas.

Atomic formulas and literals. Let $\tau = \{\mathsf{E}, \mathsf{Y}_1, \ldots, \mathsf{Y}_h\}$ be a colored-graph vocabulary and let a collection $\mathbf{c} = \{\mathsf{c}_1, \ldots, \mathsf{c}_l\}$ of constant symbols. An *atomic formula* is a formula of the form $\mathsf{x}_i = \mathsf{x}_j$, or $\mathsf{x}_i = \mathsf{c}_j$, or $\mathsf{x}_i \in \mathsf{Y}_j$, or $\mathsf{E}(\mathsf{x}_i, \mathsf{x}_j)$ or $\mathsf{dp}_k(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$ for some $k \in [r(r+1)/2]$, where $\mathsf{x}_i, \mathsf{x}_j, \mathsf{s}_1, \ldots, \mathsf{s}_k, \mathsf{t}_1, \ldots, \mathsf{t}_k$ are first-order variables. A *literal* is an atomic formula or the negation of an atomic formula.

Disjunctive normal form and full clauses. Let $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ be a quantifier-free $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$-formula. We say that $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ is in *disjunctive normal form* if there is some $k \in \mathbb{N}$ such that $\psi = c_1 \vee \cdots \vee c_k$, where for each $i \in [k]$, c_i is a conjunction of literals. We call each c_i a *clause* of ψ. We say ψ is in *full disjunctive normal form* if every ψ is in disjunctive normal form and, additionally, for every clause c of ψ it holds that

- for every distinct $i, j \in [r]$, either $\mathsf{x}_i = \mathsf{x}_j$ or its negation are literals of c,
- for every $i \in [r]$ and every $j \in [l]$, either $\mathsf{x}_i = \mathsf{c}_j$ or its negation are literals of c,
- for every $i \in [r]$ and every $j \in [h]$, either $\mathsf{x}_i \in \mathsf{Y}_j$ or its negation are literals of c,
- for every distinct $i, j \in [r]$, either $\mathsf{E}(\mathsf{x}_i, \mathsf{x}_j)$ or its negation are literals of c, and
- for every $\ell \in [r(r+1)/2]$ and every $i_1, i_2, \ldots, i_\ell, j_1, j_2, \ldots, j_\ell \in [r]$, either the atomic formula $\mathsf{dp}(\mathsf{x}_{i_1}, \mathsf{x}_{j_1}, \ldots, \mathsf{x}_{i_\ell}, \mathsf{x}_{j_\ell})$ or its negation are literals of c.

We say that a quantifier-free $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$-formula ψ' *extends* ψ if every clause of ψ is a subformula of a clause of ψ'. We now prove that, for every quantifier-free $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$-formula ψ, we can construct an equivalent quantifier-free formula ψ' of the same vocabulary that is in full disjunctive normal form and extends ψ.

Lemma 1. *For every quantifier-free $\mathsf{FOL}[(\tau+\mathsf{dp})\cup\mathbf{c}]$-formula ψ, there is a quantifier-free $\mathsf{FOL}[(\tau+\mathsf{dp}) \cup \mathbf{c}]$-formula ψ' that is in full disjunctive normal form, extends ψ, and is equivalent to ψ.*

Proof. Given a quantifier-free $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$-formula $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$, we construct the formula ψ' as follows. For every clause c of ψ, if there are $i, j \in [r]$ such that neither $\mathsf{x}_i = \mathsf{x}_j$ nor $\mathsf{x}_i \neq \mathsf{x}_j$ appear in c, then we replace c by

$$\big(c \wedge \mathsf{x}_i = \mathsf{x}_j\big) \vee \big(c \wedge \mathsf{x}_i \neq \mathsf{x}_j\big).$$

By recursively applying this procedure for $\mathsf{x}_i = \mathsf{c}_j$ for every $i \in [r]$ and every $j \in [l]$, for $\mathsf{x}_i \in \mathsf{Y}_j$ for every $i \in [r]$ and every $j \in [h]$, for $\mathsf{E}(\mathsf{x}_i, \mathsf{x}_j)$ for every distinct $i, j \in [r]$, and for dp_ℓ, for every $\ell \in [r(r+1)/2]$ and every $i_1, i_2, \ldots, i_\ell, j_1, j_2, \ldots, j_\ell \in [r]$, we obtain ψ'. $\square$

In the rest of this paper we assume that every quantifier-free formula $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ in $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$ is in disjunctive normal form.

Patterns of quantifier-free formulas. We now define a notion of pattern of quantifier-free formulas. Let $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ be a quantifier-free $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$-formula in full disjunctive normal form. We assume that ψ is satisfiable, i.e., $\mathrm{Mod}(\psi) \neq \emptyset$. For every clause c of ψ, we define the *pattern* of c, denoted by H_c, as the quintuple $(V_c, \kappa_c, \delta_c, H_c^e, \mathcal{H}_c^P)$, where

- V_c is a partition of $[r]$ into sets such that for every $A \in V_H$ and every $i, j \in A$ either $\mathsf{x}_i = \mathsf{x}_j$ appears as a literal of c or $i = j$,
- $\kappa_c = \{(i, j) \in [r] \times [l] \mid \text{the atomic formula } \mathsf{x}_i = \mathsf{c}_j \text{ appears as a literal of } c\}$,

- $\delta_c = \{(i,j) \in [r] \times [h] \mid \text{the atomic formula } \mathsf{x}_i \in \mathsf{Y}_j \text{ appears as a literal of } c\}$,
- $H^e_c = ([r], \{\{i,j\} \in [r] \times [r] \mid \text{the atomic formula } \mathsf{E}(\mathsf{x}_i, \mathsf{x}_j) \text{ appears as a literal of } c\})$, and
- $\mathcal{H}^P_c = \bigcup_{E \subseteq [r]\times[r]} \left\{ ([r], E) \;\middle|\; \begin{array}{l} \text{if } E = \{(i_1, j_1), \ldots, (i_\ell, j_\ell)\} \text{ then the atomic formula} \\ \mathsf{dp}(\mathsf{x}_{i_1}, \mathsf{x}_{j_1}, \ldots, \mathsf{x}_{i_\ell}, \mathsf{x}_{j_\ell}) \text{ appears as a literal of } c \end{array} \right\}$.

Notice that the fact that V_H is a partition of $[r]$ follows from the fact that ψ is satisfiable. Intuitively, satisfiability of ψ is asked so as to rule out the cases where, for example, all three atomic formulas $\mathsf{x}_1 = \mathsf{x}_2$, $\mathsf{x}_2 = \mathsf{x}_3$, and $\neg(\mathsf{x}_2 = \mathsf{x}_3)$ appear as literals in the same clause. Also, satisfiability of ψ and the fact that it is in full disjunctive normal form implies that for every $\{i,j\} \in I \times I \setminus E(H^e_c)$, $\neg\mathsf{E}(\mathsf{x}_i, \mathsf{x}_j)$ appears as a literal in c. If ψ is not satisfiable, then we set H_c to be $\emptyset$.

Let ψ be a quantifier-free $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$-formula. We define

$$\mathsf{ext}(\psi) = \{c \mid \ c \text{ is a clause of a formula } \psi' \text{ in full disjunctive normal form that extends } \psi\}.$$

Following Lemma 1, we get the following observation.

Observation 4. *For every quantifier-free* $\mathsf{FOL}[(\tau+\mathsf{dp})\cup\mathbf{c}]$*-formula* ψ*, the set* $\mathsf{ext}(\psi)$ *is non-empty.*

We set $\mathcal{H}_\psi$ to be the collection of patterns of all $c \in \mathsf{ext}(\psi)$, i.e.,

$$\mathcal{H}_\psi = \{H_c \mid c \in \mathsf{ext}(\psi)\},$$

and we call it the *set of patterns of* ψ. Having defined the set $\mathcal{H}_\psi$ of patterns of ψ, we now define when a boundaried colored graph *realizes* an element of $\mathcal{H}_\psi$.

Realizing a pattern. Let $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ be a quantifier-free $\mathsf{FOL}[(\tau + \mathsf{dp}) \cup \mathbf{c}]$-formula, let $c \in \mathsf{ext}(\psi)$, and let H_c be the pattern of c. Given a colored rooted graph $\mathfrak{G}$ and vertices $v_1, \ldots, v_r \in V(\mathfrak{G})$, we say that $(\mathfrak{G}, v_1, \ldots, v_r)$ *realizes* H_c if $\mathsf{pattern}(\mathfrak{G}, v_1, \ldots, v_r) = H_c$. Note that, equivalently, $(\mathfrak{G}, v_1, \ldots, v_r)$ realizes H_c if $(\mathfrak{G}, v_1, \ldots, v_r) \models c(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ (where x_i is interpreted as v_i, for every $i \in [r]$). Keep in mind that every $(\mathfrak{G}, v_1, \ldots, v_r)$ realizes at most one $H_c \in \mathcal{H}_\psi$.

Due to Observation 4 and the definition of realization of an element of $\mathcal{H}_\psi$, we obtain the following result.

Observation 5. *Let* $r \in \mathbb{N}$ *and let* $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ *be a quantifier-free* $\mathsf{FOL}[(\tau+\mathsf{dp})\cup\mathbf{c}]$*-formula. Then* $(\mathfrak{G}, v_1, \ldots, v_r) \models \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ *if and only if* $(\mathfrak{G}, v_1, \ldots, v_r)$ *realizes an element of* $\mathcal{H}_\psi$.

To conclude this subsection, we explain how to revisit the approach presented in Subsection 4.1, and, in particular, how to restate Observation 2 using patterns.

Pattern-coloring. Let $r, h, l \in \mathbb{N}$. Let $\mathfrak{G}$ be an h-colored l-rooted graph and let λ be an assignment of G to a rooted tree (T, t_0) of height r. We define the function $\mathsf{pc}_\lambda : L(T) \to \mathcal{G}^{(r,h,l)}_{\mathsf{pat}}$ such that for every $t \in L(T)$, if $(t_0, t_1, \ldots, t_r = t) \in \mathsf{Paths}(T)$, then

$$\mathsf{pc}_\lambda(t) = \mathsf{pattern}(\mathfrak{G}, \lambda(t_1), \ldots, \lambda(t)).$$

We call pc_λ the *pattern-coloring of* $L(T)$ *with respect to* λ.

Using the definition of the pattern-coloring and Observation 5, we can now restate Observation 2 as follows.

Observation 6. *Let $r, l \in \mathbb{N}$. Let τ be a colored rooted graph vocabulary. For every τ-structure $\mathfrak{G}$ and every sentence $\varphi = Q_1\mathsf{x}_1 \ \ldots Q_r\mathsf{x}_r \ \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ where $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r) \in \mathsf{FOL}[\tau + \mathsf{dp}]$, we have that $\mathfrak{G} \models \varphi$ if and only if there is*

- *an assignment λ of G to a rooted tree (T, t_0) of height r and*
- *a φ-spanning subtree (T', t_0) of (T, t_0)*

such that for every $t \in L(T')$, $\mathsf{pc}_\lambda(t) \in \mathcal{H}_\psi$.

4.4 Graphs with the same patterns satisfy the same sentences

In this subsection, we aim to prove that if two colored graphs give the same signatures, then these two colored graphs satisfy the same (annotated) sentences.

Lemma 2. *Let τ be a colored rooted graph vocabulary and let $\mathfrak{G}, \mathfrak{G}'$ be two τ-structures. For every $r \in \mathbb{N}$, if $\mathsf{sig}^r(\mathfrak{G}) = \mathsf{sig}^r(\mathfrak{G}')$, then for every sentence $\varphi \in \mathsf{FOL}[\tau + \mathsf{dp}]$ of quantifier rank at most r, it holds that*

$$\mathfrak{G} \models \varphi \iff \mathfrak{G}' \models \varphi.$$

Proof. Let $\varphi = Q_1\mathsf{x}_1 \ \ldots Q_r\mathsf{x}_r \ \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$, where $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ is a quantifier-free formula in $\mathsf{FOL}[\tau + \mathsf{dp}]$. Let λ be an assignment of $\mathfrak{G}$ to some rooted tree (T, t_0) of height r and let λ' be an assignment of $\mathfrak{G}'$ to some rooted tree $(\tilde{T}, \tilde{t}_0)$ of height r. Due to Observation 6, to prove that $\mathfrak{G} \models \varphi \iff \mathfrak{G}' \models \varphi$, it suffices to show that

$$\begin{gathered}\text{there is a } \varphi\text{-spanning subtree } (T', t_0) \text{ of } (T, t_0) \text{ certifying that } \mathfrak{G} \models \varphi \\ \iff \\ \text{there is a } \varphi\text{-spanning subtree } (\tilde{T}', \tilde{t}_0) \text{ of } (\tilde{T}, \tilde{t}_0) \text{ certifying that } \mathfrak{G}' \models \varphi.\end{gathered} \tag{1}$$

For every $i \in [0, r]$, we denote by $\varphi^i(\mathsf{x}_1, \ldots, \mathsf{x}_i)$ the formula $Q_{i+1}\mathsf{x}_{i+1} \ldots Q_r\mathsf{x}_r\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$. We first prove an analogue of (1) for the quantifier-free formula $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$. Recall that since $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ is quantifier-free, given a $t \in L(T)$, T_t has a unique $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$-spanning subtree, that is T_t itself. Nevertheless, we formulate the next statement in terms of $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$-spanning subtrees in order to use it as the base case of a recursive argument built in the course of the proof of (1).

Claim 1. *For every $(t_0, t_1, \ldots, t_r) \in \mathit{Paths}(T)$ and every $(\tilde{t}_0, \tilde{t}_1, \ldots, \tilde{t}_r) \in \mathit{Paths}(\tilde{T})$, such that for every $i \in [r]$, $\mathsf{sig}^{r-i}(\mathfrak{G}, \lambda(t_1), \ldots, \lambda(t_i)) = \mathsf{sig}^{r-i}(\mathfrak{G}', \lambda'(\tilde{t}_1), \ldots, \lambda'(\tilde{t}_i))$, it holds that*

$$\begin{gathered}\textit{there is a } \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)\textit{-spanning subtree } T'_{t_r} \textit{ of } T_{t_r} \textit{ certifying that} \\ (\mathfrak{G}, \lambda(t_1), \ldots, \lambda(t_r)) \models \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r) \\ \iff \\ \textit{there is a } \varphi^r(\mathsf{x}_1, \ldots, \mathsf{x}_r)\textit{-spanning subtree } \tilde{T}'_{\tilde{t}_r} \textit{ of } \tilde{T}_{\tilde{t}_r} \textit{ certifying that} \\ \big(\mathfrak{G}', \lambda'(\tilde{t}_1), \ldots, \lambda'(\tilde{t}_r)\big) \models \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r).\end{gathered}$$

Proof of Claim 1: Since $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ is a quantifier-free formula, T_{t_r} has a unique $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$-spanning subtree, that is T_{t_r}, and the same holds for $\tilde{T}_{\tilde{t}_r}$. By assumption, $\mathsf{sig}^0(\mathfrak{G}, \lambda(t_1), \ldots, \lambda(t_r)) = \mathsf{sig}^0(\mathfrak{G}', \lambda'(\tilde{t}_1), \ldots, \lambda'(\tilde{t}_r))$. Therefore, $\mathsf{pc}_\lambda(t_r) = \mathsf{pc}_{\lambda'}(\tilde{t}_r)$, which in turn implies that $\mathsf{pc}_\lambda(t_r) \in \mathcal{H}_\psi \iff \mathsf{pc}_{\lambda'}(\tilde{t}_r) \in \mathcal{H}_\psi$. ⋄

We now prove the following. The case where $i = 0$ proves (1).

Claim 2. *For every $i \in [0, r-1]$, for every $(t_0, t_1, \dots, t_i) \in$ Paths(T) and every $(\tilde{t}_0, \tilde{t}_1, \dots, \tilde{t}_i) \in$ Paths$(\tilde{T})$, such that for every $j \in [i]$, $\mathsf{sig}^{r-j}(\mathfrak{G}, \lambda(t_1), \dots, \lambda(t_j)) = \mathsf{sig}^{r-j}(\mathfrak{G}', \lambda'(\tilde{t}_1), \dots, \lambda'(\tilde{t}_j))$, it holds that*

$$\text{there is a } \varphi^i(\mathsf{x}_1, \dots, \mathsf{x}_i)\text{-spanning subtree } T'_{t_i} \text{ of } T_{t_i} \text{ certifying that}$$
$$(\mathfrak{G}, \lambda(t_1), \dots, \lambda(t_i)) \models \varphi^i(\mathsf{x}_1, \dots, \mathsf{x}_i)$$
$$\Longleftrightarrow$$
$$\text{there is a } \varphi^i(\mathsf{x}_1, \dots, \mathsf{x}_i)\text{-spanning subtree } \tilde{T}'_{\tilde{t}_i} \text{ of } \tilde{T}_{\tilde{t}_i} \text{ certifying that}$$
$$\big(\mathfrak{G}', \lambda'(\tilde{t}_1), \dots, \lambda'(\tilde{t}_i)\big) \models \varphi^i(\mathsf{x}_1, \dots, \mathsf{x}_i).$$

Proof of Claim 2: In Claim 1, we already proved the statement for $i = r$. Let $i \in [0, r-1]$. We assume that the statement holds for $i+1$. We will prove that it holds for i. We distinguish two cases, depending on whether $Q_{i+1} = \exists$ or $Q_{i+1} = \forall$.

Case 1: $Q_{i+1} = \exists$.

($\Leftarrow$) Suppose that there is a $\varphi^i(\mathsf{x}_1, \dots, \mathsf{x}_i)$-spanning subtree $\tilde{T}'_{\tilde{t}_i}$ of $\tilde{T}_{\tilde{t}_i}$ certifying that

$$\big(\mathfrak{G}', \lambda'(\tilde{t}_1), \dots, \lambda'(\tilde{t}_i)\big) \models \varphi^i(\mathsf{x}_1, \dots, \mathsf{x}_i)$$

(in the case where $i = 0$, we suppose that $\tilde{T}'_{\tilde{t}_i} = \tilde{T}'$ certifies that $\mathfrak{G}' \models \varphi$).

Since $Q_{i+1} = \exists$, there exists a node $z \in \mathsf{children}_{\tilde{T}}(\tilde{t}_i)$ that belongs to $V(\tilde{T}'_{\tilde{t}_i})$. Also, observe that $\tilde{T}'_z$ is a $\varphi^{i+1}(\mathsf{x}_1, \dots, \mathsf{x}_{i+1})$-spanning subtree of $\tilde{T}_z$ certifying that

$$\big(\mathfrak{G}', \lambda'(\tilde{t}_1), \dots, \lambda'(\tilde{t}_i), \lambda'(z)\big) \models \varphi^{i+1}(\mathsf{x}_1, \dots, \mathsf{x}_{i+1}).$$

Notice that, since $\mathsf{sig}^{r-i}(\mathfrak{G}, \lambda(t_1), \dots, \lambda(t_i)) = \mathsf{sig}^{r-i}(\mathfrak{G}', \lambda'(\tilde{t}_1), \dots, \lambda'(\tilde{t}_i))$, there exists a $t_{i+1} \in \mathsf{children}_T(t_i)$ such that $\mathsf{sig}^{r-i-1}(\mathfrak{G}, \lambda(t_1), \dots, \lambda(t_i), \lambda(t_{i+1})) = \mathsf{sig}^{r-i-1}(\mathfrak{G}', \lambda'(\tilde{t}_1), \dots, \lambda'(\tilde{t}_i), \lambda'(z))$. Following our recursive assumption, there is a $\varphi^{i+1}(\mathsf{x}_1, \dots, \mathsf{x}_{i+1})$-spanning subtree $T'_{t_{i+1}}$ of $T_{t_{i+1}}$ certifying that

$$\big(\mathfrak{G}, \lambda(t_1), \dots, \lambda(t_i), \lambda(t_{i+1})\big) \models \varphi^{i+1}(\mathsf{x}_1, \dots, \mathsf{x}_{i+1}).$$

Then, the graph $T'_{t_i} := T[\{t_i\} \cup V(T'_{t_{i+1}})]$ is a $\varphi^i(\mathsf{x}_1, \dots, \mathsf{x}_i)$-spanning subtree of T_{t_i} certifying that

$$\big(\mathfrak{G}, \lambda(t_1), \dots, \lambda(t_i)\big) \models \varphi^i(\mathsf{x}_1, \dots, \mathsf{x}_i).$$

In the case where $i = 0$, we have that $T'_{t_i} = T'$ certifies that $\mathfrak{G} \models \varphi$.

($\Rightarrow$) Suppose that there is a $\varphi^i(\mathsf{x}_1, \dots, \mathsf{x}_i)$-spanning subtree T'_{t_i} of T_{t_i} certifying that

$$\big(\mathfrak{G}, \lambda(t_1), \dots, \lambda(t_i)\big) \models \varphi^i(\mathsf{x}_1, \dots, \mathsf{x}_i)$$

(in the case where $i = 0$, we suppose that $T'_{t_i} = T'$ certifies that $\mathfrak{G} \models \varphi$).

Since $Q_{i+1} = \exists$, there exists a node $t_{i+1} \in \mathsf{children}_T(t_i)$ that belongs to $V(T'_{t_i})$. Also, observe that $T'_{t_{i+1}}$ is a $\varphi^{i+1}(\mathsf{x}_1, \dots, \mathsf{x}_{i+1})$-spanning subtree of $T_{t_{i+1}}$ certifying that

$$\big(\mathfrak{G}, \lambda(t_1), \dots, \lambda(t_i), \lambda(t_{i+1})\big) \models \varphi^{i+1}(\mathsf{x}_1, \dots, \mathsf{x}_{i+1}).$$

Since $\mathsf{sig}^{r-i}(\mathfrak{G}, \lambda(t_1), \dots, \lambda(t_i)) = \mathsf{sig}^{r-i}(\mathfrak{G}', \lambda'(\tilde{t}_1), \dots, \lambda'(\tilde{t}_i))$, there is a node $\tilde{t}_{i+1}$ in $\mathsf{children}_{\tilde{T}}(\tilde{t}_i)$ such that

$$\mathsf{sig}^{r-i-1}(\mathfrak{G}, \lambda(t_1), \dots, \lambda(t_i), \lambda(t_{i+1})) = \mathsf{sig}^{r-i-1}(\mathfrak{G}', \lambda'(\tilde{t}_1), \dots, \lambda'(\tilde{t}_i), \lambda'(\tilde{t}_{i+1})).$$

Following our recursive assumption, there is a $\varphi^{i+1}(\mathsf{x}_1,\ldots,\mathsf{x}_{i+1})$-spanning subtree $\tilde{T}'_{\tilde{t}_{i+1}}$ of $\tilde{T}_{\tilde{t}_{i+1}}$ certifying that

$$\big(\mathfrak{G}',\lambda'(\tilde{t}_1),\ldots,\lambda'(\tilde{t}_i),\lambda'(\tilde{t}_{i+1})\big)\models\varphi^{i+1}(\mathsf{x}_1,\ldots,\mathsf{x}_{i+1}).$$

Now, observe that the graph $\tilde{T}'_{\tilde{t}_i}=\tilde{T}[\{\tilde{t}_i\}\cup V(\tilde{T}'_{\tilde{t}_{i+1}})]$ is a $\varphi^{i}(\mathsf{x}_1,\ldots,\mathsf{x}_{i})$-spanning subtree of $\tilde{T}_{\tilde{t}_i}$ certifying that

$$\big(\mathfrak{G}',\lambda'(\tilde{t}_1),\ldots,\lambda'(\tilde{t}_i)\big)\models\varphi^{i}(\mathsf{x}_1,\ldots,\mathsf{x}_{i}).$$

In the case where $i=0$, we have that $\tilde{T}'_{\tilde{t}_i}=\tilde{T}'$ certifies that $\mathfrak{G}\models\varphi$.

$\boxed{\textbf{Case 2}}$: $Q_{i+1}=\forall$.

($\Leftarrow$) Suppose that there is a $\varphi^{i}(\mathsf{x}_1,\ldots,\mathsf{x}_{i})$-spanning subtree $\tilde{T}'_{\tilde{t}_i}$ of $\tilde{T}_{\tilde{t}_i}$ certifying that

$$\big(\mathfrak{G}',\lambda'(\tilde{t}_1),\ldots,\lambda'(\tilde{t}_i)\big)\models\varphi^{i}(\mathsf{x}_1,\ldots,\mathsf{x}_{i})$$

(in the case where $i=0$, we suppose that $\tilde{T}'_{\tilde{t}_i}=\tilde{T}'$ certifies that $\mathfrak{G}'\models\varphi$).

Since $Q_{i+1}=\forall$, every $z\in\mathsf{children}_{\tilde{T}}(\tilde{t}_i)$ belongs to $V(\tilde{T}'_{\tilde{t}_i})$. Now observe that, for every $z\in\mathsf{children}_{\tilde{T}}(\tilde{t}_i)$, $\tilde{T}'_z$ is a $\varphi^{i+1}(\mathsf{x}_1,\ldots,\mathsf{x}_{i+1})$-spanning subtree of $\tilde{T}_z$ certifying that

$$\big(\mathfrak{G}',\lambda'(\tilde{t}_1),\ldots,\lambda'(\tilde{t}_i),\lambda'(z)\big)\models\varphi^{i+1}(\mathsf{x}_1,\ldots,\mathsf{x}_{i+1}).$$

Also, since $\mathsf{sig}^{r-d}(\mathfrak{G},\lambda(t_1),\ldots,\lambda(t_i))=\mathsf{sig}^{r-d}(\mathfrak{G}',\lambda'(\tilde{t}_1),\ldots,\lambda'(\tilde{t}_i))$, for every $z\in\mathsf{children}_{\tilde{T}}(\tilde{t}_i)$ the set

$$\{t_{i+1}\in\mathsf{children}_T(t_i)\mid\mathsf{sig}^{r-i-1}(\mathfrak{G},\lambda(t_1),\ldots,\lambda(t_i),\lambda(t_{i+1}))=\mathsf{sig}^{r-i-1}(\mathfrak{G}',\lambda'(\tilde{t}_1),\ldots,\lambda'(\tilde{t}_i),\lambda'(z))\}$$

is non-empty and, also, $\bigcup_{z\in\mathsf{children}_{\tilde{T}}(\tilde{t}_i)}\{t_{i+1}\in\mathsf{children}_T(t_i)\mid\mathsf{sig}^{r-i-1}(\mathfrak{G},\lambda(t_1),\ldots,\lambda(t_i),\lambda(t_{i+1}))=\mathsf{sig}^{r-i-1}(\mathfrak{G}',\lambda'(\tilde{t}_1),\ldots,\lambda'(\tilde{t}_i),\lambda'(z))\}=\mathsf{children}_T(t_i)$. Following our recursive assumption, for every $t_{i+1}\in\mathsf{children}_T(t_i)$, there is a $\varphi^{i+1}(\mathsf{x}_1,\ldots,\mathsf{x}_{i+1})$-spanning subtree $T'_{t_{i+1}}$ of $T_{t_{i+1}}$ certifying that

$$\big(\mathfrak{G},\lambda(t_1),\ldots,\lambda(t_i),\lambda(t_{i+1})\big)\models\varphi^{i+1}(\mathsf{x}_1,\ldots,\mathsf{x}_{i+1}).$$

Then, the graph

$$T[\{t_i\}\cup\bigcup_{t_{i+1}\in\mathsf{children}_T(t_i)}V(T'_{t_{i+1}})]$$

is a $\varphi^{i}(\mathsf{x}_1,\ldots,\mathsf{x}_{i})$-spanning subtree of T_{t_i} certifying that

$$\big(\mathfrak{G},\lambda(t_1),\ldots,\lambda(t_i)\big)\models\varphi^{i}(\mathsf{x}_1,\ldots,\mathsf{x}_{i})$$

In the case where $i=0$, we have that $T'_{t_i}=T'$ certifies that $\mathfrak{G}\models\varphi$.

($\Rightarrow$) Suppose that there is a $\varphi^{i}(\mathsf{x}_1,\ldots,\mathsf{x}_{i})$-spanning subtree T'_{t_i} of T_{t_i} certifying that

$$\big(\mathfrak{G},\lambda(t_1),\ldots,\lambda(t_i)\big)\models\varphi^{i}(\mathsf{x}_1,\ldots,\mathsf{x}_{i})$$

(in the case where $i=0$, we suppose that $T'_{t_i}=T'$ certifies that $\mathfrak{G}\models\varphi$).

Since $Q_{i+1}=\forall$, every $t_{i+1}\in\mathsf{children}_T(t_i)$ belongs to $V(T'_{t_i})$. Now observe that, for every $t_{i+1}\in\mathsf{children}_T(t_i)$, $T'_{t_{i+1}}$ is a $\varphi^{i+1}(\mathsf{x}_1,\ldots,\mathsf{x}_{i+1})$-spanning subtree of $T_{t_{i+1}}$ certifying that

$$\big(\mathfrak{G},\lambda(t_1),\ldots,\lambda(t_i),\lambda(t_{i+1})\big)\models\varphi^{i+1}(\mathsf{x}_1,\ldots,\mathsf{x}_{i+1}).$$

Since $\mathsf{sig}^{r-d}(\mathfrak{G}, \lambda(t_1), \ldots, \lambda(t_i)) = \mathsf{sig}^{r-d}(\mathfrak{G}', \lambda'(\tilde{t}_1), \ldots, \lambda'(\tilde{t}_i))$, for every $t_{i+1} \in \mathsf{children}_T(t_i)$ there exists a node $z \in \mathsf{children}_{\tilde{T}}(\tilde{t}_i)$ such that

$$\mathsf{sig}^{r-i-1}(\mathfrak{G}, \lambda(t_1), \ldots, \lambda(t_i), \lambda(t_{i+1})) = \mathsf{sig}^{r-i-1}(\mathfrak{G}', \lambda'(\tilde{t}_1), \ldots, \lambda'(\tilde{t}_i), \lambda'(z)).$$

Following our recursive assumption, for every $z \in \mathsf{children}_T(t_i)$ there is a $\varphi^{i+1}(\mathsf{x}_1, \ldots, \mathsf{x}_{i+1})$-spanning subtree $\tilde{T}'_{\tilde{t}_{i+1}}$ of $\tilde{T}_{\tilde{t}_{i+1}}$ certifying that

$$\big(\mathfrak{G}, \lambda(t_1), \ldots, \lambda(t_i), \lambda(t_{i+1})\big) \models \varphi^{i+1}(\mathsf{x}_1, \ldots, \mathsf{x}_{i+1}).$$

Also, note that for every $\tilde{t}_{i+1} \in \mathsf{children}_{\tilde{T}}(\tilde{t}_i)$, there exists a node $t_{i+1} \in \mathsf{children}_T(t_i)$ such that $\mathsf{sig}^{r-i-1}(\mathfrak{G}, \lambda(t_1), \ldots, \lambda(t_i), \lambda(z)) = \mathsf{sig}^{r-i-1}(\mathfrak{G}', \lambda'(\tilde{t}_1), \ldots, \lambda'(\tilde{t}_i), \lambda'(\tilde{t}_{i+1}))$. Restating what we mentioned above for the existence of a $\varphi^{i+1}(\mathsf{x}_1, \ldots, \mathsf{x}_{i+1})$-spanning subtree of $T_{t_{i+1}}$ for each $t_{i+1} \in \mathsf{children}_T(t_i)$, we get that for every $z \in \mathsf{children}_{\tilde{T}}(\tilde{t}_i)$, there is a $\varphi^{i+1}(\mathsf{x}_1, \ldots, \mathsf{x}_{i+1})$-spanning subtree $\tilde{T}'_z$ of $\tilde{T}_z$ certifying that

$$\big(\mathfrak{G}, \lambda'(\tilde{t}_1), \ldots, \lambda'(\tilde{t}_i), \lambda'(z)\big) \models \varphi^{i+1}(\mathsf{x}_1, \ldots, \mathsf{x}_{i+1}).$$

Now, observe that the graph

$$\tilde{T}[\{\tilde{t}_i\} \cup \bigcup_{z \in \mathsf{children}_{\tilde{T}}(\tilde{t}_i)} V(\tilde{T}'_z)]$$

is a $\varphi^i(\mathsf{x}_1, \ldots, \mathsf{x}_i)$-spanning subtree of $\tilde{T}_{\tilde{t}_i}$ certifying that

$$\big(\mathfrak{G}', \lambda'(\tilde{t}_1), \ldots, \lambda'(\tilde{t}_i)\big) \models \varphi^i(\mathsf{x}_1, \ldots, \mathsf{x}_i).$$

In the case where $i = 0$, we have that $\tilde{T}'_{\tilde{t}_i} = \tilde{T}'$ certifies that $\mathfrak{G}' \models \varphi$. This concludes the proof of the Claim. $\diamond$

Following Claim 2 for $i = 0$, we deduce (1), which completes the proof of the lemma. $\square$

A comment on the use of Lemma 2. Before concluding this section, we wish to stress the following. Suppose that we are given an (annotated) colored graph and one can find an (annotated) colored graph that has the same "meta-collection" of patterns (in the sense of equivalence of leaf-labeled trees, as in the statement of Lemma 2). Lemma 2 indicates that model-checking for the original annotated colored graph can be reduced to model-checking for the second annotated colored graph and this is safe for *every* formula in $\mathsf{FOL}[\tau + \mathsf{dp}]$. Therefore, if we were able to find a way to compute such an "equivalent" graph and if its size (or, the size of its annotated set) was smaller than the original one's, we could reduce the problem of model-checking to smaller instances and therefore report some progress. However, this is not straightforward and in the next three sections, i.e., Section 5, Section 6, and Section 7, we explain how to deal with this situation. From a high-level point of view, our approach considers expressing partial patterns (and consequently, partial satisfaction of formulas in $\mathsf{FOL}[\tau + \mathsf{dp}]$) inside a bounded treewidth part of the given graph. Then, using Courcelle's Theorem, we will be able to compute *representatives* of the vertices inside this bounded treewidth part with respect to equivalence of (partial) patterns. Using the fact that we know which vertices in this part "represent" the variety of patterns, we find some "irrelevant" vertices to discard from the annotation, without changing the signature.

5 Routing linkages through railed annuli

In Section 4, we presented how to encode questions expressed in $\mathsf{FOL}[\tau+\mathsf{dp}]$ in purely graph-theoretical terms, using patterns. Recall that a pattern of a boundaried colored graph encodes, by the collection $\mathcal{H}^P$, what disjoint paths can be routed through the boundary vertices. This is the only "non-local" information encoded in the pattern, as all other information can be determined by inspecting only the boundaried vertices and the adjacencies between them. Aiming to define a notion of "partial" pattern, we have to deal with the possible ways that boundary vertices can be connected through disjoint paths and this can be seen as a question about the variety of linkages that can be routed through the boundaried vertices. A crucial tool for handling linkages is the Linkage Combing Lemma (Proposition 2) proved in [103] (see also [101]). This result is applied in the presence of a *partially annulus-embedded graph* and an *annulus-embedded railed annulus*, notions that are defined in Subsection 5.1. The definition of linkages and the Linkage Combing Lemma of [103] are presented in Subsection 5.2. Finally, in Subsection 5.3, we define linkages of boundaried graphs and we describe how to "encode" models of boundaried graphs using patterns.

5.1 Graphs partially embedded on an annulus and railed annuli

We say that a pair $(L, R) \in 2^{V(G)} \times 2^{V(G)}$ is a *separation* of G if $L \cup R = V(G)$ and there is no edge in G between a vertex in $L \setminus R$ and a vertex in $R \setminus L$. We say that two separations (X_1, Y_1) and (X_2, Y_2) of a graph G are *laminar* if $Y_1 \subseteq Y_2$ and $X_2 \subseteq X_1$.

Disks and annuli. A *cycle* is a set homeomorphic to the set $\{(x, y) \in \mathbb{R}^2 \mid x^2+y^2 = 1\}$. We define a *closed disk* (resp. *open disk*) to be a set homeomorphic to the set $\{(x, y) \in \mathbb{R}^2 \mid x^2+y^2 \leq 1\}$ (resp. $\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 < 1\}$) and a *closed annulus* (resp. *open annulus*) to be a set homeomorphic to the set $\{(x, y) \in \mathbb{R}^2 \mid 1 \leq x^2 + y^2 \leq 2\}$ (resp. $\{(x, y) \in \mathbb{R}^2 \mid 1 < x^2 + y^2 < 2\}$). Given a closed disk or a closed annulus X, we use $\mathsf{bd}(X)$ to denote the boundary of X (i.e., the set of points of X for which every neighborhood around them contains some point not in X). Notice that if X is a closed disk then $\mathsf{bd}(X)$ is a cycle, while if X is a closed annulus then $\mathsf{bd}(X) = C_1 \cup C_2$ where C_1, C_2 are the two unique connected components of $\mathsf{bd}(X)$ and C_1, C_2 are two disjoint cycles. We call C_1 and C_2 *boundaries* of X. We call C_1 the *left boundary* of X and C_2 the *right boundary* of X. Also given a closed disk (resp. closed annulus) X, we use $\mathsf{int}(X)$ to denote the open disk (resp. open annulus) $X \setminus \mathsf{bd}(X)$. When we embed a graph G in the plane, in a closed disk, or in a closed annulus, we treat G as a set of points. This permits us to make set operations between graphs and sets of points.

Partially annulus-embedded graphs. Let Δ be a closed annulus. We say that a graph G is *partially Δ-embedded*, if there is some subgraph K of G that is embedded in Δ such that $\mathsf{bd}(\Delta)$ is the disjoint union of two cycles of K and there are two laminar separations (X_1, Y_1) and (X_2, Y_2) of G such that $X_1 \cap Y_2 = V(G) \cap \Delta$. We also call the graph K *compass* of the partially Δ-embedded graph G and we always assume that we accompany a partially Δ-embedded graph G together with an embedding of its compass in Δ that is the set $G \cap \Delta$. See Figure 5 for an illustration of a partially annulus-embedded graph.

Let Δ be a closed annulus with left boundary B_1 and right boundary B_2. Also, let G be a partially Δ-embedded graph. We denote by $\mathsf{Left}_\Delta(G)$ the connected component of $G \setminus \mathsf{int}(\Delta)$ that contains B_1 and by $\mathsf{Right}_\Delta(G)$ the graph $G \setminus ((V(G) \cap \Delta) \cup V(\mathsf{Left}_\Delta(G)))$.

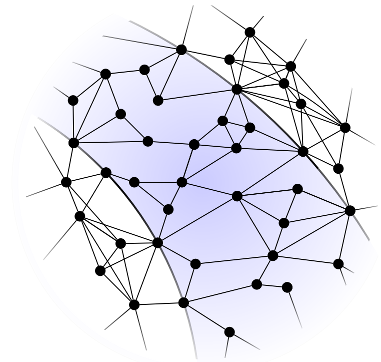

Figure 5: An illustration of a partially annulus-embedded graph.

Parallel cycles. Let Δ be a closed annulus with left boundary B_1 and right boundary B_2 and let G be a partially Δ-embedded graph. Also, let $\mathcal{C} = [C_1, \ldots, C_r]$, $r \geq 2$ be a collection of vertex disjoint cycles of G that are embedded in Δ. We say that $\mathcal{C}$ is a Δ-*parallel sequence of cycles* of G if $C_1 = B_1$, $C_r = B_2$ and, for every $i \in [2, r]$, C_1 and C_i are the boundaries of a closed annulus, denoted by Δ_i, that is a subset of Δ such that $\Delta_2 \subseteq \cdots \subseteq \Delta_r = \Delta$. We call C_1 the *leftmost* and C_r the *rightmost* cycle of $\mathcal{C}$. From now on, each Δ-parallel sequence $\mathcal{C}$ of cycles will be accompanied with the sequence $[\Delta_2, \ldots, \Delta_r]$ of the corresponding closed annuli. Given $i, j \in [2, r]$, where $i \leq j$, we call the set $\Delta_i \setminus \mathsf{int}(\Delta_j)$ (i, j)-*annulus* of $\mathcal{C}$ and we denote it by $\mathsf{ann}(\mathcal{C}, i, j)$. Also, for every $i \in [2, r]$, we set Δ_i to be the $(1, i)$-*annulus* of $\mathcal{C}$ and we also denote it by $\mathsf{ann}(\mathcal{C}, 1, i)$.

Railed annuli. Let G be a graph. Also, let $r \in \mathbb{N}_{\geq 3}$ and $q \in \mathbb{N}_{\geq 3}$ and assume that r is an odd number. An (r, q)-*railed annulus* of G is a pair $\mathcal{A} = (\mathcal{C}, \mathcal{P})$ where $\mathcal{C} = [C_1, \ldots, C_r]$ is a sequence of cycles of G and $\mathcal{P} = [P_1, \ldots, P_q]$ is a collection of pairwise vertex-disjoint paths in G such that

- For every $j \in [q]$, the endpoints of P_j are vertices of C_r and C_1 and
- for every $(i, j) \in [r] \times [q]$, $C_i \cap P_j$ is a non-empty path, that we denote $P_{i,j}$.

See Figure 6 for an example. We refer to the paths of $\mathcal{P}$ as the *rails* of $\mathcal{A}$ and to the cycles of $\mathcal{C}$ as the *cycles* of $\mathcal{A}$. We use $V(\mathcal{A})$ to denote the vertex set $\bigcup_{i\in[r]} V(C_i) \cup \bigcup_{j\in[q]} V(P_j)$ and $E(\mathcal{A})$ to denote the edge set $\bigcup_{i\in[r]} E(C_i) \cup \bigcup_{j\in[q]} E(P_j)$. We can see each path P_j in $\mathcal{P}$ as being oriented towards the "inner" part of $calA$, i.e., starting from a vertex of C_p and finishing to a vertex of $P_{1,j}$. For every $(i, j) \in [r] \times [q]$, we denote by $s_{i,j}$ (resp. $t_{i,j}$) the first (resp. last) vertex of $P_{i,j}$ when traversing P_j according to this orientation. If $(i, i') \in [r]^2$ with $i < i'$ then we define $\mathcal{A}_{i,i'}$ to be the railed annulus $([C_i, \cdots, C_{i'}], P'_1, \ldots, P'_q)$, where for every $j \in [q]$, P'_j is the subpath of P_j between $s_{i,j}$ and $t_{i',j}$.

Annulus-embedded railed annuli. Let Δ be a closed annulus and let G be a partially Δ-embedded graph. Also, let $r \in \mathbb{N}_{\geq 3}$ and $q \in \mathbb{N}_{\geq 3}$ and assume that r is an odd number. A (r, q)-railed annulus $\mathcal{A} = (\mathcal{C}, \mathcal{P})$ of G is called Δ-*embedded* if $\mathcal{C} = [C_1, \ldots, C_r]$ is a Δ-parallel sequence of cycles of G. We use $\mathsf{ann}(\mathcal{A})$ to denote $\mathsf{ann}(\mathcal{C}, 1, r)$.

The following proposition [14, Proposition 5.1] states that large railed annuli can be found inside a slightly larger wall and will be used in the next section. For the definition of a wall see Section B.1.

Proposition 1. *If $x, z \geq 3$ are odd integers, $y \geq 1$, and W is an* $\mathsf{odd}(2x + \max\{z, \frac{y}{4} - 1\})$*-wall, then*

- *there is a collection $\mathcal{P}$ of y paths in W such that if $\mathcal{C}$ is the collection of the first x layers of W, then $(\mathcal{C}, \mathcal{P})$ is an (x, y)-railed annulus of W where the first cycle of $\mathcal{C}$ is the perimeter of W, and*
- *the open disk defined by the x-th cycle of $\mathcal{C}$ contains the vertices of the compass of the central z-subwall of W.*

5.2 Combing linkages

In this subsection we define linkages and we present the Linkage Combing Lemma from [103] (Proposition 2) – see also [101].

Linkages. A *linkage* in a graph G is a subgraph L of G whose connected components are non-trivial paths. The *paths* of a linkage are its connected components and we denote them by $\mathcal{P}(L)$. We call $|\mathcal{P}|$ the *size* of $\mathcal{P}(L)$. The *terminals* of a linkage L, denoted by $T(L)$, are the endpoints of the paths of L, and the *pattern* of L is the set $\{\{s,t\} \mid \mathcal{P}(L)$ contains some (s,t)-path$\}$. Two linkages L_1, L_2 of G are *equivalent* if they have the same pattern and we denote this fact by $L_1 \equiv L_2$. Let Δ be a closed annulus or a closed disk, let G be a partially Δ-embedded graph, L be a linkage of G, and D be a subset of Δ. We say that L is *D-avoiding* if $T(L) \cap D = \emptyset$ (see Figure 6).

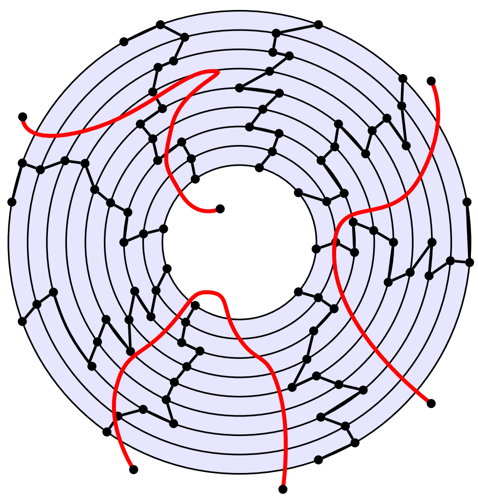

Figure 6: An example of a Δ-embedded railed annulus $\mathcal{A}$ and a linkage L (depicted in red) that is Δ-avoiding.

Linkages confined in annuli. Let $t \in \mathbb{N}_{\geq 1}$, let $p = 2t + 1$, and let $s \in [p]$ where $s = 2t' + 1$. Also, let Δ be a closed annulus and $\mathcal{A} = (\mathcal{C}, \mathcal{P})$ be a Δ-embedded (p, q)-railed annulus of a partially Δ-embedded graph G. Given some $I \subseteq [q]$, we say that a linkage L of G is *(s, I)-confined in $\mathcal{A}$* if

$$L \cap \mathsf{ann}(\mathcal{C}, t + 1 - t', t + 1 + t') \subseteq \bigcup_{i \in I} P_i.$$

Intuitively, the above definition demands that L traverses the "middle" (s, q)-annulus by intersecting it only at the rails of $\mathcal{A}$.

We now state the Linkage Combing Lemma from [103] (see also [101]). Intuitively, it says that in the presence of a "big enough" Δ-embedded railed annulus $\mathcal{A}$ in a partially Δ-embedded graph

G, where Δ is a closed annulus, every linkage of G can be "combed" through the rails of $\mathcal{A}$ in some central buffer inside $\mathcal{A}$.

Proposition 2 (Linkage Combing)**.** *There exist two functions $f_2, f_3 : \mathbb{N} \to \mathbb{N}$, where the images of f_3 are even, such that for every odd $s \in \mathbb{N}_{\geq 1}$ and every $k \in \mathbb{N}$, if*

- *Δ is a closed annulus,*
- *G is a graph that is partially Δ-embedded,*
- *$\mathcal{A} = (\mathcal{C}, \mathcal{P})$ is a Δ-embedded (p, q)-railed annulus of G, where $p \geq f_3(k) + s$ and $q \geq 5/2 \cdot f_2(k)$,*
- *L is a Δ-avoiding linkage of size at most k, and*
- *$I \subseteq [q]$, where $|I| > f_2(k)$,*

then G contains a linkage $\tilde{L}$ where $\tilde{L} \equiv L$, $\tilde{L} \setminus \Delta \subseteq L \setminus \Delta$, and $\tilde{L}$ is (s, I)-confined in $\mathcal{A}$. Moreover, $f_3(k) = \mathcal{O}((f_2(k))^2)$.

5.3 Linkages in boundaried graphs

In this subsection, we aim to define the set of *models* of a boundaried graph, that is all linkages that can be routed inside this graph and contain the boundary vertices as terminals. This collection of graphs, can be encoded in abstract terms of collections of binary relations between indices of the boundaried vertices, representing the existence of (disjoint) paths between the corresponding boundary vertices. This encoding is, in fact, present in the encoding of the pattern (see Observation 7).

Linkages of boundaried graphs. A *pairing* is a t-boundaried graph $\mathbf{L} = (L, v_1, \ldots, v_t)$ where L is a linkage and $T(L) \subseteq \{v_1, \ldots, v_t\}$. We use $\mathcal{B}^{(t)}_{\mathsf{pair}}$ to denote the set of all (pairwise non-isomorphic) t-boundaried graphs that are pairings. A path of a linkage L is *non-trivial* if it is not a single edge. Given a k-boundaried graph $(G, u_1, \ldots, u_k)$, a linkage L of G and some $v_1, \ldots, v_t \in T(L) \cup \{\text{\textvisiblespace}\}$ such that $\{u_1, \ldots, u_k\} \subseteq \{v_1, \ldots, v_t\}$, we say that $(L, v_1, \ldots, v_t)$ is a *boundaried linkage of* $(G, u_1, \ldots, u_k)$.

Let $(G, v_1, \ldots, v_t)$ be a t-boundaried graph. We define the set of *pairings* of $(G, v_1, \ldots, v_t)$, denoted by $\mathsf{Pairings}(G, v_1, \ldots, v_t)$, to be the set

$$\mathsf{Pairings}(G, v_1, \ldots, v_t) = \left\{ (L, v_1, \ldots, v_t) \in \mathcal{B}^{(t)}_{\mathsf{pair}} \;\middle|\; \begin{array}{l} (L, v_1, \ldots, v_t) \text{ is a boundaried linkage} \\ \text{of } (G, v_1, \ldots, v_t) \text{ and every path of } L \\ \text{is non-trivial} \end{array} \right\}.$$

Imprint of linkages. Let $r, h, l \in \mathbb{N}$. Let $\mathbf{G} = (G, X_1, \ldots, X_h, \mathbf{a}, v_1, \ldots, v_r) \in \mathcal{B}^{(r,h,l)}$. Given an $\mathbf{L} \in \mathsf{Pairings}(G, v_1, \ldots, v_r)$, we define the *imprint* of $\mathbf{L}$, denoted by $\mathsf{imp}(\mathbf{L})$, to be the graph whose vertex set is $I = \{i \in [r] \mid v_i \neq \text{\textvisiblespace}\}$ and two vertices $i, j \in I$ are adjacent if $\{v_i, v_j\}$ belongs to the pattern of L. We define the *compression* of $\mathbf{G}$, denoted by $\mathsf{compression}(\mathbf{G})$ to be the quintuple $(\mathcal{V}, \kappa, \delta, H^e, \mathcal{H}^P)$, where

- $\mathcal{V}$ is the partition of I to sets of pairwise equal vertices,
- κ is the partial function mapping each $i \in I$ to $j \in [l]$ such that $v_i = a_j$,
- δ is the partial function mapping each $i \in I$ to $j \in [h]$ such that $v_i \in X_j$,

- $H^e = \mathsf{Ind}_G(I)$, and
- $\mathcal{H}^P = \{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(G, v_1, \ldots, v_r)\}$.

It is easy to observe the following.

Observation 7. *For every* $\mathbf{G} \in \mathcal{B}^{(r,h,l)}$, $\mathsf{compression}(\mathbf{G}) = \mathsf{pattern}(\mathbf{G})$.

We stress that the only difference between the collection $\mathcal{H}^P$ of imprints of all pairings of $(G, v_1, \ldots, v_r)$ and the set $\mathsf{Pairings}(G, v_1, \ldots, v_r)$ is that the first is a collection of graphs whose vertex set is the set of indices $I \subseteq [r]$, while graphs in $\mathsf{Pairings}(G, v_1, \ldots, v_r)$ are subgraphs of G. The distinction between the two is essential in order to 1) encode the patterns of linkages "abstractly" (in terms of graphs on indices) and 2) decode the presence of same variety of linkages inside graphs with the same pattern.

Before concluding this section, we present some additional definitions on linkages of boundaried colored graphs and the reason is the following. In Section 7, we describe how, given a colored graph partially embedded in a "big enough" railed annulus, construct (a series of) boundaried graphs whose boundary vertices will also be some vertices of the rails of the annuli. These boundary vertices will be chosen to be the "few" vertices in which every linkage can be combed, due to Proposition 2. Therefore, as we are about to prove in Section 7 (in particular, Lemma 4), after combing, linkages can be separated on a "left" and a "right" part. Therefore, we need to define a way to *glue* pairings.

Gluing pairings. We now define compatibility between pairings. Let $\ell, r \in \mathbb{N}$. Let $\mathbf{L} = (L, u_1, \ldots, u_{r+\ell})$ and $\mathbf{L}' = (L', v_1, \ldots, v_{r+\ell})$ be two pairings. We say that $\mathbf{L}$ and $\mathbf{L}'$ are *ℓ-compatible* if $u_{r+1}, \ldots, u_{r+\ell} \in T(L)$, $v_{r+1}, \ldots, v_{r+\ell} \in T(L')$, for every $i \in [r]$, $u_i = \textvisiblespace$ if and only if $v_i \neq \textvisiblespace$, and for every $i, j \in [\ell]$, $\{v_{r+i}, v_{r+j}\}$ is in the pattern of L if and only if $\{v_{r+i}, v_{r+j}\}$ is not in the pattern of L'. Given two families $\mathcal{F}_1, \mathcal{F}_2 \subseteq \mathcal{B}^{(r+\ell)}$ of pairings, we say that $\mathcal{F}_1$ and $\mathcal{F}_2$ are *ℓ-compatible* if for every $\mathbf{L}_1 \in \mathcal{F}_1$ and every $\mathbf{L}_2 \in \mathcal{F}_2$, $\mathbf{L}_1$ and $\mathbf{L}_2$ are ℓ-compatible.

Let $\mathbf{L}_1 = (L_1, u_1, \ldots, u_{r+\ell})$ and $\mathbf{L}_2 = (L_2, v_1, \ldots, v_{r+\ell})$ be two ℓ-compatible pairings. We denote by $(L_1, u_1, \ldots, u_{r+\ell}) \oplus_\ell (L_2, v_1, \ldots, v_{r+\ell})$ the pairing $(L, c_1, \ldots, c_r)$, where

- L is the linkage obtained from the disjoint union of L_1 and L_2 after identifying, for every $i \in [r+1, \ell]$, the vertices u_i and v_i, and
- $c_i = u_i$, if $u_i \in T(L_1)$, while $c_i = v_i$, if $v_i \in T(L_2)$ (recall that by definition of compatibility, for every $i \in [r]$, at least one of u_i and v_i is equal to $\textvisiblespace$ but not both).

Given two ℓ-compatible families $\mathcal{F}_1, \mathcal{F}_2 \subseteq \mathcal{B}^{(r+\ell)}$ of pairings, we denote by $\mathcal{F}_1 \oplus_\ell \mathcal{F}_2$ the collection $\{\mathbf{L}_1 \oplus_\ell \mathbf{L}_2 \mid \mathbf{L}_1 \in \mathcal{F}_1 \text{ and } \mathbf{L}_2 \in \mathcal{F}_2\}$.

6 Dealing with apices

In this section, we show how to deal with apex vertices that can be adjacent to vertices in a flat area of the graph in a completely arbitrary way. In Subsection 6.1, we define a certain "transformation" from a colored graph to another colored graph (with more colors), interpreting neighborhoods of some predetermined vertices as new colors. Moreover, in Subsection 6.2, we describe how to get an equivalent version of a formula in this "projected" setting (see Lemma 3). This will allow us to define an equivalent version of any given formula in $\mathsf{FOL}[\tau{+}\mathsf{dp}]$ that will transform a question from a flat graph with apices to a flat graph without apices (where adjacencies with apices are interpreted as colors in the graph).

6.1 Projections of graphs with respect to some apex-tuple

In this subsection, we define a way to "project" a given colored graph with respect to a given apex-tuple and a way to define the "projected" version of a sentence such that the initial colored graph satisfies the initial sentence if and only if the "projected" colored graph satisfies the "projected" sentence (see Lemma 3). This transformation will allow us to work with the projected colored graphs, where the absence of apices in a graph of big enough treewidth implies a flat bidimensional structure (see Proposition 9 in Appendix B). Flatness is particularly critical for our techniques, as it will be explained in Section 7.

We proceed to formalize the idea of "projecting" a structure with respect to an apex-tuple. First, we describe what is the vocabulary of the "projected" structure.

Constant-projections of vocabularies. Let τ be a colored-graph vocabulary, let $l \in \mathbb{N}$, and let $\mathbf{c} = \{\mathsf{c}_1, \ldots, \mathsf{c}_l\}$ be a collection of l constant symbols. We define the *constant-projection* $\tau^{\langle \mathbf{c} \rangle}$ of $(\tau \cup \mathbf{c})$ to be the vocabulary $(\tau \cup \mathbf{c} \cup \{\mathsf{C}_1, \ldots, \mathsf{C}_l\})$, where $\mathsf{C}_1, \ldots, \mathsf{C}_l$ are unary relation symbols not contained in τ.

Given a colored-graph vocabulary τ and some collection of constant symbols $\mathbf{c}$, we proceed to define a way to construct a $\tau^{\langle \mathbf{c} \rangle}$-structure from a given $(\tau \cup \mathbf{c})$-structure $(\mathfrak{G}, \mathbf{a})$. The obtained $\tau^{\langle \mathbf{c} \rangle}$-structure is the "projection" of $(\mathfrak{G}, \mathbf{a})$ with respect to the apex-tuple $\mathbf{a}$.

Projecting a colored graph with respect to an apex-tuple. Let τ be a colored-graph vocabulary, let $l \in \mathbb{N}$, and let $\mathbf{c} = \{\mathsf{c}_1, \ldots, \mathsf{c}_l\}$ be a collection of l constant symbols. Let also $\tau^{\langle \mathbf{c} \rangle}$ be the constant-projection of $(\tau \cup \mathbf{c})$. Given a $(\tau \cup \mathbf{c})$-structure $(\mathfrak{G}, \mathbf{a})$, where $\mathbf{a} = (a_1, \ldots, a_l)$ is an apex-tuple of G of size l and, for every $i \in [l]$, $\mathsf{c}_i^{(\mathfrak{G}, \mathbf{a})} = a_i$, we define the structure $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}, \mathbf{a})$ to be the $\tau^{\langle \mathbf{c} \rangle}$-structure obtained as follows:

- $V(\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}, \mathbf{a})) = V(G)$,
- for every $i \in [l]$ $\mathsf{c}_i^{\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}, \mathbf{a})} = a_i$,
- $\mathsf{E}^{\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}, \mathbf{a})} = \mathsf{E}^{\mathfrak{G}} \cap ((V(G) \setminus V(\mathbf{a}))^2 \cup V(\mathbf{a})^2)$,
- every $\mathsf{R} \in \tau \setminus \{\mathsf{E}\}$ is interpreted in $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}, \mathbf{a})$ as in $\mathfrak{G}$, and
- for every $i \in [l]$, C_i is interpreted in $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}, \mathbf{a})$ as $N_G(a_i) \setminus V(\mathbf{a})$.

Notice that if $a_i = \text{\textvisiblespace}$, C_i is interpreted in $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}, \mathbf{a})$ as the empty set. Intuitively, we introduce a color $C_i, i \in [a]$ for each a_i. We keep the same universe, we keep only edges that are either between apices or between non-apices, and we color the neighbors of a_i by color C_i. See Figure 7 for an example.

6.2 Apex-projected sentences

Having defined the structure $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}, \mathbf{a})$, we now define for every sentence $\varphi \in \mathsf{FOL}[\tau{+}\mathsf{dp}]$ its *l-apex-projected sentence* φ^l. This will be a sentence in $\mathsf{FOL}[\tau^{\langle \mathbf{c} \rangle} + \mathsf{dp}]$ (see Observation 8) and can be seen as the equivalent (to φ) question that is asked to be satisfied by $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}, \mathbf{a})$.

Let τ be a colored-graph vocabulary, let $l \in \mathbb{N}$, and let $\mathbf{c} = \{\mathsf{c}_1, \ldots, \mathsf{c}_l\}$ be a collection of l constant symbols. Given a set S and an $\ell \in \mathbb{N}_{\geq 1}$, we use $\mathcal{P}_\ell(S)$ to denote all partitions of S into ℓ parts, i.e., all collections of ℓ pairwise disjoint subsets $S_1, \ldots, S_\ell$ of S such that $\bigcup_{i \in [\ell]} S_i = S$.

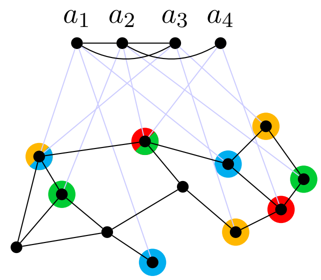

Figure 7: An example of a graph G with an apex-tuple $\mathbf{a} = (a_1, \ldots, a_4)$. The set C_1 contains all cyan vertices (the neighbors of a_1), the set C_2 contains all green vertices, the set C_3 contains all orange vertices and the set C_4 contains all red vertices. Vertices that are neighbors to more than one a_i are depicted with multiple colors. The structure $\mathsf{ap}_{\mathsf{c}}(\mathfrak{G}, \mathbf{a})$ is obtained from G after introducing colors $C_i, i \in [4]$ and after removing all edges between a vertex in $\{a_1, \ldots, a_4\}$ and a vertex in $V(G) \setminus \{a_1, \ldots, a_4\}$.

For every sentence $\varphi \in \mathsf{FOL}[\tau + \mathsf{dp}]$, we define its *$l$-apex-projected sentence* φ^l to be the sentence obtained from φ by replacing each atomic formula $\mathsf{E}(\mathsf{x}, \mathsf{y})$ by

$$\mathsf{E}(\mathsf{x}, \mathsf{y}) \vee \bigvee_{i \in [l]} \big((\mathsf{x} = \mathsf{c}_i \wedge \mathsf{y} \in \mathsf{C}_i) \vee (\mathsf{y} = \mathsf{c}_i \wedge \mathsf{x} \in \mathsf{C}_i) \big)$$

and each atomic formula $\mathsf{dp}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$ by the formula $\zeta_{\mathsf{dp}}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$ that we proceed to define. We set $\mathsf{distinct}(\mathsf{x}_1, \ldots, \mathsf{x}_t) := \bigwedge_{i,j \in [t], i \neq j} (\mathsf{x}_i \neq \mathsf{x}_j)$. We define $\zeta_{\mathsf{dp}}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$ as follows (see next paragraph for an intuitive explanation):

$$\bigvee_{B\subseteq[l]}\ \bigvee_{d\in[k]}\ \bigvee_{(I_1,\dots,I_d)\in\mathcal{P}_d(B)}\ \bigvee_{\substack{\text{all injective functions}\\ \rho:\{I_1,\dots,I_d\}\to[k]}} \tag{2}$$

$$\Big(\exists\, \mathsf{x}_1^{I_1},\dots,\mathsf{x}_{|I_1|}^{I_1},\dots,\mathsf{x}_1^{I_d},\dots,\mathsf{x}_{|I_d|}^{I_d} : \mathsf{distinct}(\mathsf{x}_1^{I_1},\dots,\mathsf{x}_{|I_1|}^{I_1},\dots,\mathsf{x}_1^{I_d},\dots,\mathsf{x}_{|I_d|}^{I_d}) \tag{3}$$

$$\wedge \bigwedge_{\substack{i,i'\in[d],\\ i\neq i'}}\ \bigwedge_{j\in[2,|I_i|-1]} (\mathsf{x}_j^{I_i}\neq \mathsf{s}_{\rho(I_{i'})}\wedge \mathsf{x}_j^{I_i}\neq \mathsf{t}_{\rho(I_{i'})}) \tag{4}$$

$$\wedge \bigwedge_{i\in[d]}\ \bigvee_{\substack{\text{all bijections}\\ \lambda_{I_i}:[|I_i|]\to I_i}} \Big(\bigwedge_{i\in[d]}\ \bigwedge_{j\in[|I_i|]} \mathsf{x}_j^{I_i}=\mathsf{c}_{\lambda_{I_i}(j)} \tag{5}$$

$$\wedge \bigwedge_{i\in[d]}\ \bigvee_{J_i\subseteq[|I_i|-1]} \Big(\bigwedge_{i\in[d]}\ \bigwedge_{j\in[|I_i|-1]\setminus J_i} \mathsf{E}(\mathsf{x}_j^{I_i},\mathsf{x}_{j+1}^{I_i}) \tag{6}$$

$$\wedge\ \exists_{j\in J_1}\mathsf{y}_j^{I_1},\dots,\exists_{j\in J_d}\mathsf{y}_j^{I_d},\exists_{j\in J_1}\mathsf{z}_j^{I_1},\dots,\exists_{j\in J_d}\mathsf{z}_j^{I_d}, \tag{7}$$

$$\mathsf{distinct}((\mathsf{y}_j^{I_1})_{j\in J_1},\dots,(\mathsf{y}_j^{I_d})_{j\in J_d},\mathsf{s}_1,\mathsf{t}_1,\dots,\mathsf{s}_k,\mathsf{t}_k) \tag{8}$$

$$\wedge\ \mathsf{distinct}((\mathsf{z}_j^{I_1})_{j\in J_1},\dots,(\mathsf{z}_j^{I_d})_{j\in J_d},\mathsf{s}_1,\mathsf{t}_1,\dots,\mathsf{s}_k,\mathsf{t}_k) \tag{9}$$

$$\wedge \bigwedge_{\substack{i,i'\in[d],\\ i\neq i'}}\ \bigwedge_{\substack{j\in J_i,\\ j'\in J_{i'}}} (\mathsf{y}_j^{I_i}\neq \mathsf{z}_{j'}^{I_{i'}}) \wedge \bigwedge_{i\in[d]}\ \bigwedge_{\substack{j,j'\in J_i,\\ j\neq j'}} (\mathsf{y}_j^{I_i}\neq \mathsf{z}_{j'}^{I_i}) \tag{10}$$

$$\wedge\Big(\bigwedge_{i\in[d]}\ \bigwedge_{j\in J_i} (\mathsf{y}_j^{I_i}\in \mathsf{C}_{\lambda_{I_i}(j)}\wedge \mathsf{z}_j^{I_i}\in \mathsf{C}_{\lambda_{I_i}(j+1)})\ \wedge \tag{11}$$

$$\bigvee_{(X_i,Y_i,Z_i)\in\mathcal{P}_3(J_i)} \Big(\bigwedge_{j\in X_i} (\mathsf{y}_j^{I_i}=\mathsf{z}_j^{I_i}) \wedge \bigwedge_{j\in Y_i} \mathsf{E}(\mathsf{y}_j^{I_i},\mathsf{z}_j^{I_i}) \wedge\ \psi_{I_i,\rho,Z_i})\Big)\Big)\Big)\Big), \tag{12}$$

where ψ_{I_i,ρ,Z_i} is the formula

$$\big(\mathsf{s}_{\rho(I_i)}=\mathsf{x}_1^{I_i}\wedge \mathsf{t}_{\rho(I_i)}=\mathsf{x}_{|I_i|}^{I_i}\wedge\ \mathsf{dp}(\xi_{I_i,\rho,Z_i}(\mathsf{s}_1,\mathsf{t}_1,\dots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\dots,\mathsf{c}_l,\mathsf{c}_l))\big)$$

$$\vee$$

$$\big(\mathsf{s}_{\rho(I_i)}\neq \mathsf{x}_1^{I_i}\wedge \mathsf{t}_{\rho(I_i)}=\mathsf{x}_{|I_i|}^{I_i}$$
$$\wedge\ \exists\ \tilde{\mathsf{s}}_{\rho(I_i)}\in \mathsf{C}_{\lambda_{I_i}(1)}\wedge \mathsf{distinct}(\tilde{\mathsf{s}}_{\rho(I_i)},(\mathsf{s}_j,\mathsf{t}_j)_{j\neq\rho(I_i)})\wedge\ \mathsf{dp}(\kappa_{I_i,\rho,Z_i}(\mathsf{s}_1,\mathsf{t}_1,\dots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\dots,\mathsf{c}_l,\mathsf{c}_l))\big)$$

$$\vee$$

$$\big(\mathsf{s}_{\rho(I_i)}=\mathsf{x}_1^{I_i}\wedge \mathsf{t}_{\rho(I_i)}\neq \mathsf{x}_{|I_i|}^{I_i}$$
$$\wedge\ \exists\ \tilde{\mathsf{t}}_{\rho(I_i)}\in \mathsf{C}_{\lambda_{I_i}(|I_i|)}\wedge \mathsf{distinct}(\tilde{\mathsf{t}}_{\rho(I_i)},(\mathsf{s}_j,\mathsf{t}_j)_{j\neq\rho(I_i)})\wedge\ \mathsf{dp}(\delta_{I_i,\rho,Z_i}(\mathsf{s}_1,\mathsf{t}_1,\dots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\dots,\mathsf{c}_l,\mathsf{c}_l))\big)$$

$$\vee$$

$$\big(\mathsf{s}_{\rho(I_i)}\neq \mathsf{x}_1^{I_i}\wedge \mathsf{t}_{\rho(I_i)}\neq \mathsf{x}_{|I_i|}^{I_i}$$
$$\wedge\ \exists\ \tilde{\mathsf{s}}_{\rho(I_i)}\in \mathsf{C}_{\lambda_{I_i}(1)}\wedge \mathsf{distinct}(\tilde{\mathsf{s}}_{\rho(I_i)},(\mathsf{s}_j,\mathsf{t}_j)_{j\neq\rho(I_i)})$$
$$\wedge\ \exists\ \tilde{\mathsf{t}}_{\rho(I_i)}\in \mathsf{C}_{\lambda_{I_i}(|I_i|)}\ \wedge \mathsf{distinct}(\tilde{\mathsf{t}}_{\rho(I_i)},(\mathsf{s}_j,\mathsf{t}_j)_{j\neq\rho(I_i)})\wedge \mathsf{dp}(\zeta_{I_i,\rho,Z_i}(\mathsf{s}_1,\mathsf{t}_1,\dots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\dots,\mathsf{c}_l,\mathsf{c}_l))\big),$$

and

- $\xi_{I_i,\rho,Z_i}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\ldots,\mathsf{c}_l,\mathsf{c}_l)$ is used to denote the tuple obtained from the tuple $(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\ldots,\mathsf{c}_l,\mathsf{c}_l)$ after removing, for every $i \in [\ell]$, $(\mathsf{s}_{\rho(I_i)},\mathsf{t}_{\rho(I_i)})$ and adding $(\mathsf{y}_j^{I_i},\mathsf{z}_{j+1}^{I_i})$, for all $j \in Z_i$.
- $\kappa_{I_i,\rho,Z_i}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\ldots,\mathsf{c}_l,\mathsf{c}_l)$ is used to denote the tuple obtained from the tuple $(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\ldots,\mathsf{c}_l,\mathsf{c}_l)$ after removing, for every $i \in [\ell]$, $(\mathsf{s}_{\rho(I_i)},\mathsf{t}_{\rho(I_i)})$ and adding $(\mathsf{s}_{\rho(I_i)},\tilde{\mathsf{s}}_{\rho(I_i)})$ and $(\mathsf{y}_j^{I_i},\mathsf{z}_{j+1}^{I_i})$, for all $j \in Z_i$.
- $\delta_{I_i,\rho,Z_i}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\ldots,\mathsf{c}_l,\mathsf{c}_l)$ is used to denote the tuple obtained from the tuple $(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\ldots,\mathsf{c}_l,\mathsf{c}_l)$ after removing, for every $i \in [\ell]$, $(\mathsf{s}_{\rho(I_i)},\mathsf{t}_{\rho(I_i)})$ and adding $(\mathsf{t}_{\rho(I_i)},\tilde{\mathsf{t}}_{\rho(I_i)})$ and $(\mathsf{y}_j^{I_i},\mathsf{z}_{j+1}^{I_i})$, for all $j \in Z_i$.
- $\zeta_{I_i,\rho,Z_i}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\ldots,\mathsf{c}_l,\mathsf{c}_l)$ is used to denote the tuple obtained from the tuple $(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k,\mathsf{c}_1,\mathsf{c}_1,\ldots,\mathsf{c}_l,\mathsf{c}_l)$ after removing, for every $i \in [\ell]$, $(\mathsf{s}_{\rho(I_i)},\mathsf{t}_{\rho(I_i)})$ and adding $(\mathsf{s}_{\rho(I_i)},\tilde{\mathsf{s}}_{\rho(I_i)}),(\mathsf{t}_{\rho(I_i)},\tilde{\mathsf{t}}_{\rho(I_i)})$, and $(\mathsf{y}_j^{I_i},\mathsf{z}_{j+1}^{I_i})$, for all $j \in Z_i$.

Intuitive explanation of the above formulas. We decode step by step the intuition behind the above formulas. First, we replace $\mathsf{E}(\mathsf{x},\mathsf{y})$ by $\mathsf{E}(\mathsf{x},\mathsf{y}) \vee \bigvee_{i\in[l]} \big((\mathsf{x}=\mathsf{c}_i \wedge \mathsf{y} \in \mathsf{C}_i) \vee (\mathsf{y}=\mathsf{c}_i \wedge \mathsf{x} \in \mathsf{C}_i)\big)$ in order to encode that, in the colored graph obtained after the removal of all edges between apex-vertices and the rest vertices of the graph, two vertices x, y are adjacent if either the edge $\{x,y\}$ is present in the modified graph or one of x and y is an apex-vertex a_i (interpreting c_i) and the other is colored by the corresponding color C_i, that is the color that all neighbors of a_i receive.

For the the atomic formula $\zeta_{\mathsf{dp}}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k)$, the intuition is the following. We want to separate the formula into many parts, i.e., many questions for disjoint paths or adjacencies, guessing whether the variables $\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k$ are assigned to apex-vertices and/or whether the apex-vertices are internal vertices of the disjoint paths between $\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_k,\mathsf{t}_k$.

For this reason, in line (2) of the above definition, and in particular in "$\bigvee_{B\subseteq[l]}$", we start by guessing the subset B of apex-vertices that are part of the disjoint paths (either as endpoints or as internal vertices). We will refer to this set as *active apices*. Then, with "$\bigvee_{d\in[k]}$", we guess how many among the k disjoint paths contain active apices (and we call them *active paths*) and then, with "$\bigvee_{(I_1,\ldots,I_d)\in\mathcal{P}_d(B)}$" we guess how B is partitioned in d sets, each set corresponding to active apices that belong to the same active path and with "$\bigvee_{\text{all injective functions } \rho:\{I_1,\ldots,I_d\}\to[k]}$", we guess which active apices belong to each active path.

Having made all these guesses, in line (3), we ask for variables ("$\exists\, \mathsf{x}_1^{I_1},\ldots,\mathsf{x}_{|I_1|}^{I_1},\ldots,\mathsf{x}_1^{I_d},\ldots,\mathsf{x}_{|I_d|}^{I_d}$") that will be interpreted as the active apices and we ask these variables to be interpreted as pairwise disjoint vertices ("$\mathsf{distinct}(\mathsf{x}_1^{I_1},\ldots,\mathsf{x}_{|I_1|}^{I_1},\ldots,\mathsf{x}_1^{I_d},\ldots,\mathsf{x}_{|I_d|}^{I_d})$"). Also, in line (4), we ask that all vertices that interpret $\mathsf{x}_j^{I_i}$ for $i \in [d]$ and $j \in [2, |I_i|-1]$ are different from the endpoints of all the other active paths ("$\bigwedge_{i,i'\in[d],i\neq i'} \bigwedge_{j\in[2,|I_i|-1]} (\mathsf{x}_j^{I_i} \neq \mathsf{s}_{\rho(I_{i'})} \wedge \mathsf{x}_j^{I_i} \neq \mathsf{t}_{\rho(I_{i'})})$"). The two later properties are necessary for the disjointness of the demanded paths. For the moment, we *do not* demand the interpretations of $\mathsf{x}_1^{I_i}$ and $\mathsf{x}_{|I_i|}^{I_i}$, for any $i \in [d]$, to be distinct from the endpoints of the paths.

To express that these variables correspond to apices, in line (5), first we guess to which element of B each $\mathsf{x}_j^{I_i}$ corresponds ("$\bigwedge_{i\in[d]} \bigvee_{\text{all bijections } \lambda_{I_i}:[|I_i|]\to I_i}$"). We stress that the order of $\mathsf{x}_j^{I_i}$ is fixed and implicitly corresponds to the order that active apices are transversed by the corresponding active path. For each $i \in [d]$, the bijection λ_{I_i} is used to correspond the ascending indices j of the variables $\mathsf{x}_j^{I_i}$ to the actual active apices. Then, with "$\bigwedge_{i\in[d]} \bigwedge_{j\in[|I_i|]} \mathsf{x}_j^{I_i} = \mathsf{c}_{\lambda_{I_i}(j)}$", we check whether this guess indeed corresponds to an interpretation of each $\mathsf{x}_j^{I_i}$ with the appropriate active apex $a_{\lambda_{I_i}(j)}$.

In line (6), we partition active apices into two sets. First, we have the active apices whose next neighbor on the corresponding active path is *not* an apex-vertex (we orient paths according to the ordering given by the ascending ordering of the indices of $\mathsf{x}_j^{I_i}$). These active apices are guessed using "$\bigwedge_{\in[d]} \bigvee_{J_i \subseteq [|I_i|-1]}$". These, we call them *shifting active apices*. For the remaining ones ("$\bigwedge_{i\in[d]} \bigwedge_{j\in[|I_i|-1]\setminus J_i}$"), we check if indeed their next neighbor on the corresponding active path *is* an apex-vertex ("$\mathsf{E}(\mathsf{x}_j^{I_i}, \mathsf{x}_{j+1}^{I_i})$").

Now, in line (7), we deal with the shifting active apices (apices that would be transversed by path that would be routed through at least one non-apex vertex before entering again the set of apex vertices). For each shifting active apex, we ask for the existence of two extra vertices, corresponding to the first vertex after this shifting active apex and the last vertex before the next apex in a supposed path. This is done in "$\exists_{j\in J_1} \mathsf{y}_j^{I_1}, \ldots, \exists_{j\in J_d} \mathsf{y}_j^{I_d}, \exists_{j\in J_1} \mathsf{z}_j^{I_1}, \ldots, \exists_{j\in J_d} \mathsf{z}_j^{I_d}$".

The variables $\mathsf{y}_j^{I_i}$, $i \in [d]$, $j \in J_i$ and the variables $\mathsf{z}_j^{I_i}$, $i \in [d]$, $j \in J_i$ should be interpreted as pairwise disjoint vertices that are also different from the endpoints of all the other (active or not) paths. We check this in lines (8) and (9) using "$\mathsf{distinct}((\mathsf{y}_j^{I_1})_{j\in J_1}, \ldots, (\mathsf{y}_j^{I_d})_{j\in J_d}, \mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$" and "$\mathsf{distinct}((\mathsf{z}_j^{I_1})_{j\in J_1}, \ldots, (\mathsf{z}_j^{I_d})_{j\in J_d}, \mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$". Also, we demand that variables $\mathsf{y}_j^{I_i}$ and $\mathsf{z}_{j'}^{I_{i'}}$ are interpreted as disjoint vertices if they correspond to different paths (i.e., $i \neq i'$) or if they belong to the same path but they are not consecutive (i.e., $j \neq j'$). This is done in line (10).

For every $i \in [d]$, the vertices interpreting $\mathsf{y}_j^{I_i}, j \in J_i$ should appear in the corresponding active path directly *after* the corresponding shifting active apex $a_{\lambda_{I_i}(j)}$ and the vertices interpreting $\mathsf{z}_j^{I_i}, j \in J_i$ should appear in the corresponding active path directly *before* the next corresponding apex $a_{\lambda_{I_i}(j+1)}$. Since edges between apices and non-apices are no longer present in the graph, we encode this "succession" by using the colors of the neighborhood of the apices. This is done in "$\bigwedge_{i\in[d]} \bigwedge_{j\in J_i} (\mathsf{y}_j^{I_i} \in \mathsf{C}_{\lambda_{I_i}(j)} \wedge \mathsf{z}_j^{I_i} \in \mathsf{C}_{\lambda_{I_i}(j+1)})$", in line (11).

At this point, we have dealt with the internal part of the paths (we have not yet discussed what happens with the endpoints; we will do so in the next paragraph) for what concerns the apices. In fact, we already explained how to guess which part of the apices will be part of the supposed disjoint paths (lines (2)-(5)), how to guess which part of the paths is routed *only* through apices (line (6)), and where the paths "exit" and "enter" the set of apices (lines (7)-(11)). What remains is to describe how to formulate the question on the graph without the apices and how we deal with the endpoints of the paths.

In fact, we already mentioned that, for every $i \in [d]$ and every $j \in [d]$, $\mathsf{y}_j^{I_i}$ and $\mathsf{z}_j^{I_i}$ are interpreted as two vertices that are not apices and for the supposed path corresponding to index i, the part between the interpretations of $\mathsf{y}_j^{I_i}$ and $\mathsf{z}_j^{I_i}$ is a maximal path that does not contain apices. Having this in mind, in line (12), we guess what is the relation between $\mathsf{y}_j^{I_i}$ and $\mathsf{z}_j^{I_i}$, for all $j \in J_i$. We partition J_i to three sets X_i, Y_i, and Z_i ("$\bigvee_{(X_i,Y_i,Z_i)\in\mathcal{P}_3(J_i)}$"). The set X_i contains all indices j for the demanded path that passes through the apices $a_{\lambda_{I_i}(j)}$ and $a_{\lambda_{I_i}(j+1)}$ has to be routed through a single vertex between $a_{\lambda_{I_i}(j)}$ and $a_{\lambda_{I_i}(j+1)}$, or, in other words, $\mathsf{y}_j^{I_i}$ and $\mathsf{z}_j^{I_i}$ are asked to be interpreted as the same vertex. ("$\bigwedge_{j\in X_i} (\mathsf{y}_j^{I_i} = \mathsf{z}_j^{I_i})$"). The set Y_i contains all indices j for which the demanded path that passes through the apices $a_{\lambda_{I_i}(j)}$ and $a_{\lambda_{I_i}(j+1)}$ has to be routed through an edge connecting the interpretations of $\mathsf{y}_j^{I_i}$ and $\mathsf{z}_j^{I_i}$ ("$\bigwedge_{j\in Y_i} \mathsf{E}(\mathsf{y}_j^{I_i}, \mathsf{z}_j^{I_i})$"). Finally, Z_i contains all remaining $j \in J_i$, i.e., all indices j for which the demanded path has to be routed through path of length two between the interpretations of $\mathsf{y}_j^{I_i}$ and $\mathsf{z}_j^{I_i}$.

To finish the description of the formula $\zeta_{\mathsf{dp}}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$, we have to discuss how the final new demands for disjoint paths are formulated. This is encoded in the formula ψ_{I_i,ρ,Z_i}. There, for each active path, say indexed by $\rho(I_i)$, we have to "update" the demand for a path from $\mathsf{s}_{\rho(I_i)}$ to

$\mathsf{t}_{\rho(I_i)}$ to the demand for paths between $\mathsf{y}_j^{I_i}$ and $\mathsf{z}_{j+1}^{I_i}$), for all $j \in Z_i$ and this update has to be done for all active paths.

In the last argument of the previous paragraph, we omitted some important detail. The aforementioned update is correct (in the sense that the two questions for disjoint paths are equivalent) only if, for all active paths, its first and last active apex are its endpoints, i.e., if $\mathsf{s}_{\rho(I_i)} = \mathsf{x}_1^{I_i}$ and $\mathsf{t}_{\rho(I_i)} = \mathsf{x}_{|I_i|}^{I_i}$ are true. This is why the first disjunctive term of ψ_{I_i,ρ,Z_i} is "$\mathsf{s}_{\rho(I_i)} = \mathsf{x}_1^{I_i} \wedge \mathsf{t}_{\rho(I_i)} = \mathsf{x}_{|I_i|}^{I_i} \wedge \mathsf{dp}(\xi_{i,\rho,Z_i}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k, \mathsf{c}_1, \mathsf{c}_1, \ldots, \mathsf{c}_l, \mathsf{c}_l))$", where $\xi_{I_i,\rho,Z_i}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k, \mathsf{c}_1, \mathsf{c}_1, \ldots, \mathsf{c}_l, \mathsf{c}_l)$ is the tuple obtained from $(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k, \mathsf{c}_1, \mathsf{c}_1, \ldots, \mathsf{c}_l, \mathsf{c}_l)$ after removing, for every $i \in [\ell]$, $(\mathsf{s}_{\rho(I_i)}, \mathsf{t}_{\rho(I_i)})$ and adding $(\mathsf{y}_j^{I_i}, \mathsf{z}_{j+1}^{I_i})$, for all $j \in Z_i$. Since all non-shifting active apices are adjacent (see line (6)) and for all $j \in J_i \setminus Z_i$, the interpretations of $\mathsf{y}_j^{I_i}$ and $\mathsf{z}_{j+1}^{I_i}$) are either identical or adjacent (see line (12)), what remains is to find disjoint paths between the interpretations of $\mathsf{y}_j^{I_i}$ and $\mathsf{z}_{j+1}^{I_i}$), for every $j \in Z_i$. Of course, having dealt with the case that paths contains apices, we want the new paths that we search to be disjoint from all apices. To express this, we add "$\mathsf{c}_1, \mathsf{c}_1, \ldots, \mathsf{c}_l, \mathsf{c}_l$" to the above tuples, asking that the rest (disjoint) paths are also disjoint from the path of zero lenght that starts and finishes to the intepretation of c_i, for every $i \in [l]$.

To deal with the case where, for an active paths, either its first or last active apex are not its endpoints, we have to guess the existence of an extra "entering" or "exiting" point, respectively, and ask for some supplementary disjoint path between this new guessed vertex and the corresponding endpoint (see last three disjunctive terms of ψ_{I_i,ρ,Z_i}). This concludes the intuitive explanation of the formula $\zeta_{\mathsf{dp}}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$.

Note that if $\varphi \in \mathsf{FOL}[\tau+\mathsf{dp}]$ and $\mathbf{c}$ is a collection of l constant symbols, then $\varphi^l \in \mathsf{FOL}[\tau^{\langle \mathbf{c} \rangle} + \mathsf{dp}]$.

Observation 8. *Let τ be a colored-graph vocabulary. For every $\varphi \in \mathsf{FOL}[\tau+\mathsf{dp}]$, every $l \in \mathbb{N}$, and every collection $\mathbf{c}$ of l constant symbols, $\varphi^l \in \mathsf{FOL}[\tau^{\langle \mathbf{c} \rangle} + \mathsf{dp}]$.*

Also, note that in the definition of φ^l, we add some extra quantified first-order variables (see the definition of $\zeta_{\mathsf{dp}}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$; line (3), line (7), and the definition of ψ_{I_i,ρ,Z_i}).

Observation 9. *Let τ be a colored-graph vocabulary and let $r, l \in \mathbb{N}$. There is a function $f_4 : \mathbb{N}^2 \to \mathbb{N}$ such that for every $\varphi \in \mathsf{FOL}[\tau+\mathsf{dp}]$, if φ has quantifier rank r, then φ^l has quantifier rank $f_4(r, l)$.*

The definition of the l-apex-projected sentence φ^l implies the following lemma, which can be seen as a generalization of [79, Lemma 26] that deals with graphs to also "interpret" the vertex-disjoint paths predicates.

Lemma 3. *Let τ be a colored-graph vocabulary, let $l \in \mathbb{N}$, and let $\mathbf{c}$ be a collection of l constant symbols. For every $\varphi \in \mathsf{FOL}[\tau+\mathsf{dp}]$, every τ-structure $\mathfrak{G}$, and every apex-tuple $\mathbf{a}$ of $\mathfrak{G}$ of size l, it holds that $\mathfrak{G} \models \varphi \iff \mathsf{ap}_{\mathbf{c}}(\mathfrak{G}, \mathbf{a}) \models \varphi^l$ (where $\mathbf{c}$ is interpreted as $\mathbf{a}$).*

7 Partial signatures and exchangability

As explained in the end of Subsection 4.4, after proving Lemma 2 our strategy is, given an annotated colored graph to find a way to construct another annotated colored graph that is equivalent to the original one in the sense that they have the same signature. In this section, we describe how, in the presence of a big enough flat railed annulus in the given graph, we can define a way to encode only patterns of "one side" of the annulus. In fact, in Subsection 7.1, we define *stamps* of vertices with respect to some given railed annulus, which will help us to group vertices in a way that encodes their relative position to the cycles of the railed annulus. Then, in Subsection 7.2, in order to facilitate reading, we establish some conventions for boundaried colored graphs in flat railed annuli. Using the notation introduced in these first two subsections, in Subsection 7.3, we show how to reformulate Proposition 2 and how linkages of the given graph (that contains a big enough railed annulus) are "combed" inside boundaried graphs whose boundary vertices are either some particular vertices of the railed annulus, or vertices of appropriate stamps (Lemma 4). Then, to capture the "pattern-behavior" of the considered series of boundaried graphs in the railed annulus, we define the notion of *partial signature* of a graph as a "meta-collection" of patterns of boundaried graphs (as the ones described in Subsection 7.2) for boundary vertices of particular stamps. Finally, in Subsection 7.5, we show that boundaried (annotated) colored graphs with the same partial signature can be "replaced", maintaining the same (global) signature (see Lemma 5).

7.1 Stamps of vertices with respect to annuli

In this subsection, we describe how to attribute *stamps* to the vertices of a given graph G with a *flat* railed annulus $\mathcal{A}$ that encode the relative position of these vertices with respect to the cycles of $\mathcal{A}$. The definition of a flat railed annulus is given in Appendix B, and, in particular, in Subsection B.3. Intuitively, it is the analogue of flat walls but in terms of railed annuli.

Let $r \in \mathbb{N}$. Let G be a graph, let $(\mathcal{A}, \mathfrak{R})$ be a (p,q)-railed annulus flatness pair of G, where $p = 2^r + 1$. For every $\bar{w} \in \{0,1\}^r$, we denote by $C_{\bar{w}}$ the cycle $C_{n_{\bar{w}}}$ of $\mathcal{C}$, where $n_{\bar{w}} = 1 + \sum_{i \in [r]} w_i 2^{r-i+1}$.

Given an $i \in [r]$ and a $\bar{w} \in \{0,1\}^{i-1}$, for every vertex $v \in V(G)$ we define the $\bar{w}$-*stamp* of v as follows:

$$\mathsf{stamp}_{\bar{w}}(v) = \begin{cases} (0,\bullet), & \text{if } v \in V(\mathsf{Influence}_{\mathfrak{R}}(C_{\bar{w}00^{r-i}})), \\ (0,\circ), & \text{if } v \in V(\mathsf{Influence}_{\mathfrak{R}}(C_{\bar{w}11^{r-i}})) \setminus V(\mathsf{Influence}_{\mathfrak{R}}(C_{\bar{w}10^{r-i}})), \\ (1,\bullet), & \text{if } v \in V(\mathsf{Influence}_{\mathfrak{R}}(C_{\bar{w}10^{r-i}})) \setminus V(\mathsf{Influence}_{\mathfrak{R}}(C_{\bar{w}00^{r-i}})), \text{ and} \\ (1,\circ), & \text{if } v \notin V(\mathsf{Influence}_{\mathfrak{R}}(C_{\bar{w}11^{r-i}})). \end{cases}$$

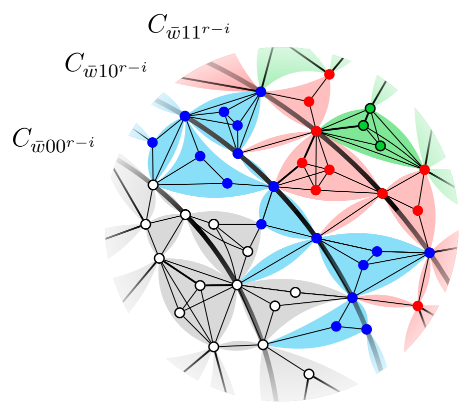

Figure 8: An illustration of some vertices of a graph which contains a flat railed annulus $\mathcal{A} = (\mathcal{C}, \mathcal{P})$, colored with respect to their stamps. The white-colored vertices have $\bar{w}$-stamp equal to $(0,\bullet)$, the blue-colored vertices have $\bar{w}$-stamp equal to $(1,\bullet)$, the red-colored vertices have $\bar{w}$-stamp equal to $(0,\circ)$, and the green-colored vertices have $\bar{w}$-stamp equal to $(1,\circ)$.

Trace of tuples of vertices with respect to annuli. Let $r \in \mathbb{N}$. Let G be a graph, let $(\mathcal{A}, \mathfrak{R})$ be a (p,q)-railed annulus flatness pair, where $p = 2^r + 1$. Given a tuple $(v_1, \ldots, v_r)$ of vertices of G, we define the *trace of* $(v_1, \ldots, v_r)$ *with respect to* $(\mathcal{A}, \mathfrak{R})$, denoted by $\mathsf{trace}_{(\mathcal{A},\mathfrak{R})}(v_1, \ldots, v_r)$, to be the pair $(w_1 \ldots w_r, \omega_1 \ldots \omega_r)$ where $(w_1, \omega_1) = \mathsf{stamp}_\varepsilon(v_1)$ and for every $i \in [2, r]$, we set $(w_i, \omega_i) = \mathsf{stamp}_{w_1 \ldots w_{i-1}}(v_i)$.

7.2 Some conventions for boundaried graphs in flat railed annuli

We proceed to define a series of boundaried graphs in a given railed annulus flatness pair $(\mathcal{A}, \mathfrak{R})$. In fact, we define these boundaried graphs in the *leveling* $\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G)$ of $(\mathcal{A}, \mathfrak{R})$, that is the "planar representation" of $(\mathcal{A}, \mathfrak{R})$, as defined in Subsection B.7. In order $(\mathcal{A}, \mathfrak{R})$ to has this "planar representation" property, it has to be *well-aligned* (see also Subsection B.7).

Let G be a graph and let $(\mathcal{A}, \mathfrak{R})$ be a well-aligned (p, q)-railed annulus flatness pair of G. We consider the graph $\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G)$ and keep in mind that $\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G)$ contains the representation $R_\mathcal{A}$ of $\mathcal{A}$, that is a Δ-embedded (p, q)-railed annulus. Let $R_\mathcal{A} = (\mathcal{C}, \mathcal{P})$ and keep in mind that $|\mathcal{C}| = p$ and p is an odd integer in $\mathbb{N}_{\geq 3}$. Intuitively, the cycle $C_{(p+1)/2}$ is the "middle" cycle of $\mathcal{C}$. We can see each path P_j in $\mathcal{P}$ as being oriented towards the "inner" part of $R_\mathcal{A}$, i.e., starting from an endpoint of $P_{p,j}$ and finishing to an endpoint of $P_{1,j}$. For every $j \in [q]$, we define r_j as the first vertex of P_j that appears in $P_{(p+1)/2,j}$ (recall that $P_{(p+1)/2,j}$ is the intersection of the cycle $C_{(p+1)/2}$ and the rail P_j) while traversing P_j according to this orientation. Given a $t \in [q]$, we define the t-boundaried graph

$$\mathbf{G}^{(t)}_\mathcal{A} = (G', r_1 \ldots, r_t),$$

where G' is the graph $\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G) \setminus V(\mathsf{Right}_{\Delta_{(p+1)/2}}(\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G)))$ (recall that $\Delta_{(p+1)/2}$ is the closed annulus cropped by C_1 and $C_{(p+1)/2}$). We call $r_1, \ldots, r_t$ the *boundary vertices* of $\mathbf{G}^{(t)}_\mathcal{A}$.

Assume now that $\mathcal{A}$ is a well-aligned $(2^r \cdot (p-1)+1, q)$-railed annulus flatness pair of G. We set $\mathcal{C}^{(p)} = C'_1, \ldots, C'_{2^r+1}$, where for every $i \in [2^r + 1]$, $C'_i = C_{1+(p-1)\cdot(i-1)}$. Intuitively, we pick $\mathcal{C}^{(p)}$ in a way that every two consecutive cycles C'_i and C'_{i+1} of $\mathcal{C}^{(p)}$ crop a (p, q)-railed annulus. For every $\bar{w} \in \{0, 1\}^r$, we use $\mathcal{A}_{\bar{w}}$ to denote the (p, q)-annulus that is cropped by the cycles $C'_{n_{\bar{w}}}$ and $C'_{n_{\bar{w}}+1}$ of $\mathcal{C}^{(p)}$, where $n_{\bar{w}} = 1 + \sum_{i\in[r]} w_i 2^{r-i+1}$ (i.e., $\mathcal{A}_{\bar{w}} = \mathcal{A}_{1+(p-1)\cdot(n_{\bar{w}}-1),1+(p-1)\cdot n_{\bar{w}}}$). See Figure 9 for an example.

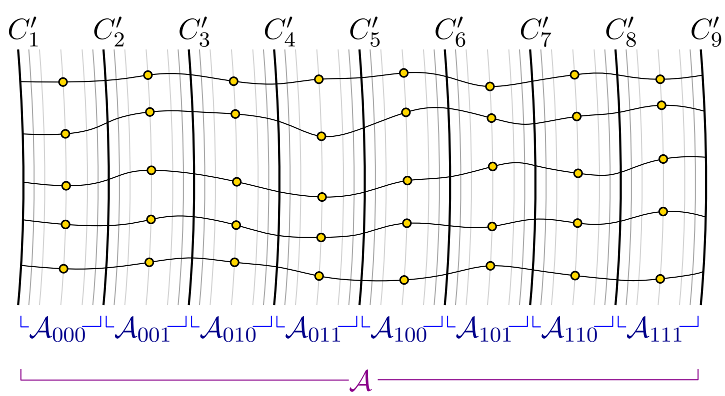

Figure 9: An illustration of the railed annuli $\mathcal{A}_{\bar{w}}$, for every $\bar{w} \in \{0, 1\}^3$, of a railed annulus $\mathcal{A}$. For each $\bar{w} \in \{0, 1\}^3$, the yellow vertices inside $\mathcal{A}_{\bar{w}}$ are the boundary vertices of $\mathbf{G}^{(t)}_{\bar{w}}$.

Given a $\bar{w} \in \{0, 1\}^r$, we define $\mathbf{G}^{(t)}_{\bar{w}}$ to be the graph $\mathbf{G}^{(t)}_{\mathcal{A}_{\bar{w}}}$. In the rest of the paper, given an $r \in \mathbb{N}$ and a $\bar{w} \in \{0, 1\}^r$, we denote by $\mathbf{G}_{\bar{w}}$ the ℓ-boundaried graph $\mathbf{G}^{(\ell)}_{\bar{w}}$, where $\ell := f_2(\binom{r}{2}) + 1$ (where f_2 is the first of the two functions of Proposition 2), we use $G_{\bar{w}}$ to denote its underlying graph, and $v_{r+1}, \ldots, v_{r+\ell}$ to denote its boundary vertices. Also, we denote by $G^{\mathsf{out}}_{\bar{w}}$ the graph $G \setminus (V(G_{\bar{w}}) \setminus \{v_{r+1}, \ldots, v_{r+\ell}\})$ and by $\mathbf{G}^{\mathsf{out}}_{\bar{w}}$ the ℓ-boundaried graph $(G^{\mathsf{out}}_{\bar{w}}, v_{r+1}, \ldots, v_{r+\ell})$.

7.3 Combing linkages in levelings of flat annuli

We now formulate Proposition 2 in terms of pairings of the graphs $\mathbf{G}_{\bar{w}}$ and $\mathbf{G}^{\mathsf{out}}_{\bar{w}}$. That is, for every linkage with terminals $v_1, \ldots, v_r$, we prove that another equivalent linkage can be found, being combed through the extra boundary vertices of $\mathbf{G}_{\bar{w}}$ and $\mathbf{G}^{\mathsf{out}}_{\bar{w}}$ (see Lemma 4). Before presenting Lemma 4 and its proof, we introduce some additional notation.

Let G be a graph and let $(\mathcal{A}, \mathfrak{R})$ be a well-aligned $(2^r \cdot (p-1)+1, q)$-railed annulus flatness pair of G, for some odd $p \in \mathbb{N}_{\geq 3}$, and some $q \in \mathbb{N}_{\geq 3}$. Let $v_1, \ldots, v_r \in V(G)$ and let $(\bar{w}, \omega_1 \ldots \omega_r)$ be the trace of $(v_1, \ldots, v_r)$. We use $\bar{v}^{\bullet}$ to denote the tuple $(v^{\bullet}_1, \ldots, v^{\bullet}_r)$, where for every $i \in [r]$, $v^{\bullet}_i = v_i$, if $\omega_i = \bullet$, and $v^{\bullet}_i = \text{\textvisiblespace}$, if $\omega_i = \circ$. Also, we use $\bar{v}^{\circ}$ to denote the tuple $(v^{\circ}_1, \ldots, v^{\circ}_r)$, where for every $i \in [r]$, $v^{\circ}_i = v_i$, if $\omega_i = \circ$, and $v^{\circ}_i = \text{\textvisiblespace}$, if $\omega_i = \bullet$. We use $(\mathbf{G}_{\bar{w}}, \bar{v}^{\bullet})$ to denote $(G_{\bar{w}}, \bar{v}^{\bullet}, v_{r+1}, \ldots, v_{r+\ell})$ and $(\mathbf{G}^{\mathsf{out}}_{\bar{w}}, \bar{v}^{\circ})$ to denote $(G^{\mathsf{out}}_{\bar{w}}, \bar{v}^{\circ}, v_{r+1}, \ldots, v_{r+\ell})$.

Lemma 4. *There is a function $f_5 : \mathbb{N} \to \mathbb{N}$ such that for every $r \in \mathbb{N}$, if G is a graph, $(\mathcal{A}, \mathfrak{R})$ is a well-aligned $(f_5(r), \ell)$-railed annulus flatness pair of G, and $(L, v_1, \ldots, v_r) \in \mathsf{Pairings}(G, v_1, \ldots, v_r)$, where $v_1, \ldots, v_r \in V(G)$, then there exists a $(\tilde{L}, v_1, \ldots, v_r) \in \mathsf{Pairings}(G, v_1, \ldots, v_r)$ such that $L \equiv \tilde{L}$ and $(\tilde{L}, v_1, \ldots, v_r) \in \mathsf{Pairings}(\mathbf{G}_{\bar{w}}, \bar{v}^{\bullet}) \oplus \mathsf{Pairings}(\mathbf{G}^{\mathsf{out}}_{\bar{w}}, \bar{v}^{\circ})$, where $(\bar{w}, \bar{\omega}) = \mathsf{trace}_{(\mathcal{A},\mathfrak{R})}(v_1, \ldots, v_r)$.*

Proof. We set $f_5(r) = 2^r \cdot (p-1) + 1$, where $p = f_3(\binom{r}{2}) + 5$. Recall that $\mathbf{G}_{\bar{w}}$ and $\mathbf{G}^{\mathsf{out}}_{\bar{w}}$ are $(r+\ell)$-boundaried graphs where $\ell = f_2(\binom{r}{2}) + 1$ and whose boundary vertices are $v_1, \ldots, v_{r+\ell}$. Let $(L, v_1, \ldots, v_r) \in \mathsf{Pairings}(G, v_1, \ldots, v_r)$.

We set $(\bar{w}, \omega_1 \ldots \omega_r) = \mathsf{trace}_{(\mathcal{A},\mathfrak{R})}(v_1, \ldots, v_r)$ and we consider the (p, q)-railed annulus $\mathcal{A}_{\bar{w}}$ of G. Observe that $v_1, \ldots, v_r \notin \mathsf{Influence}_{\mathfrak{R}}(\mathcal{A}_{\bar{w}})$ and therefore, by Lemma 12, we have that $\mathsf{Leveling}_{(\mathcal{A}_{\bar{w}},\mathfrak{R})}(G)$ contains a linkage $\hat{L}$ that is equivalent to L and it is $\mathsf{ann}(R_{\mathcal{A}_{\bar{w}}})$-avoiding. We set $\Delta = \mathsf{ann}(R_{\mathcal{A}_{\bar{w}}})$ and we note that $R_{\mathcal{A}_{\bar{w}}}$ is a Δ-embedded (p, q)-railed annulus of $\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G)$. By applying Proposition 2 for the (p, q)-railed annulus $R_{\mathcal{A}_{\bar{w}}}$, for $s = 3$, and for $I = [\ell]$, we have that $\mathsf{Leveling}_{(\mathcal{A}_{\bar{w}},\mathfrak{R})}(G)$ contains a linkage $\tilde{L}$ that is equivalent to $\hat{L}$ (and, therefore, equivalent to L) and is $(3, I)$-confined in $R_{\mathcal{A}_{\bar{w}}}$.

We set $\tilde{L}^{\bullet} := \tilde{L} \cap G_{\bar{w}}$ and $\tilde{L}^{\circ} := \tilde{L} \cap G^{\mathsf{out}}_{\bar{w}}$. Observe that $(\tilde{L}^{\bullet}, \bar{v}^{\bullet}, v_{r+1}, \ldots, v_{r+\ell}) \in \mathsf{Pairings}(\mathbf{G}_{\bar{w}}, \bar{v}^{\bullet})$ and $(\tilde{L}^{\circ}, \bar{v}^{\circ}, v_{r+1}, \ldots, v_{r+\ell}) \in \mathsf{Pairings}(\mathbf{G}^{\mathsf{out}}_{\bar{w}}, \bar{v}^{\circ})$. Also, the pairings $(\tilde{L}^{\bullet}, \bar{v}^{\bullet}, v_{r+1}, \ldots, v_{r+\ell})$ and $(\tilde{L}^{\circ}, \bar{v}^{\circ}, v_{r+1}, \ldots, v_{r+\ell})$ are ℓ-compatible and

$$(\tilde{L}, v_1, \ldots, v_r) = (\tilde{L}^{\bullet}, \bar{v}^{\bullet}, v_{r+1}, \ldots, v_{r+\ell}) \oplus (\tilde{L}^{\circ}, \bar{v}^{\circ}, v_{r+1}, \ldots, v_{r+\ell}).$$

Since $(\tilde{L}^{\bullet}, \bar{v}^{\bullet}, v_{r+1}, \ldots, v_{r+\ell}) \in \mathsf{Pairings}(\mathbf{G}_{\bar{w}}, \bar{v}^{\bullet})$ and $(\tilde{L}^{\circ}, \bar{v}^{\circ}, v_{r+1}, \ldots, v_{r+\ell}) \in \mathsf{Pairings}(\mathbf{G}^{\mathsf{out}}_{\bar{w}}, \bar{v}^{\circ})$, we have that $(\tilde{L}, v_1, \ldots, v_r) \in \mathsf{Pairings}(\mathbf{G}_{\bar{w}}, \bar{v}^{\bullet}) \oplus \mathsf{Pairings}(\mathbf{G}^{\mathsf{out}}_{\bar{w}}, \bar{v}^{\circ})$. ☐

7.4 Partial signatures of tuples of vertices

We now present the definition of partial signatures. It is a recursive definition, that in the base case captures the pattern of a boundaried graph and for every recursive step, asks for all possible partial signatures that can be obtained after fixing another (boundary) vertex or its absence. Intuitively, the symbol "␣" expresses the absence of a vertex for this entry, or, in other words, that this vertex should be picked inside some boundaried graph that should be glued to our considered boundaried graph. In this sense, partial signatures express "partial" patterns (see Observation 10).

Let τ be a colored-graph vocabulary and let $\mathbf{c} = \{\mathsf{c}_1, \ldots, \mathsf{c}_l\}$ be a collection of l constant symbols. Let $r \in \mathbb{N}$. Recall that $\ell = f_2(\binom{r}{2}) + 1$, where f_2 is the second function of Proposition 2. Let $\mathfrak{G}$ be a τ-structure, let $R_1, \ldots, R_r \subseteq V(G)$, and let $(\mathcal{A}, \mathfrak{R})$ be a well-aligned $(2^r + 1, \ell)$-railed annulus flatness pair of G. For every $i \in [r]$ and every $\bar{w} \in \{0,1\}^i$, we define

$$R_i^{\bar{w}} = \{v \in R_i \mid \mathsf{stamp}_{w_1 \ldots w_{i-1}}(v) = (w_i, \bullet)\}$$

and

$$V^{\bar{w}} = \{v \in V(G) \mid \mathsf{stamp}_{w_1 \ldots w_{i-1}}(v) = (w_i, \bullet)\}.$$

Partial signature. Let $d \in \mathbb{N}$ and $r \in [d-1]$. Given a $\bar{w} \in \{0,1\}^d$ and $v_1, \ldots, v_d$ such that for every $i \in [d]$, $v_i \in V^{w_1 \ldots w_i} \cup \{\text{\textvisiblespace}\}$ we define

$$\mathsf{partial\text{-}sig}_r^0(\mathfrak{G}, \bar{R}, \bar{w}, v_1, \ldots, v_d) = \mathsf{pattern}(\mathbf{G}_{\bar{w}}, \text{\textvisiblespace}^l, v_1, \ldots, v_d).$$

Also, for each $i \in [d-r]$, every $\bar{w} \in \{0,1\}^{d-i}$ and every $v_1, \ldots, v_{d-i} \in V(G) \cup \{\text{\textvisiblespace}\}$ such that for every $j \in [d-i]$, $v_j \in V^{w_1 \ldots w_j} \cup \{\text{\textvisiblespace}\}$, we define

$$\begin{aligned}
\mathsf{partial\text{-}sig}_r^i(\mathfrak{G}, \bar{R}, \bar{w}, v_1, \ldots, v_{d-i}) \quad = \quad & \{\mathsf{partial\text{-}sig}_r^{i-1}(\mathfrak{G}, \bar{R}, \bar{w}0, v_1, \ldots, v_{d-i}, v) \mid v \in V^{\bar{w}0}\} \\
& \cup \{\mathsf{partial\text{-}sig}_r^{i-1}(\mathfrak{G}, \bar{R}, \bar{w}1, v_1, \ldots, v_{d-i}, v) \mid v \in V^{\bar{w}1}\} \\
& \cup \{\mathsf{partial\text{-}sig}_r^{i-1}(\mathfrak{G}, \bar{R}, \bar{w}0, v_1, \ldots, v_{d-i}, \text{\textvisiblespace})\} \\
& \cup \{\mathsf{partial\text{-}sig}_r^{i-1}(\mathfrak{G}, \bar{R}, \bar{w}1, v_1, \ldots, v_{d-i}, \text{\textvisiblespace})\}
\end{aligned}$$

Also, for each $i \in [d-r+1, d-1]$, every $\bar{w} \in \{0,1\}^{d-i}$ and every $v_1, \ldots, v_{d-i} \in V(G) \cup \{\text{\textvisiblespace}\}$ such that for every $j \in [d-i]$, $v_j \in V^{w_1 \ldots w_j} \cup \{\text{\textvisiblespace}\}$, we define

$$\begin{aligned}
\mathsf{partial\text{-}sig}_r^i(\mathfrak{G}, \bar{R}, \bar{w}, v_1, \ldots, v_{d-i}) \quad = \quad & \{\mathsf{partial\text{-}sig}_r^{i-1}(\mathfrak{G}, \bar{R}, \bar{w}0, v_1, \ldots, v_{d-i}, v) \mid v \in R_{d-i+1}^{\bar{w}0}\} \\
& \cup \{\mathsf{partial\text{-}sig}_r^{i-1}(\mathfrak{G}, \bar{R}, \bar{w}1, v_1, \ldots, v_{d-i}, v) \mid v \in R_{d-i+1}^{\bar{w}1}\} \\
& \cup \{\mathsf{partial\text{-}sig}_r^{i-1}(\mathfrak{G}, \bar{R}, \bar{w}0, v_1, \ldots, v_{d-i}, \text{\textvisiblespace})\} \\
& \cup \{\mathsf{partial\text{-}sig}_r^{i-1}(\mathfrak{G}, \bar{R}, \bar{w}1, v_1, \ldots, v_{d-i}, \text{\textvisiblespace})\}
\end{aligned}$$

Finally, we define

$$\begin{aligned}
\mathsf{partial\text{-}sig}_r^d(\mathfrak{G}, \bar{R}) \quad = \quad & \{\mathsf{partial\text{-}sig}_r^{d-1}(\mathfrak{G}, \bar{R}, 0, v) \mid v \in R_1^0\} \\
& \cup \{\mathsf{partial\text{-}sig}_r^{d-1}(\mathfrak{G}, \bar{R}, 1, v) \mid v \in R_1^1\} \\
& \cup \{\mathsf{partial\text{-}sig}_r^{d-1}(\mathfrak{G}, \bar{R}, 0, \text{\textvisiblespace})\} \\
& \cup \{\mathsf{partial\text{-}sig}_r^{d-1}(\mathfrak{G}, \bar{R}, 1, \text{\textvisiblespace})\}
\end{aligned}$$

We use $\mathbb{P}(A)$ to denote the powerset of a set A and for every $d \in \mathbb{N}$, we use $\mathbb{P}^d(A)$ to denote the set $\underbrace{\mathbb{P}(\mathbb{P}(\cdots \mathbb{P}}_{d\text{-times}}(A)))$. We observe the following.

Observation 10. *Let $d \in \mathbb{N}$ and $r \in [d-1]$. For every $i \in [0, d-1]$, every $\bar{w} \in \{0,1\}^{d-i}$, and every $\bar{v} \in (V(G) \cup \{\text{\textvisiblespace}\})^{r-i}$, where for every $j \in [d-i]$, $v_j \in V^{w_1 \ldots w_j} \cup \{\text{\textvisiblespace}\}$, it holds that* $\mathsf{partial\text{-}sig}_r^i(\mathfrak{G}, \bar{R}, \bar{w}, \bar{v}) \in \mathbb{P}^i(\mathcal{G}_{\mathsf{pat}}^{(d+\ell,h,l)})$ *and* $\mathsf{partial\text{-}sig}_r^d(\mathfrak{G}, \bar{R}) \in \mathbb{P}^d(\mathcal{G}_{\mathsf{pat}}^{(d+\ell,h,l)})$.

Following the above definition, we also define (global) signatures where some vertices, up to some index, are asked to belong to the corresponding R_i while the rest of them can belong in $V(G)$. More formally, given some $d \in \mathbb{N}$ and some $r \in [0, d]$, for every $(v_1, \ldots, v_d) \in (V(G) \cup \{\_\})^d$, we define $\mathsf{sig}^0_r(\mathfrak{G}, \bar{R}, v_1, \ldots, v_d)$ to be the atomic type of $(v_1, \ldots, v_d)$. Also, for each $i \in [d-r]$, and every $v_1, \ldots, v_{d-i} \in V(G) \cup \{\_\}$, we define

$$\mathsf{sig}^i_r(\mathfrak{G}, \bar{R}, v_1, \ldots, v_{d-i}) = \big\{\mathsf{sig}^{i-1}_r(\mathfrak{G}, \bar{R}, v_1, \ldots, v_{d-i}, u) \mid u \in V(G) \cup \{\_\}\big\}.$$

Also, for each $i \in [d-r+1, d-1]$ and every $v_1, \ldots, v_{d-i} \in V(V(G) \cup \{\_\}$, we define

$$\mathsf{sig}^i_r(\mathfrak{G}, \bar{R}, v_1, \ldots, v_{d-i}) = \big\{\mathsf{sig}^{i-1}_r(\mathfrak{G}, \bar{R}, v_1, \ldots, v_{d-i}, u) \mid u \in V(G) \cup \{\_\}\big\},$$

while for each $i \in [d-r]$ and every $v_1, \ldots, v_{d-i} \in V(V(G) \cup \{\_\}$, we define

$$\mathsf{sig}^i_r(\mathfrak{G}, \bar{R}, v_1, \ldots, v_{d-i}) = \big\{\mathsf{sig}^{i-1}_r(\mathfrak{G}, \bar{R}, v_1, \ldots, v_{d-i}, u) \mid u \in R_{d-i+1} \cup \{\_\}\big\}.$$

Finally, we define

$$\mathsf{sig}^d_r(\mathfrak{G}, \bar{R}) = \big\{\mathsf{sig}^{d-1}_r(\mathfrak{G}, \bar{R}, v) \mid v \in R_1 \cup \{\_\}\big\}.$$

7.5 Exchangeability of graphs with the same partial signature

The goal of this section is to present Lemma 5 and its proof. This result states that in the presence of a railed annulus flatness pair in side an (annotated) colored graph $(\mathfrak{D}, R^\diamond_1, \ldots, R^\diamond_r)$, two (annotated) colored graphs $(\mathfrak{G}, R_1, \ldots, R_r)$, $(\mathfrak{G}', R'_1, \ldots, R'_r)$ that yield the same partial signature when "glued in the inner part" of $(\mathfrak{D}, R^\diamond_1, \ldots, R^\diamond_r)$, satisfy the same formulas when we additionally glue another (annotated) colored graph $(\mathfrak{F}, R^\star_1, \ldots, R^\star_r)$ in the outer part of $(\mathfrak{D}, R^\diamond_1, \ldots, R^\diamond_r)$. We now formalize the idea of "gluing".

Compatible colored graphs. Let $\mathfrak{G} = (G, X_1, \ldots, X_h), \mathfrak{G}' = (G', X'_1, \ldots, X'_h)$ be two colored graphs and let a partial function $\eta : V(G) \to V(G')$. We say that $\mathfrak{G}$ and $\mathfrak{G}'$ are *η-compatible* if for every $i \in [h]$ and every $v \in X_i$, $v \in X_i \iff \eta(v) \in X'_i$.

Gluing colored graphs. Let $\mathfrak{G}, \mathfrak{D}$ be two colored graphs and let a partial function $\eta : V(G) \to V(H)$ such that $\mathfrak{G}$ and $\mathfrak{D}$ are η-compatible. We denote by $\mathfrak{G} \oplus_\eta \mathfrak{D}$ the colored graph obtained from the disjoint union of $\mathfrak{G}$ and $\mathfrak{D}$ after identifying vertices $v \in V(G)$ and $u \in V(H)$ if $\eta(v) = u$.

Inner- and outer-compatibility functions. Let two colored graphs $\mathfrak{G}, \mathfrak{H}$, let $\mathbf{a}$ be an apex-tuple of $\mathfrak{G}$, and let $(\mathcal{A}, \mathfrak{R})$ be a railed annulus flatness pair of $\mathfrak{G} \setminus V(\mathbf{a})$. Given that $\mathfrak{R} = (X_1, Y_1, X_2, Y_2, Z_1, Z_2, \Gamma, \sigma, \pi)$, we call a partial function $\eta : V(X_1 \cap Y_1) \cup V(\mathbf{a}) \to V(H)$ (resp. $\xi : V(X_2 \cap Y_2) \cup V(\mathbf{a}) \to V(H)$) such that $\mathfrak{G}$ and $\mathfrak{H}$ are η-compatible (resp. ξ-compatible) an *inner-compatibility (resp. outer-compatibility) function of $\mathfrak{G}$ and $\mathfrak{H}$.*

Lemma 5. *Let τ be a colored-graph vocabulary and let $r, l \in \mathbb{N}$ and let $d = f_4(r, l)$. Let $(\mathfrak{D}, \bar{R}^\diamond)$ be a colored graph, let $\mathbf{a}$ be an apex-tuple of $\mathfrak{D}$ of size l, and let $(\mathcal{A}, \mathfrak{R})$ be a well-aligned $(f_5(r), \ell)$-railed annulus flatness pair of $\mathfrak{D} \setminus V(\mathbf{a})$. Also, let $(\mathfrak{G}, \bar{R}), (\mathfrak{G}', \bar{R}')$ be two colored graphs and let two inner-compatibility functions η, η' of $(\mathfrak{D}, \bar{R}^\diamond)$ and $(\mathfrak{G}, \bar{R})$ (resp. $(\mathfrak{G}', \bar{R}')$). If*

$$\text{partial-}\mathsf{sig}^d_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G}, \bar{R}) \oplus_\eta (\mathfrak{D}, \bar{R}^\diamond), \mathbf{a})) = \text{partial-}\mathsf{sig}^d_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G}', \bar{R}') \oplus_{\eta'} (\mathfrak{D}, \bar{R}^\diamond), \mathbf{a})),$$

then for every $(\tau \cup \{\mathsf{R}_1, \ldots, \mathsf{R}_r\})$-structure $(\mathfrak{F}, \bar{R}^\star)$, every outer-compatibility function ξ of $(\mathfrak{D}, \bar{R}^\diamond)$ and $(\mathfrak{F}, \bar{R}^\star)$, it holds that

$$\mathsf{sig}^r\big((\mathfrak{G}, \bar{R}) \oplus_\eta (\mathfrak{D}, \bar{R}^\diamond) \oplus_\xi (\mathfrak{F}, \bar{R}^\star)\big) = \mathsf{sig}^r\big((\mathfrak{G}', \bar{R}') \oplus_{\eta'} (\mathfrak{D}, \bar{R}^\diamond) \oplus_\xi (\mathfrak{F}, \bar{R}^\star)\big).$$

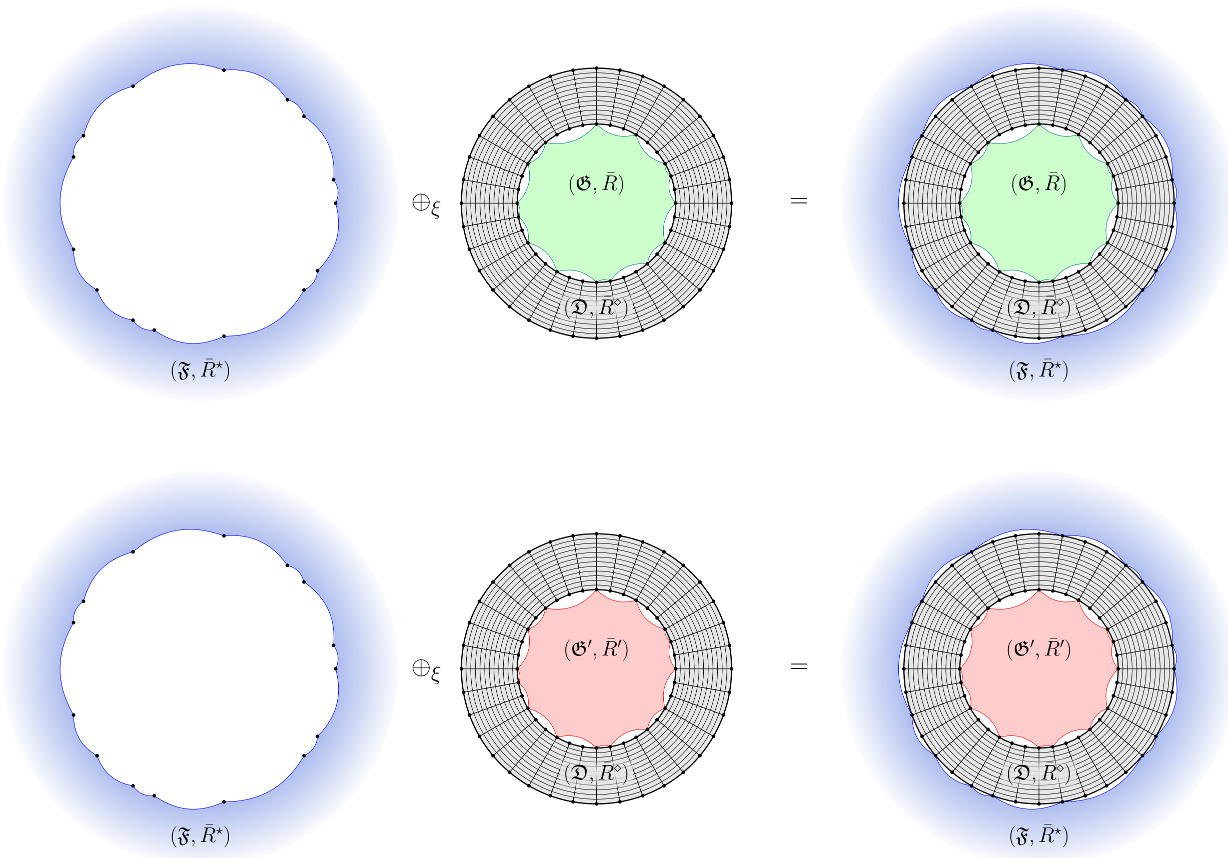

Figure 10: A simplified visualization of the statement of Lemma 5. If the upper middle and the lower middle colored graphs have the same partial signature, then for every colored graph $(\mathfrak{F}, \bar{R}^\star)$ (on the left) glued to them, the resulting colored graphs (the upper right and the lower right) have the same (global) signature.

See Figure 10 for a simplified visualization of the statement of Lemma 5.

Proof. Let $\mathfrak{F}$ be a colored graph, let $\bar{R}^\star = (R_1^\star, \ldots, R_r^\star)$, where $R_1^\star, \ldots, R_r^\star \subseteq V(\mathfrak{F})$, and let ξ be an outer-compatibility function of $(\mathfrak{D}, \bar{R}^\diamond)$ and $(\mathfrak{F}, \bar{R}^\star)$.

Given that $\mathfrak{R} = (X_1, Y_1, X_2, Y_2, Z_1, Z_2, \Gamma, \sigma, \pi)$, we set $\tilde{\mathfrak{R}} = (\tilde{X}_1, \tilde{Y}_1, \tilde{X}_2, \tilde{Y}_2, Z_1, Z_2, \Gamma, \sigma, \pi)$ and $\tilde{\mathfrak{R}}' = (\tilde{X}_1', \tilde{Y}_1, \tilde{X}_2', \tilde{Y}_2, Z_1, Z_2, \Gamma, \sigma, \pi)$, where

- $\tilde{Y}_1$ (resp. $\tilde{Y}_2$) is the vertex set obtained from the union of Y_1 (resp. Y_2) and $V(\mathfrak{G})$ after identifying the vertices in $(X_1 \cap Y_1) \cup V(\mathbf{a})$ with their images via η,
- $\tilde{Y}_1'$ (resp. $\tilde{Y}_2'$) is the vertex set obtained from the union of Y_1 (resp. Y_2) and $V(\mathfrak{G}')$ after identifying the vertices in $(X_1 \cap Y_1) \cup V(\mathbf{a})$ with their images via η', and
- $\tilde{X}_1$ (resp. $\tilde{X}_2$) is the vertex set obtained from the union of X_1 (resp. X_2) and $V(\mathfrak{F})$ after identifying the vertices in $(X_2 \cap Y_2) \cup V(\mathbf{a})$ with their images via ξ.

Observe that $(\mathcal{A}, \tilde{\mathfrak{R}})$ is a railed annulus flatness pair of $(\mathfrak{G} \oplus_\eta \mathfrak{D} \oplus_\xi \mathfrak{F}) \setminus V(\mathbf{a})$ and $(\mathcal{A}, \tilde{\mathfrak{R}}')$ is a railed annulus flatness pair of $(\mathfrak{G}' \oplus_{\eta'} \mathfrak{D} \oplus_\xi \mathfrak{F}) \setminus V(\mathbf{a})$.

We now consider the $\tau^{\langle \mathbf{c} \rangle}$-structures $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G} \oplus_\eta \mathfrak{D} \oplus_\xi \mathfrak{F}, \mathbf{a})$ and $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}' \oplus_{\eta'} \mathfrak{D} \oplus_\xi \mathfrak{F}, \mathbf{a})$. Note that $(\mathcal{A}, \tilde{\mathfrak{R}})$ is also a railed annulus flatness pair of $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G} \oplus_\eta \mathfrak{D} \oplus_\xi \mathfrak{F}, \mathbf{a}) \setminus V(\mathbf{a})$. Also, since in

$\mathsf{ap}_{\mathbf{c}}(\mathfrak{G} \oplus_\eta \mathfrak{D} \oplus_\xi \mathfrak{F}, \mathbf{a})$ there are no edges between $V(\mathbf{a})$ and $V(\mathsf{ap}_{\mathbf{c}}(\mathfrak{G} \oplus_\eta \mathfrak{D} \oplus_\xi \mathfrak{F}, \mathbf{a}) \setminus V(\mathbf{a}))$, we can update $\mathfrak{R}$ by adding $V(\mathbf{a})$ to $\tilde{X}_2$ and observe that after this modification of $\tilde{\mathfrak{R}}$, $(\mathcal{A}, \tilde{\mathfrak{R}})$ is a railed annulus flatness pair of $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G} \oplus_\eta \mathfrak{D} \oplus_\xi \mathfrak{F}, \mathbf{a})$. For the same reasons, we can assume that $(\mathcal{A}, \tilde{\mathfrak{R}}')$ is a railed annulus flatness pair of $\mathsf{ap}_{\mathbf{c}}(\mathfrak{G}' \oplus_{\eta'} \mathfrak{D} \oplus_\xi \mathfrak{F}, \mathbf{a})$. We set

- $(\mathfrak{Z}, \tilde{R}_1, \ldots, \tilde{R}_r) := (\mathfrak{G}, \bar{R}) \oplus_\eta (\mathfrak{D}, \bar{R}^\diamond) \oplus_\xi (\mathfrak{F}, \bar{R}^\star)$,
- $(\mathfrak{Z}', \tilde{R}'_1, \ldots, \tilde{R}'_r) := (\mathfrak{G}', \bar{R}') \oplus_{\eta'} (\mathfrak{D}, \bar{R}^\diamond) \oplus_\xi (\mathfrak{F}, \bar{R}^\star)$,
- $(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}) := \mathsf{ap}_{\mathbf{c}}(\mathfrak{Z}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a})$, and
- $(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a}) := \mathsf{ap}_{\mathbf{c}}(\mathfrak{Z}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a})$.

Our goal is to prove that $\mathsf{sig}^r(\mathfrak{Z}, \tilde{R}_1, \ldots, \tilde{R}_r) = \mathsf{sig}^r(\mathfrak{Z}', \tilde{R}'_1, \ldots, \tilde{R}'_r)$. To achieve this, it will suffice to show that $\mathsf{sig}^d_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}) = \mathsf{sig}^d_r(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a})$. Let us first prove that

$$\mathsf{sig}^d_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}) = \mathsf{sig}^d_r(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a}) \text{ implies } \mathsf{sig}^r(\mathfrak{Z}, \tilde{R}_1, \ldots, \tilde{R}_r) = \mathsf{sig}^r(\mathfrak{Z}', \tilde{R}'_1, \ldots, \tilde{R}'_r).$$

For this, we will prove that for every quantifier-free formula $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ of $\mathsf{FOL}[\tau + \mathsf{dp}]$ on r free variables, the following holds: if $v_1, \ldots, v_r \in V(\mathfrak{Z})$ and $v'_1, \ldots, v'_r \in V(\mathfrak{Z}')$ such that for every $i \in [r-1]$,

$$\mathsf{sig}^i(\mathfrak{Z}, \tilde{R}_1, \ldots, \tilde{R}_r, v_1, \ldots, v_{r-i}) = \mathsf{sig}^i(\mathfrak{Z}', \tilde{R}'_1, \ldots, \tilde{R}'_r, v'_1, \ldots, v'_{r-i}),$$

then it holds that $(\mathfrak{Z}, v_1, \ldots, v_r) \models \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r) \iff (\mathfrak{Z}, v_1, \ldots, v_r) \models \psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$.

Let $\psi(\mathsf{x}_1, \ldots, \mathsf{x}_r)$ be such a formula and let $v_1, \ldots, v_r \in V(\mathfrak{Z})$ and $v'_1, \ldots, v'_r \in V(\mathfrak{Z}')$ such that $\mathsf{sig}^i(\mathfrak{Z}, \tilde{R}_1, \ldots, \tilde{R}_r, v_1, \ldots, v_{r-i}) = \mathsf{sig}^i(\mathfrak{Z}', \tilde{R}'_1, \ldots, \tilde{R}'_r, v'_1, \ldots, v'_{r-i})$, for every $i \in [r-1]$. Assuming that $\mathsf{sig}^d_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}) = \mathsf{sig}^d_r(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a})$, it holds that

$$\mathsf{sig}^{d-r}_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}, v_1, \ldots, v_r) = \mathsf{sig}^{d-r}_r(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a}, v'_1, \ldots, v'_r).$$

Therefore, by Lemma 2, we have $(\mathfrak{H}, \mathbf{a}, v_1, \ldots, v_r) \models \psi^l \iff (\mathfrak{H}', \mathbf{a}, v'_1, \ldots, v'_r) \models \psi^l$, where ψ^l is the apex-projection of ψ (see Subsection 6.2). Also, by Lemma 3, $(\mathfrak{Z}, v_1, \ldots, v_r) \models \psi \iff (\mathfrak{H}, \mathbf{a}, v_1, \ldots, v_r) \models \psi^l$ and $(\mathfrak{Z}, v_1, \ldots, v_r) \models \psi \iff (\mathfrak{H}', \mathbf{a}, v'_1, \ldots, v'_r) \models \psi^l$. Combining these last three logical equivalences, we get $(\mathfrak{Z}, v_1, \ldots, v_r) \models \psi \iff (\mathfrak{Z}, v_1, \ldots, v_r) \models \psi$.

We devote the rest of the proof to show that $\mathsf{sig}^d_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}) = \mathsf{sig}^d_r(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a})$. Let λ be an assignment of $(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r)$ to a rooted tree (T, t_0) and let λ' be an assignment of $(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r)$ to a rooted tree (T', t'_0).

To prove that $\mathsf{sig}^d_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}) = \mathsf{sig}^d_r(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a})$, it suffices to prove that for every $i \in [0, d]$ and every $\gamma \in \mathbb{P}^{d-i}(\mathcal{G}^{(t,h)}_{\mathsf{pat}})$, the two following statements hold:

(i) For every $(t_0, \ldots, t_i) \in \mathsf{Paths}(T)$ there exist $(t'_0, \ldots, t'_i) \in \mathsf{Paths}(T')$ such that

$$\mathsf{sig}^{d-i}_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}, \lambda(t_1), \ldots, \lambda(t_i)) = \mathsf{sig}^t(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a}, \lambda'(t'_1), \ldots, \lambda'(t'_i)).$$

(ii) for every $(t'_0, \ldots, t'_i) \in \mathsf{Paths}(T')$ there exist $(t_0, \ldots, t_i) \in \mathsf{Paths}(T)$ such that

$$\mathsf{sig}^{d-i}_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}, \lambda(t_1), \ldots, \lambda(t_i)) = \mathsf{sig}^t(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a}, \lambda'(t'_1), \ldots, \lambda'(t'_i)).$$

We will show only a proof for (i), since the proof of (ii) will be totally symmetric to the one of (i). In fact, we will prove the following statement, which is equivalent to (i).

Claim 3. *For every $i \in [0,d]$ and every $\beta \in \mathcal{G}^{(d,h)}_{\mathsf{pat}}$, it holds that: For every $(t_0,\dots,t_i) \in \mathsf{Paths}(T)$ there is a $(t'_0,\dots,t'_i) \in \mathsf{Paths}(T')$ such that for every $t_d \in L(T_{t_i})$, where $(t_0,\dots,t_d) \in \mathsf{Paths}(T)$, there is a $t'_d \in L(T'_{t'_i})$, where $(t'_0,\dots,t'_d) \in \mathsf{Paths}(T')$, such that*

$$\mathsf{sig}^0_r(\mathfrak{H}, \tilde{R}_1,\dots,\tilde{R}_r,\mathbf{a},\lambda(t_1),\dots,\lambda(t_d)) = \mathsf{sig}^0_r(\mathfrak{H}', \tilde{R}'_1,\dots,\tilde{R}'_r,\mathbf{a},\lambda'(t'_1),\dots,\lambda'(t'_d)).$$

Proof of Claim 3. Let $i \in [0,d]$. Let $t_0,\dots,t_i \in \mathsf{Paths}(T)$, let $(v_1,\dots,v_i) = (\lambda(t_1),\dots,\lambda(t_i))$. We will prove that there is a $(t'_1,\dots,t'_i) \in \mathsf{Paths}(T')$ such that for every $(t_{i+1},\dots,t_d) \in \mathsf{Paths}(T)$, where $t_{i+1} \in \mathsf{children}_T(t_i)$, there is a $(t'_{i+1},\dots,t'_d) \in \mathsf{Paths}(T')$, where $t'_{i+1} \in \mathsf{children}_{T'}(t'_i)$, such that

$$\mathsf{sig}^0_r(\mathfrak{H}, \tilde{R}_1,\dots,\tilde{R}_r,\mathbf{a},\lambda(t_1),\dots,\lambda(t_d)) = \mathsf{sig}^0_r(\mathfrak{H}', \tilde{R}'_1,\dots,\tilde{R}'_r,\mathbf{a},\lambda'(t'_1),\dots,\lambda'(t'_d)).$$

For every $j \in [i]$, let $(w_j,\omega_j) = \mathsf{stamp}_{w_1\dots w_{j-1}}(v_j)$. For every $j \in [i]$, we set

$$v^\bullet_j = \begin{cases} v_j, & \text{if } \omega_j = \bullet \\ \llcorner\!\lrcorner, & \text{if } \omega_j = \circ. \end{cases}$$

Since $\mathsf{partial\text{-}sig}^d_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G},\bar{R}) \oplus_\eta (\mathfrak{D},\bar{R}^\diamond),\mathbf{a})) = \mathsf{partial\text{-}sig}^d_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G}',\bar{R}') \oplus_{\eta'} (\mathfrak{D},\bar{R}^\diamond),\mathbf{a}))$, there is a $t'_i \in D_i(T)$, such that if $t'_0,\dots,t'_i \in \mathsf{Paths}(T')$ and $(u_1,\dots,u_i) = (\lambda'(t'_1),\dots,\lambda'(t'_i))$, then it holds that $\mathsf{trace}(v_1,\dots,v_i) = \mathsf{trace}(u_1,\dots,u_i)$ and

$$\begin{gathered}\mathsf{partial\text{-}sig}^{d-i}_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G},\bar{R}) \oplus_\eta (\mathfrak{D},\bar{R}^\diamond),\mathbf{a}),\bar{w},v^\bullet_1,\dots,v^\bullet_i)\\ = \\ \mathsf{partial\text{-}sig}^{d-i}_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G}',\bar{R}') \oplus_{\eta'} (\mathfrak{D},\bar{R}^\diamond),\mathbf{a}),\bar{w},u^\bullet_1,\dots,u^\bullet_i),\end{gathered} \tag{13}$$

where for every $j \in [i]$, $u^\bullet_j = u_j$ if $\omega_j = \bullet$ and $u^\bullet_j = \llcorner\!\lrcorner$ if $\omega_j = \circ$. Let $v_{i+1},\dots,v_d \in V(\mathfrak{H})$, where for each $j \in [i+1,r]$, $v_j \in \tilde{R}_j$. For every $j \in [i+1,d]$, let $(w_j,\omega_j) = \mathsf{stamp}_{w_1\dots w_{j-1}}(v_j)$. For every $j \in [i+1,d]$, we set

$$v^\bullet_j = \begin{cases} v_j, & \text{if } \omega_j = \bullet \\ \llcorner\!\lrcorner, & \text{if } \omega_j = \circ. \end{cases}$$

Therefore, for every $j \in [i+1,r]$, $v^\bullet_j \in \tilde{R}^{w_1\dots w_j}_j \cup \{\llcorner\!\lrcorner\}$ and for every $j \in [r+1,d]$, $v^\bullet_j \in V(\mathfrak{H})^{w_1\dots w_j} \cup \{\llcorner\!\lrcorner\}$.

Following (13), there are $u^\bullet_{i+1},\dots,u^\bullet_d \in V(\mathsf{ap}_{\mathbf{c}}(\mathfrak{G} \oplus_\eta \mathfrak{D},\mathbf{a})) \cup \{\llcorner\!\lrcorner\}$ where for every $j \in [i+1,r]$, $u^\bullet_j \in (R'_j \cup R^\diamond_j)^{w_1\dots w_j} \cup \{\llcorner\!\lrcorner\}$, and for every $j \in [r+1,d]$, $u^\bullet_j \in V(\mathsf{ap}_{\mathbf{c}}(\mathfrak{G} \oplus_\eta \mathfrak{D},\mathbf{a}))^{w_1\dots w_j} \cup \{\llcorner\!\lrcorner\}$, such that $\mathsf{pattern}(\mathbf{G}_{\bar{w}}, \llcorner\!\lrcorner^l, v^\bullet_1,\dots,v^\bullet_d) = \mathsf{pattern}(\mathbf{G}'_{\bar{w}}, \llcorner\!\lrcorner^l, u^\bullet_1,\dots,u^\bullet_d)$. Then, by Observation 7, we have that

$$\mathsf{compression}(\mathbf{G}_{\bar{w}}, \llcorner\!\lrcorner^l, v^\bullet_1,\dots,v^\bullet_d) = \mathsf{compression}(\mathbf{G}'_{\bar{w}}, \llcorner\!\lrcorner^l, u^\bullet_1,\dots,u^\bullet_d)$$

and therefore, if $I = \{j \in [d] \mid v^\bullet_j \neq \llcorner\!\lrcorner\}$ and $I' = \{j \in [d] \mid u^\bullet_j \neq \llcorner\!\lrcorner\}$, then

(P1) $I = I'$ and for every $j,\ell \in [d]$ $v^\bullet_j = v^\bullet_\ell$ if and only if $u^\bullet_j = u^\bullet_\ell$,

(P2) $\kappa : [d] \to [l]$ is the empty function,

(P3) for every $j \in [d]$, $\delta(j) = \delta'(j)$,

(P4) $\mathsf{Ind}_{G_{\bar{w}}}(I) = \mathsf{Ind}_{G'_{\bar{w}}}(I)$, and

(P5) $\{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(\mathbf{G}_{\bar{w}}, v^\bullet_1,\dots,v^\bullet_d)\} = \{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(\mathbf{G}'_{\bar{w}}, u^\bullet_1,\dots,u^\bullet_d)\}$.

We now define a sequence of vertices $u_{i+1}, \ldots, u_d \in V(\mathfrak{H}')$, as follows:

$$\text{for every } j \in [i+1, r], u_j = \begin{cases} u_j^\bullet, & \text{if } \omega_j = \bullet \\ v_j, & \text{if } \omega_j = \circ, \end{cases}$$

and we aim to show that $\mathsf{sig}_r^0(\mathfrak{H}, \mathbf{a}, v_1, \ldots, v_d) = \mathsf{sig}_r^0(\mathfrak{H}', \mathbf{a}, u_1, \ldots, u_d)$. To prove this, intuitively, we have to show that (P1)-(P5) also hold for the tuples $(v_1, \ldots, v_d)$ and $(u_1, \ldots, u_d)$. Keep in mind that $\mathsf{trace}_{(\mathcal{A}, \tilde{\mathfrak{R}}')}(u_1, \ldots, u_d) = \mathsf{trace}_{(\mathcal{A}, \tilde{\mathfrak{R}})}(v_1, \ldots, v_d)$.

First observe that, by (P1) and by the definition of $u_j, j \in [d]$, the partition of $[d]$ with respect to the equal vertices in $(v_1, \ldots, v_d)$ is the same as the one with respect to the equal vertices in $(u_1, \ldots, u_d)$.

Also, observe that using the definition of $\mathfrak{H}$ and $\mathfrak{H}'$, the fact that $\mathbf{a}$ is an apex-tuple of $\mathfrak{D}$, together with the properties (P2) and (P3), we have that, when modifiying the functions κ, δ to map the indices of $(v_1, \ldots, v_d)$ to $[l]$ and $[h]$ (before they mapped the indices of $(v_1^\bullet, \ldots, v_d^\bullet)$) and the functions κ', δ' to map the indices of $(u_1, \ldots, u_d)$ (before they mapped the indices of $(u_1^\bullet, \ldots, u_d^\bullet)$), for every $j \in [d]$, $\kappa(j) = \kappa'(j)$ and for every $j \in [d]$, $\delta(j) = \delta'(j)$.

We next show that $\mathsf{Ind}_{\mathfrak{H}}([d]) = \mathsf{Ind}_{\mathfrak{H}'}([d])$. Recall that, by (P1), $I = \{j \in [d] \mid v_j^\bullet \neq \text{\textvisiblespace}\} = \{j \in [d] \mid u_j^\bullet \neq \text{\textvisiblespace}\}$ and, by (P4), $\mathsf{Ind}_{G_{\bar{w}}}(I) = \mathsf{Ind}_{G'_{\bar{w}}}(I)$. Also, since $\mathsf{trace}_{(\mathcal{A}, \tilde{\mathfrak{R}}')}(u_1, \ldots, u_d) = \mathsf{trace}_{(\mathcal{A}, \tilde{\mathfrak{R}})}(v_1, \ldots, v_d) = (\bar{w}, \bar{\omega})$, we have that $\{v_1, \ldots, v_d\} \cap V(\mathsf{Influence}_{\tilde{\mathfrak{R}}}(\mathcal{A}_{\bar{w}})) = \emptyset$ and $\{u_1, \ldots, u_d\} \cap V(\mathsf{Influence}_{\tilde{\mathfrak{R}}'}(\mathcal{A}_{\bar{w}}))$. This implies that $\mathsf{Ind}_{\mathfrak{H}}(I) = \mathsf{Ind}_{\mathfrak{H}'}(I)$. Also, let $J = \{j \in [d] \mid v_j^\bullet = \text{\textvisiblespace}\} = \{j \in [d] \mid u_j^\bullet = \text{\textvisiblespace}\}$ and observe that $[d] \setminus I = J$. Also, observe that $\mathsf{Ind}_{\mathfrak{H}}(J) = \mathsf{Ind}_{\mathfrak{H}'}(J)$ and that $\mathsf{Ind}_{\mathfrak{H}}(J)$ is a subgraph of $\mathfrak{D} \oplus_\xi \mathfrak{F}$. The fact that $\{v_1, \ldots, v_d\} \cap V(\mathsf{Influence}_{\tilde{\mathfrak{R}}}(\mathcal{A}_{\bar{w}})) = \emptyset$ and $\{u_1, \ldots, u_d\} \cap V(\mathsf{Influence}_{\tilde{\mathfrak{R}}'}(\mathcal{A}_{\bar{w}}))$ implies that there is no edge neither in $\mathfrak{H}$ nor in $\mathfrak{H}'$ between vertices indexed by I and J. Therefore, both $\mathsf{Ind}_{\mathfrak{H}}([d])$ and $\mathsf{Ind}_{\mathfrak{H}'}([d])$ are equal to the disjoint union of $\mathsf{Ind}_{\mathfrak{H}}(I)$ and $\mathsf{Ind}_{\mathfrak{H}}(J)$.

We conclude the proof of the claim by showing that

$$\{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(\mathfrak{H}, v_1, \ldots, v_d)\} = \{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(\mathfrak{H}', u_1, \ldots, u_d)\}.$$

Let $(L, v_1, \ldots, v_d) \in \mathsf{Pairings}(\mathfrak{H}, v_1, \ldots, v_d)$. By Lemma 4, there exists a $(\tilde{L}, v_1, \ldots, v_d) \in \mathsf{Pairings}(\mathfrak{H}, v_1, \ldots, v_d)$ such that $L \equiv \tilde{L}$ and

$$(\tilde{L}, v_1, \ldots, v_d) \in \mathsf{Pairings}(\mathbf{G}_{\bar{w}}, \bar{v}^\bullet) \oplus \mathsf{Pairings}(\mathbf{G}_{\bar{w}}^{\mathsf{out}}, \bar{v}^\circ).$$

Recall that, by definition, $\mathsf{Leveling}_{(\mathcal{A}, \tilde{\mathfrak{R}})}(\mathfrak{H}) = \mathbf{G}_{\bar{w}} \oplus \mathbf{G}_{\bar{w}}^{\mathsf{out}}$. Note that the vertex set of $\mathbf{G}_{\bar{w}}^{\mathsf{out}}$ is a subset of $V(\mathfrak{D} \oplus_\xi \mathfrak{F})$ and therefore, $\mathsf{Leveling}_{(\mathcal{A}, \tilde{\mathfrak{R}}')}(\mathfrak{H}') = \mathbf{G}'_{\bar{w}} \oplus \mathbf{G}_{\bar{w}}^{\mathsf{out}}$. Also, since $\bar{v}^\circ = \bar{u}^\circ$, we have that $\mathsf{Pairings}(\mathbf{G}_{\bar{w}}^{\mathsf{out}}, \bar{v}^\circ) = \mathsf{Pairings}(\mathbf{G}_{\bar{w}}^{\mathsf{out}}, \bar{u}^\circ)$.

We set $\tilde{\mathbf{L}}^\bullet = (\tilde{L}, v_1, \ldots, v_d) \cap (\mathbf{G}_{\bar{w}}, \bar{v}^\bullet)$. By (P5), we have that $\{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(\mathbf{G}_{\bar{w}}, \bar{v}^\bullet)\} = \{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(\mathbf{G}'_{\bar{w}}, \bar{u}^\bullet)\}$. Therefore, there is an $\mathbf{L}' \in \mathsf{Pairings}(\mathbf{G}'_{\bar{w}}, \bar{u}^\bullet)$ such that $\mathsf{imp}(\tilde{\mathbf{L}}^\bullet) = \mathsf{imp}(\mathbf{L}')$. This, together with the fact that $\mathsf{Pairings}(\mathbf{G}_{\bar{w}}^{\mathsf{out}}, \bar{v}^\circ) = \mathsf{Pairings}(\mathbf{G}_{\bar{w}}^{\mathsf{out}}, \bar{u}^\circ)$, implies that there exists an $(L', u_1, \ldots, u_d) \in \mathsf{Pairings}(\mathfrak{H}', u_1, \ldots, u_r)$ such that $\mathsf{imp}(\tilde{L}, v_1, \ldots, v_d) = \mathsf{imp}(L', u_1, \ldots, u_d)$. Since $L \equiv \tilde{L}$ implies that $\mathsf{imp}(L, v_1, \ldots, v_d) = \mathsf{imp}(\tilde{L}, v_1, \ldots, v_d)$, we have that $\mathsf{imp}(L, v_1, \ldots, v_d) = \mathsf{imp}(L', u_1, \ldots, u_d)$. Therefore, we can conclude that $\{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(\mathfrak{H}, v_1, \ldots, v_d)\} \subseteq \{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(\mathfrak{H}', u_1, \ldots, u_d)\}$.

To show that $\{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(\mathfrak{H}', u_1, \ldots, u_d)\} \subseteq \{\mathsf{imp}(\mathbf{L}) \mid \mathbf{L} \in \mathsf{Pairings}(\mathfrak{H}, v_1, \ldots, v_d)\}$, we follow the same arguments. We consider an $(L', u_1, \ldots, u_d) \in \mathsf{Pairings}(\mathfrak{H}', u_1, \ldots, u_d)$, we apply Lemma 4 and we obtain a $(\tilde{L}', u_1, \ldots, u_d) \in \mathsf{Pairings}(\mathfrak{H}', u_1, \ldots, u_d)$ such that $(\tilde{L}', u_1, \ldots, u_d) \in \mathsf{Pairings}(\mathbf{G}'_{\bar{w}}, \bar{u}^\bullet) \oplus \mathsf{Pairings}(\mathbf{G}_{\bar{w}}^{\mathsf{out}\prime}, \bar{u}^\circ)$ and $\mathsf{imp}(L', u_1, \ldots, u_d) = \mathsf{imp}(\tilde{L}', u_1, \ldots, u_d)$. This, by (P5), implies the existence of an $\mathbf{L} \in \mathsf{Pairings}(\mathbf{G}_{\bar{w}}, \bar{v}^\bullet)$ such that $\mathsf{imp}(\tilde{\mathbf{L}}'^\bullet) = \mathsf{imp}(\mathbf{L})$, which, in turn, implies that there exists an $(L, v_1, \ldots, v_d) \in \mathsf{Pairings}(\mathfrak{H}, v_1, \ldots, v_d)$ such that $\mathsf{imp}(L, v_1, \ldots, v_d) = \mathsf{imp}(\tilde{L}', u_1, \ldots, u_d)$.

Therefore, for every $i \in [0, d]$ and every $\beta \in \mathcal{G}^{(d,h)}_{\mathsf{pat}}$, it holds that: For every $(t_0, \ldots, t_i) \in \mathsf{Paths}(T)$ there is a $(t'_0, \ldots, t'_i) \in \mathsf{Paths}(T')$ such that for every $t_d \in L(T_{t_i})$, where $(t_0, \ldots, t_d) \in \mathsf{Paths}(T)$, there is a $t'_d \in L(T'_{t'_i})$, where $(t'_0, \ldots, t'_d) \in \mathsf{Paths}(T')$, such that $\mathsf{sig}^0_r(\mathfrak{H}, \mathbf{a}, \lambda(t_1), \ldots, \lambda(t_d)) = \mathsf{sig}^0_r(\mathfrak{H}', \mathbf{a}, \lambda'(t'_1), \ldots, \lambda'(t'_d))$. This concludes the proof of the claim. $\diamond$

By Claim 3, we have that for every $i \in [0, d]$ and every $(t_0, \ldots, t_i) \in \mathsf{Paths}(T)$ there exist $(t'_0, \ldots, t'_i) \in \mathsf{Paths}(T')$ such that

$$\mathsf{sig}^{d-i}_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}, \lambda(t_1), \ldots, \lambda(t_i)) = \mathsf{sig}^{d-i}_r(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a}, \lambda'(t'_1), \ldots, \lambda'(t'_i)).$$

Symmetrically, one can prove that for every $i \in [0, d]$ and every $(t'_0, \ldots, t'_i) \in \mathsf{Paths}(T')$ there exist $(t_0, \ldots, t_i) \in \mathsf{Paths}(T)$ such that

$$\mathsf{sig}^{d-i}_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}, \lambda(t_1), \ldots, \lambda(t_i)) = \mathsf{sig}^{d-i}_r(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a}, \lambda'(t'_1), \ldots, \lambda'(t'_i)).$$

These two imply that $\mathsf{sig}^d_r(\mathfrak{H}, \tilde{R}_1, \ldots, \tilde{R}_r, \mathbf{a}) = \mathsf{sig}^d_r(\mathfrak{H}', \tilde{R}'_1, \ldots, \tilde{R}'_r, \mathbf{a})$. $\square$

8 Proof of Theorem 3

In this section, we aim to show Theorem 3. In this direction, in Subsection 8.1, we define *representatives* of vertices, following the notion of partial signatures defined in the previous section. Using Courcelle's theorem, we are able to compute these representatives and therefore obtain a "reduced" colored graph that has the same signature as the initial one (Lemma 6). Using this and in the presence of a big enough flat wall, in Subsection 8.2 we argue how to safely remove vertices from a bidimensional area of the graph where no variables of the sentence are quantified and obtain a "reduced" equivalent instance of the (annotated) problem. This will allow us to apply iteratively this procedure in order to reduce the treewidth of the given colored graph. We wrap-up the proof of Theorem 3 in Subsection 8.3.

8.1 Representatives

Let $d \in \mathbb{N}$ and $r \in [d-1]$. Let $(\mathfrak{G}, R_1, \ldots, R_r)$ be an (annotated) colored graph and let $(\mathcal{A}, \mathfrak{R})$ be a well-aligned $(f_5(d), \ell)$-railed annulus flatness pair of G. Recall that for every $i \in [d]$ and every $w_1, \ldots, w_i \in \{0, 1\}$, $R_i^{w_1 \ldots w_i} = \{v \in R_i \mid \mathsf{stamp}_{w_1 \ldots w_{i-1}}(v) = (w_i, \bullet)\}$ and $V^{w_1 \ldots w_i} = \{v \in V(\mathfrak{G}) \mid \mathsf{stamp}_{w_1 \ldots w_{i-1}}(v) = (w_i, \bullet)\}$. Given an $i \in [d]$ and $v_1, \ldots, v_{i-1} \in V(G) \cup \{\_\}$ such that for every $j \in [d-i]$, $v_j \in V^{w_1 \ldots w_j} \cup \{\_\}$, where $(w_1, \ldots, w_{i-1}, \bar{\omega}) = \mathsf{trace}_{(\mathcal{A}, \mathfrak{R})}(v_1, \ldots, v_{i-1})$, we say that two vertices $v, v' \in V(\mathfrak{G})$ such that $v, v' \in R_i^{w_1 \ldots w_{i-1} 0} \cup R_i^{w_1 \ldots w_{i-1} 1}$ if $i \in [\min\{r, d-i\}]$, and $v, v' \in V^{w_1 \ldots w_{i-1} 0} \cup V^{w_1 \ldots w_{i-1} 1}$ if $i \in [\min\{r, d-i\} + 1, d-i]$, are $((v_1, \ldots, v_{i-1}), i)$-*equivalent*, which we denote by $v \sim_i^{(v_1, \ldots, v_{i-1})} v'$, if $\mathsf{stamp}_{w_1 \ldots w_{i-1}}(v) = \mathsf{stamp}_{w_1 \ldots w_{i-1}}(v') = (w_i, \bullet)$ and

$$\mathsf{sig}^{d-i}_r(\mathfrak{G}, \bar{R}, w_1 \ldots w_{i-1} w_i, v_1, \ldots, v_{i-1}, v) = \mathsf{sig}^{d-i}_r(\mathfrak{G}, \bar{R}, w_1 \ldots w_{i-1} w_i, v_1, \ldots, v_{i-1}, v').$$

Following Observation 3 and Observation 10, we can easily derive an upper bound to the number of equivalence classes of each equivalence relation above.

Observation 11. *There is a function* $f_6 : \mathbb{N} \to \mathbb{N}$ *such that for every* $i \in [d]$ *and every* $v_1, \ldots, v_{i-1} \in V(G) \cup \{\_\}$*, such that for every* $j \in [i-1]$*,* $v_j \in V(G)^{w_1 \cdots w_j} \cup \{\_\}$*, where* $(w_1, \ldots, w_{i-1}, \bar{\omega}) = \mathsf{trace}_{(\mathcal{A},\mathfrak{R})}(v_1, \ldots, v_{i-1})$*, it holds that the number of equivalence classes of* $\sim_i^{(v_1,\ldots,v_{i-1})}$ *is at most* $f_6(d)$*.*

We can now prove the main result of this subsection.

Lemma 6. *There is a function* $f_7 : \mathbb{N} \to \mathbb{N}$ *and an algorithm that, given* $d, r, \mathsf{tw} \in \mathbb{N}$*, where* $r \in [d-1]$*, an* n*-vertex colored graph* $\mathfrak{G}$*, an apex-tuple* $\mathbf{a}$ *of* $\mathfrak{G}$ *of size* l*, a well-aligned* $(f_5(d), \ell)$*-railed annulus flatness pair* $(\mathcal{A}, \mathfrak{R})$ *of* $\mathfrak{G} \setminus V(\mathbf{a})$*, where* $\mathbf{tw}(G[Y_2]) \leq \mathsf{tw}$*, a set* $Z \subseteq Y_1$*, and sets* $R_1, \ldots, R_r \subseteq V(G)$*, outputs, in time* $\mathcal{O}_{d,\mathsf{tw}}(n)$*, sets* $R'_1, \ldots, R'_r \subseteq V(\mathfrak{G})$ *such that for every* $i \in [r]$*,* $R'_i \subseteq R_i$*,* $|R'_i \cap Z| \leq f_7(d)$*, and*

$$\mathsf{partial\text{-}sig}^d_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G}, R_1, \ldots, R_r), \mathbf{a})) = \mathsf{partial\text{-}sig}^d_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G}, R'_1, \ldots, R'_r), \mathbf{a})).$$

Proof. We set $f_7(d) = f_6(d)^{d+1}$. First observe that $\mathsf{partial\text{-}sig}^d_r$ can be expressed in MSOL. Assume that we have computed, for some $i \geq 1$, sets $R'_1, \ldots, R'_{i-1}$ as claimed. We show how to compute R'_i. The fact that $\mathbf{tw}(G[Y_2]) \leq \mathsf{tw}$ and that for every $\bar{w} \in \{0,1\}^d$, $V(\mathbf{G}_{\bar{w}})$ is a subset of X_2 implies that, by using Courcelle's theorem, we can compute, in time $\mathcal{O}_{d,\mathsf{tw}}(n)$, a subset R'_i of R_i such that $Y_2 \cap X_2 \subseteq R'_i$ and for every $v_1 \ldots, v_{i-1} \in V(G)$, where $v_j \in R'_j, j \in [i-1]$, it holds that for every v in R'_i such that $v \in Z$, if $v \sim_i^{(v_1,\ldots,v_{i-1})} u$, for some $u \in R_i$, then $u = v$. By keeping one v for every $v_1 \ldots, v_{i-1} \in V(G)$, where $v_j \in R'_j, j \in [i-1]$, we obtain a set R'_i as claimed. Intuitively, we obtain R'_i from R_i by keeping only one representative from each equivalence class (that itself can be expressed in MSOL) inside Z. By Observation 11, it follows that $|R'_i \cap Z| \leq f_7(d)$. $\square$

To conclude this subsection, we next show how to combine Lemma 6 and Lemma 5 in order to compute an equivalent colored graph with reduced annotation. Let two colored graphs $\mathfrak{G}, \mathfrak{H}$, let $\mathbf{a}$ be an apex-tuple of $\mathfrak{G}$, and let $(W, \mathfrak{R})$ be a flatness pair of $\mathfrak{G} \setminus V(\mathbf{a})$. Given that $\mathfrak{R} = (X, Y, P, C, \Gamma, \sigma, \pi)$, we call a partial function $\xi : V(X \cap Y) \cup V(\mathbf{a}) \to V(\mathfrak{H})$ such that $\mathfrak{G}$ and $\mathfrak{H}$ are ξ-compatible a *compatibility function of* $\mathfrak{G}$ *and* $\mathfrak{H}$.

Lemma 7. *Let* τ *be a colored-graph vocabulary. There is a function* $f_8 : \mathbb{N}^3 \to \mathbb{N}$ *and an algorithm that, given*

- $r, l, q \in \mathbb{N}$*,*
- *a* τ*-structure* $\mathfrak{G}$*,*
- *an apex-tuple* $\mathbf{a}$ *of* $\mathfrak{G}$ *of size* l*,*
- *a regular flatness pair* $(W, \mathfrak{R})$ *of* $\mathfrak{G} \setminus V(\mathbf{a})$ *of height at least* $f_8(r, l, q)$ *whose compass has treewidth at most* tw*, and*
- *sets* $R_1, \ldots, R_r \subseteq V(\mathfrak{G})$*,*

outputs, in time $\mathcal{O}_{r,l,q,\mathsf{tw}}(n)$*, sets* $R'_1, \ldots, R'_r \subseteq V(\mathfrak{G})$ *such that for every* $i \in [r]$*,* $R'_i \subseteq R_i$*, and a flatness pair* $(\tilde{W}', \tilde{\mathfrak{R}}')$ *of* $G \setminus V(\mathbf{a})$ *that is a* W'*-tilt of* $(W, \mathfrak{R})$ *for some* q*-subwall* W' *of* W *such thatfor every* $i \in [r]$*,* $R'_i \cap V(\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')) = \emptyset$ *and for every* $(\tau \cup \{\mathsf{R}_1, \ldots, \mathsf{R}_r\})$*-structure* $(\mathfrak{F}, \bar{R}^\star)$ *and every compatibility function* ξ *of* $(\mathfrak{G}, R_1, \ldots, R_r)$ *and* $(\mathfrak{F}, \bar{R}^\star)$*, it holds that*

$$\mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G}, R_1, \ldots, R_r)) = \mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G}, R'_1, \ldots, R'_r)).$$

Proof. We set and let $d = f_4(r,l)$, $p = f_5(d)$, $y = \ell$, $g = f_{17}(f_7(d)+1, q)$, and $f_8(d,y) = \mathsf{odd}(2p + \max\{g, \frac{y}{4} - 1\})$. We first apply Proposition 1, and obtain a (p,y)-railed annulus $\mathcal{A}$ of $\mathfrak{G} \setminus V(\mathbf{a})$. We set $(X, Y, P, C, \Gamma, \pi, \sigma) = \mathfrak{R}$. Let Σ be the union of all $W^{(g+1)}$-internal cells of $\mathfrak{R}$. By setting $Y_2 := Y$, $X_2 := X$, $Y_1 = \Sigma \cup \bigcup\{V(\sigma) \mid c \text{ is a } W^{(g+1)}\text{-marginal cell of } \mathfrak{R}\}$, and $X_1 = V(G) \setminus \Sigma$, we obtain a tuple $\mathfrak{R}' = (X_1, Y_1, X_2, Y_2, Z_1, Z_2, \Gamma', \pi', \sigma')$, where Γ', π', σ' are the restrictions of Γ, π, σ, respectively to $\Gamma \setminus \Sigma$.

Observe that since $(W, \mathfrak{R})$ is a flatness pair of $\mathfrak{G} \setminus V(\mathbf{a})$, then $(\mathcal{A}, \mathfrak{R}')$ is a railed annulus flatness pair of $\mathfrak{G} \setminus V(\mathbf{a})$. Also, since $(W, \mathfrak{R})$ is regular, it is also well-aligned (Proposition 10), which implies that $(\mathcal{A}, \mathfrak{R}')$ is also well-aligned. Moreover, since $\mathsf{Compass}_{\mathfrak{R}}(W)$ has treewidth at most tw, we have that $\mathbf{tw}(G[Y_2]) \leq \mathsf{tw}$. Therefore, by applying the algorithm of Lemma 6, we find, in time $\mathcal{O}_{d,\mathsf{tw}}(n)$, sets $R'_1, \ldots, R'_r \subseteq V(\mathfrak{G})$ such that for every $i \in [r]$, $R'_i \subseteq R_i$ and $|R'_i \cap Y_1| \leq f_7(d)$, and $\mathsf{partial\text{-}sig}^d_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G}, R_1, \ldots, R_r), \mathbf{a})) = \mathsf{partial\text{-}sig}^d_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G}, R'_1, \ldots, R'_r), \mathbf{a}))$. By using Proposition 6, we can compute, in linear time, a q-subwall W' of W such that $V(\mathsf{Influence}_{\mathfrak{R}}(W')) \subseteq Y_1$ and $R'_i \cap V(\mathsf{Influence}_{\mathfrak{R}}(W')) = \emptyset$ for every $i \in [r]$. Then, by applying the algorithm of Proposition 7 we compute, in linear time, a W'-tilt $(\tilde{W}', \tilde{\mathfrak{R}}')$ of $(W, \mathfrak{R})$ such that $R'_i \cap V(\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')) = \emptyset$, for every $i \in [r]$.

Since

$$\mathsf{partial\text{-}sig}^d_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G}, R_1, \ldots, R_r), \mathbf{a})) = \mathsf{partial\text{-}sig}^d_r(\mathsf{ap}_{\mathbf{c}}((\mathfrak{G}, R'_1, \ldots, R'_r), \mathbf{a})),$$

by Lemma 5, we have that for every $(\tau \cup \{\mathsf{R}_1, \ldots, \mathsf{R}_r\})$-structure $(\mathfrak{F}, \bar{R}^\star)$, every compatibility function ξ of $(\mathfrak{G}, R_1, \ldots, R_r)$ and $(\mathfrak{F}, \bar{R}^\star)$, it holds that $\mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G}, R_1, \ldots, R_r)) = \mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G}, R'_1, \ldots, R'_r))$. ☐

8.2 Reducing the instance

In this subsection we describe how to remove problem-irrelevant vertices inside a "big enough" annotation-irrelevant bidimensional area of our instance. In order to achieve this, we have to argue that a linkage whose terminals are not intersecting a big enough bidimensional area can be rerouted away from some central part of this area. This is guaranteed by the following version of the *Unique Linkage Theorem* that can be derived from [13, Theorem 23] (see also [3, 148, 171, 194, 195]).

Proposition 3. *There exists a function $f_9 : \mathbb{N}^3 \to \mathbb{N}$ such that, for every $l, z, k \in \mathbb{N}$, if G is a graph, $\mathbf{a}$ is an apex-tuple of G of size l, if $(W, \mathfrak{R})$ is flatness pair of $G \setminus V(\mathbf{a})$ of height $f_9(l, z, k)$ then for every linkage L of size at most k such that $T(L) \cap V(\mathsf{Compass}_{\mathfrak{R}}(W)) = \emptyset$, there is a linkage L' such that $L \equiv L'$ and $L \cap V(\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')) = \emptyset$, where $(\tilde{W}', \tilde{\mathfrak{R}}')$ is a $W^{(z)}$-tilt of $(W, \mathfrak{R})$.*

Using Proposition 3, we can easily prove the following result that allows us to remove problem-irrelevant vertices inside a "big enough" annotation-irrelevant bidimensional area of our instance.

Lemma 8. *Let τ be a colored-graph vocabulary. There is a function $f_{10} : \mathbb{N}^2 \to \mathbb{N}$ such that, if*

- *$r, l, z \in \mathbb{N}$,*
- *$\mathfrak{G}$ is a τ-structure,*
- *$\mathbf{a}$ is an apex-tuple of $\mathfrak{G}$ of size l,*
- *$(W, \mathfrak{R})$ is a flatness pair of $\mathfrak{G} \setminus V(\mathbf{a})$ of height $f_{10}(r, l) + z$, and*
- *sets $R_1, \ldots, R_r \subseteq V(\mathfrak{G})$, where for every $i \in [r]$, $R_i \cap V(\mathsf{Compass}_{\mathfrak{R}}(W)) = \emptyset$,*

then for every flatness pair $(\tilde{W}', \tilde{\mathfrak{R}}')$ of $\mathfrak{G} \setminus V(\mathbf{a})$ that is a W'-tilt of $(W, \mathfrak{R})$ for some $f_{10}(q,l)$-internal subwall W' of W of height z, for every $(\tau \cup \{\mathsf{R}_1, \ldots, \mathsf{R}_r\})$-structure $(\mathfrak{F}, \bar{R}^\star)$, every compatibility function ξ of $(\mathfrak{G}, R_1, \ldots, R_r)$ and $(\mathfrak{F}, \bar{R}^\star)$, and every $Y \subseteq V(\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}'))$, it holds that

$$\mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G}, R_1, \ldots, R_r)) = \mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G} \setminus Y, R_1, \ldots, R_r)).$$

Proof. We set $f_{10}(r,l) = f_9(l,z,r)$. Let $\mathfrak{G}$ be a τ-structure, let $\mathbf{a}$ be an apex-tuple of $\mathfrak{G}$ of size l, let $(W, \mathfrak{R})$ be regular flatness pair of $\mathfrak{G} \setminus V(\mathbf{a})$ of height $f_{10}(r,l) + z$, and $R_1, \ldots, R_r \subseteq V(\mathfrak{G})$, where $R_i \cap V(\mathsf{Compass}_{\mathfrak{R}}(W)) = \emptyset$ for every $i \in [r]$. Also, let $(\tilde{W}', \tilde{\mathfrak{R}}')$ be a flatness pair of $\mathfrak{G} \setminus V(\mathbf{a})$ that is a W'-tilt of $(W, \mathfrak{R})$ for some $f_{10}(q,l)$-internal subwall W' of W of height z. Let also $Y \subseteq V(\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}'))$. Observe that, for every $k \in \lfloor r/2 \rfloor$ and every $s_1, t_1 \ldots, s_k, t_k \in R$, by Proposition 3, if G is the Gaifman graph of $(\mathfrak{G}, R)$, then $(G, s_1, t_1, \ldots, s_k, t_k) \models \mathsf{dp}_k(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_k, \mathsf{y}_k)$ if and only if $(G \setminus Y, s_1, t_1, \ldots, s_k, t_k) \models \mathsf{dp}_k(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_k, \mathsf{y}_k)$. This implies that $\mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G}, R_1, \ldots, R_r)) = \mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G} \setminus Y, R_1, \ldots, R_r))$. $\square$

By applying Lemma 7 and then Lemma 8, we get the following:

Lemma 9. *Let τ be a colored-graph vocabulary. There is a function $f_{11} : \mathbb{N}^3 \to \mathbb{N}$ and an algorithm that, given $r, l, q \in \mathbb{N}$, a τ-structure $\mathfrak{G}$, an apex-tuple $\mathbf{a}$ of $\mathfrak{G}$ of size l, a regular flatness pair $(W, \mathfrak{R})$ of $\mathfrak{G} \setminus V(\mathbf{a})$ of height $h \geq f_{11}(r,l,q)$ whose compass has treewidth at most tw, and sets $R_1, \ldots, R_r \subseteq V(G)$, outputs, in time $\mathcal{O}_{r,l,q,\mathsf{tw}}(n)$, sets $R'_1, \ldots, R'_r \subseteq V(\mathfrak{G})$ and a flatness pair $(\tilde{W}', \tilde{\mathfrak{R}}')$ of $\mathfrak{G} \setminus V(\mathbf{a})$ that is a W'-tilt of $(W, \mathfrak{R})$ for some q-subwall W' of W such that for every $i \in [r]$, $R'_i \subseteq R_i$ and $R'_i \cap V(\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')) = \emptyset$, and for every $(\tau \cup \{\mathsf{R}_1, \ldots, \mathsf{R}_r\})$-structure $(\mathfrak{F}, \bar{R}^\star)$, every compatibility function ξ of $(\mathfrak{G}, R_1, \ldots, R_r)$ and $(\mathfrak{F}, \bar{R}^\star)$, and every $Y \subseteq V(\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}'))$, it holds that*

$$\mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G}, R_1, \ldots, R_r)) = \mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G} \setminus Y, R'_1, \ldots, R'_r)).$$

8.3 Proof of Theorem 3

We conclude this section by presenting the proof of Theorem 3.

Proof of Theorem 3. Given a sentence $\varphi \in \mathsf{FOL}[\tau + \mathsf{dp}]$ of quantifier rank q, we set

$$\begin{aligned} c &:= \mathsf{hw}(G), \\ l &:= f_{19}(c) \text{ where } f_{19} \text{ is the function of Proposition 9, } \textit{and} \\ r &:= f_{11}(q, l, 3). \end{aligned}$$

Our algorithm consists of three steps.

Step 1: Run the algorithm of Proposition 9 for G, r, and c. This algorithm outputs, in linear time, either a tree decomposition of G of width at most $f_{20}(c) \cdot r$, or a set $A \subseteq V(G)$, where $|A| \leq l$, a regular flatness pair $(W, \mathfrak{R})$ of $G \setminus A$ of height r, and a tree decomposition of $\mathsf{compass}_{\mathfrak{R}}(W)$ of width at most $f_{20}(c) \cdot r$. In the first possible output, i.e., a tree decomposition of G of width at most $f_{20}(c) \cdot r$, proceed to Step 3. In the second possible output, proceed to Step 2.

Step 2: We run the algorithm of Lemma 9 for $q, l, 3$, $\mathfrak{G}$, $R_1, \ldots, R_q$ $\mathbf{a}$, and $(W, \mathfrak{R})$, and we obtain, in linear time, sets $R'_1, \ldots, R'_q \subseteq V(G)$ and a flatness pair $(\tilde{W}', \tilde{\mathfrak{R}}')$ of $G \setminus V(\mathbf{a})$ that is a W'-tilt of $(W, \mathfrak{R})$ for some subwall W' of W of height 3 such that for every $i \in [q]$, $V(\mathsf{compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')) \cap R'_i = \emptyset$ and $R'_i \subseteq R_i$, and $\mathsf{sig}^q(\mathfrak{G}, R_1, \ldots, R_q) = \mathsf{sig}^q(\mathfrak{G} \setminus V(\mathsf{compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')), R'_1, \ldots, R'_q)$. Then, we set $\mathfrak{G} := \mathfrak{G} \setminus V(\mathsf{compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}'))$, for every $i \in [q]$, $R_i := R'_i$, and we run again Step 1.

Step 3: Given a tree decomposition of G of width at most $f_{20}(c) \cdot r$, and since sig^q is expressible in $\mathsf{MSOL}[\tau]$, by using Courcelle's theorem, in linear time we can compute $\mathsf{sig}^q(\mathfrak{G}, R_1, \ldots, R_q)$ and check the existence of a φ-spanning subtree of the tree obtained by $\mathsf{sig}^q(\mathfrak{G}, R_1, \ldots, R_q)$ and therefore decide whether $\mathfrak{G} \models \varphi$.

Observe that the first and the second step of the algorithm are executed in linear time and they can be repeated no more than a linear number of times. Therefore, the overall algorithm runs in quadratic time, as claimed. $\square$

9 Logic for s-scattered paths

In this section we define a class of extensions of FOL, in the same spirit as the disjoint paths logic.

9.1 Definition of s-scattered paths logic

Let $r \in \mathbb{N}$. We define the $2k$-ary predicate $s\text{-}\mathsf{sdp}_k(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_k, \mathsf{y}_k)$, which evaluates true in a τ-structure $\mathfrak{G}$ if and only if there are paths $P_1, \ldots, P_k$ of $(V(\mathfrak{G}), \mathsf{E}^{\mathfrak{G}})$ of length at least 2 between (the interpretations of) x_i and y_i for all $i \in [k]$ such that for every $i, j \in [k]$, $j \neq i$, $V(P_i) \cap N^{(\leq s)}_{(V(\mathfrak{G}), \mathsf{E}^{\mathfrak{G}})}(V(P_j)) = \emptyset$. We let $\tau + s\text{-}\mathsf{sdp} := \tau \cup \{i\text{-}\mathsf{sdp}_k \mid k \geq 1, i \leq s\}$, where each $s\text{-}\mathsf{sdp}_k$ is a $2k$-ary relation symbol. We use $s\text{-}\mathsf{sdp}$ instead of $s\text{-}\mathsf{sdp}_k$ when k is clear from the context. It is easy to see that for every $k \in \mathbb{N}$ and every $x_1, y_1, \ldots, x_k, y_k \in V(G)$,

$$G \models 0\text{-}\mathsf{sdp}_k(x_1, y_1, \ldots, x_k, y_k) \iff G \models \mathsf{dp}_k(x_1, y_1, \ldots, x_k, y_k).$$

Therefore, we can observe the following.

Observation 12. *For every colored-graph vocabulary τ, it holds that* $\{\mathrm{Mod}(\varphi) \mid \varphi \in \mathsf{FOL}[\tau + \mathsf{dp}]\} = \{\mathrm{Mod}(\varphi) \mid \varphi \in \mathsf{FOL}[\tau + 0\text{-}\mathsf{sdp}]\}$.

Our main result is the following:

Theorem 4. *For every colored-graph vocabulary τ, every $s \in \mathbb{N}$, and every sentence $\varphi \in \mathsf{FOL}[\tau + s\text{-}\mathsf{sdp}]$, there exists an algorithm that, given a τ-structure G of size n, outputs whether $G \models \varphi$ in time $\mathcal{O}_{|\varphi|,k}(n^2)$, where k is the Euler genus of G.*

9.2 Proof of Theorem 4

In this subsection, we sketch how to prove Theorem 4. Our strategy is the same as for the proof of Theorem 3 and in what follows we discuss how to recreate definitions and results of the previous sections in the setting of s-scattered paths. We use $s\text{-}\mathsf{sig}$ to denote the signature obtained from sig where we replace the atomic types concerning disjoint paths predicates with scattered disjoint path predicates.

First of all, we stress that since we work with graphs of bounded Euler genus, we can avoid the use of the framework of flat walls. In fact, using [102, Lemma 6.2] (see also [18,47,60,60–62,83,96]), we can find a disk-embedded wall of bounded treewidth compass in a surface-embedded of large enough treewidth. Before presenting this result, we give some additional definitions. Given a closed disk Δ and an integer $q \in \mathbb{N}_{\geq 3}$, a Δ-*embedded q-wall* of G is a q-wall W that is embedded on Δ and whose perimeter is the boundary of Δ. The *compass* of W, denoted by $\mathsf{Compass}(W)$, is the graph $G \cap \Delta$.

Proposition 4. *There exists a constant c_1 and an algorithm that given an n-vertex graph G of Euler genus at most g and an integer $q \in \mathbb{N}_{\geq 3}$, outputs either a closed disk Δ and a Δ-embedded q-wall W of G whose compass has treewidth at most $c_1 \cdot q$ or a tree decomposition of G of width at most $c_1 \cdot q$. Moreover, this algorithm runs in $2^{\mathcal{O}_g(q^2)} \cdot n$ time.*

Patterns for s-scattered linkages. Next step is to define *patterns* that encode s-scattered linkages. Given an $s \in \mathbb{N}$, we say that a linkage L of G is *s-scattered* if for every $v \in V(L)$ it holds that $N_G^{(\leq s)}(v) \cap (V(L) \setminus V(C_v)) = \emptyset$, where C_v is the connected component of L that contains v. Observe that, since $N_G^{(\leq 0)}(v) = \{v\}$, every linkage is 0-scattered.

To define *patterns* that encode s-scattered linkages, the definition of a pattern of a boundaried colored graph in Subsection 4.2 has to be modified as follows: the collection $\mathcal{H}^P$ is defined as all graphs (I, E), where $E \subseteq I \times I$, such that G contains s-scattered paths of length at least two between the vertices v_i, v_j for all $\{i, j\} \in E$.

Also, as in Subsection 4.3, we define the *pattern* of a quantifier-free formula in $\mathsf{FOL}[\tau + s\text{-}\mathsf{sdp}]$ by modifying the corresponding definition for formulas in $\mathsf{FOL}[\tau{+}\mathsf{dp}]$. We do this by replacing, in both definitions of full clauses and in the definition of the collection $\mathcal{H}_c^P$, every appearance of the atomic formula dp with the atomic formula s-sdp.

After the above modifications, Observation 6 holds also for sentences $\varphi \in \mathsf{FOL}[\tau + s\text{-}\mathsf{sdp}]$. Based on this observation (for s-scattered linkages), we can then prove Lemma 2 for any sentence $\varphi \in \mathsf{FOL}[\tau + s\text{-}\mathsf{sdp}]$.

Routing s-scattered linkages through railed annuli. Having Lemma 2 in hand, our next goal is to prove Lemma 4 for the case of s-scattered linkages. Recall that Lemma 4 intuitively states that, in the presence of a flat railed annulus $\mathcal{A}$ inside a given graph G, every linkage L of G can be combed through some paths of $\mathcal{A}$ in some "buffer" (obtained by some hierarchical refinement of $\mathcal{A}$; see Subsection 7.2) of $\mathcal{A}$ corresponding to the position of the terminals of L. In the case of $\mathsf{FOL}[\tau{+}\mathsf{dp}]$, this is essentially a reformulation of Proposition 2 to the setting of pairings and flat railed annuli. Therefore, to generalize Lemma 4 to s-scattered linkages, one has to prove the analogue of Proposition 2 for s-scattered linkages. This was done in [103] and is stated below. However, this result is proven for graphs embeddable in some fixed surface and this is the reason why Theorem 4 holds up to graphs of bounded-genus.

Proposition 5. *There exist two functions $f_{12}, f_{13} : \mathbb{N}^3 \to \mathbb{N}$ such that for every odd $\ell \in \mathbb{N}$ and every $s, k, g \in \mathbb{N}$, if*

- *Σ is a surface of Euler genus g,*
- *Δ is a closed annulus of Σ,*
- *G is a graph embedded in Σ*
- *$\mathcal{A} = (\mathcal{C}, \mathcal{P})$ is a Δ-embedded (p, q)-railed annulus of G, where $p \geq f_{12}(s, k, g) + \ell$ and $q \geq \frac{2s+5}{2} \cdot f_{13}(s, k, g)$*
- *L is a Δ-avoiding s-scattered linkage of size at most k, and*
- *$I \subseteq [q]$, where $|I| > f_{13}(s, k, g) \cdot (s + 1)$,*

then G contains an s-scattered linkage $\tilde{L}$ where $\tilde{L} \equiv L$, $\tilde{L} \setminus \Delta \subseteq L \setminus \Delta$, and $\tilde{L}$ is (ℓ, I)-confined in $\mathcal{A}$. Moreover, $f_{12}(s, k, g) = \mathcal{O}((f_{13}(s, k, g))^2)$ and $f_{13}(s, k, g) = s \cdot 2^{\mathcal{O}(k+g)}$.

Using Proposition 5 and adjusting the definitions in Subsection 5.3 for the s-scattered linkages setting (this is done by just modifying the definition of pairings to consider (boundaried) linkages that are s-scattered), we can prove the analogue of Lemma 4.

Partial signatures and exchangeability for $\mathsf{FOL}[\tau + s\text{-}\mathsf{sdp}]$. The next important task is to define partial signatures of colored graphs (equipped with a disk-embedded railed annulus) that encode the (recursive) containment of "meta-collections" of s-scattered linkages in a recursively obtained collection of boundaried (sub)graphs. This is done as for $\mathsf{FOL}[\tau{+}\mathsf{dp}]$ (see Subsection 7.4) by building the recursive definition of partial signatures using the "s-scattered" patterns, as defined two paragraphs above, for the base of the recursion. Using this definition, one can formulate Lemma 5 for $\mathsf{FOL}[\tau + s\text{-}\mathsf{sdp}]$. The proof of this analogous version of Lemma 5 is actually the same as the one in Subsection 7.5 and uses the "s-scattered" versions of Lemma 2 and Lemma 4. Let us stress that since Proposition 5 demands that the given graph G is embedded on a fixed surface, this also holds for the "s-scattered" version of Lemma 4 and therefore, in the "s-scattered" version of Lemma 5, all considered colored graphs $\mathfrak{G}, \mathfrak{G}', \mathfrak{H}$, and $\mathfrak{F}$ should be embedded on some fixed surface and their "gluing" should also preserve embeddability. Also, since we deal with graphs of bounded Euler genus and using Proposition 4 we can directly obtain a disk-embedded wall if our input graph has large enough treewidth, there are no apices to deal with. Therefore, for $\mathsf{FOL}[\tau + s\text{-}\mathsf{sdp}]$, we do not need to apply the transformations of Section 6.

Finding representatives and proof of Theorem 4. From this point on, the steps towards the proof of Theorem 4 are completely analogous to the ones in Section 8 for the proof of Theorem 3. With the "s-scattered" version of Lemma 5 in our toolbox, we have to *find* an (annotated) colored graph with the same partial signature as the original one. This will allow us to reduce the annotation of the original colored graph. To do this, we define representatives of vertices with the same recursive "s-scattered" partial signature as in Subsection 8.1 and we deduce the "s-scattered" analogue of Lemma 6. In turn, Lemma 6, when combined with Lemma 5, implies Lemma 7 for sentences in $\mathsf{FOL}[\tau + s\text{-}\mathsf{sdp}]$. Last remaining piece is to prove the following analogue of Lemma 8 for sentences in $\mathsf{FOL}[\tau + s\text{-}\mathsf{sdp}]$. Given two colored graphs $\mathfrak{G}, \mathfrak{H}$ and a disk-embedded wall W of $\mathfrak{G}$, a *perimeter-compatibility function* of $\mathfrak{G}$ and $\mathfrak{H}$ is any partial function $\xi : V(D(W)) \to V(\mathfrak{H})$ such that $\mathfrak{G}$ and $\mathfrak{H}$ are ξ-compatible.

Lemma 10. *Let τ be a colored-graph vocabulary. There is a function $f_{14} : \mathbb{N}^3 \to \mathbb{N}$ such that, if*

- *$s, g, r, q \in \mathbb{N}$,*
- *$\mathfrak{G}$ is a τ-structure embedded in a surface Σ of Euler genus g,*
- *Δ is a closed disk of Σ,*
- *W is a Δ-embedded $(f_{14}(s, r, g) + q)$-wall of $\mathfrak{G}$, and*
- *$R_1, \ldots, R_r \subseteq V(\mathfrak{G})$, where $\cup_{i \in [r]} R_i$ is disjoint from $V(\mathsf{Compass}(W))$,*

then for each $f_{14}(s, r, g)$-internal q-subwall W' of W, every $(\tau \cup \{\mathsf{R}_1, \ldots, \mathsf{R}_r\})$-structure $(\mathfrak{F}, \bar{R}^\star)$, every compatibility function ξ of $(\mathfrak{G}, R_1, \ldots, R_r)$ and $(\mathfrak{F}, \bar{R}^\star)$, and every $Y \subseteq V(\mathsf{Compass}(W'))$, it holds that

$$s\text{-}\mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G}, R_1, \ldots, R_r)) = s\text{-}\mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G} \setminus Y, R_1, \ldots, R_r)).$$

The proof of Lemma 10 is obtained by the proof of Lemma 8 (see Subsection 8.2) by replacing the application of Proposition 3 by Proposition 5 combined with an "s-scattered" version of Lemma 12 (for the proof of the latter, it is easy to observe that it can be directly generalized to s-scattered linkages).

Then, using the "s-scattered" version of Lemma 7 and Lemma 10, we obtain the analogue of Lemma 9, which we state below.

Lemma 11. *There are two functions $f_{15} : \mathbb{N}^3 \to \mathbb{N}$ and $f_{16} : \mathbb{N}^4 \to \mathbb{N}$ and an algorithm that, given*

- $s, g, r, q \in \mathbb{N}$,
- *a τ-structure $\mathfrak{G}$ of Euler genus at most g,*
- *a closed disk Δ,*
- *a Δ-embedded wall W of $\mathfrak{G}$ of height at least $f_{15}(s, r, q)$ whose compass has treewidth at most* tw, *and*
- *sets $R_1, \ldots, R_r \subseteq V(\mathfrak{G})$,*

outputs, in time $\mathcal{O}_{s,g,t,q,\mathsf{tw}}(n)$, sets $R'_1, \ldots, R'_r \subseteq V(\mathfrak{G})$ and a q-subwall W' of W such that for every $Y \subseteq V(\mathsf{Compass}(W'))$, every $(\tau \cup \{\mathsf{R}_1, \ldots, \mathsf{R}_r\})$-structure $(\mathfrak{F}, \bar{R}^\star)$, every compatibility function ξ of $(\mathfrak{G}, R_1, \ldots, R_r)$ and $(\mathfrak{F}, \bar{R}^\star)$,

$$s\text{-}\mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G}, R_1, \ldots, R_r)) = s\text{-}\mathsf{sig}^r((\mathfrak{F}, \bar{R}^\star) \oplus_\xi (\mathfrak{G} \setminus Y, R'_1, \ldots, R'_r)).$$

The proof of Theorem 4 is obtained from the one of Theorem 3 (see Subsection 8.3), by plugging Proposition 4 and Lemma 11 instead of Proposition 9 and Lemma 9, respectively.

10 Conclusions and open problems

In this paper we proved two AMT's for the logic FOL+DP and its newly introduced extension FOL+SDP on graphs of bounded Hadwiger number and Euler genus respectively. These two logics can be seen as non-trivial extensions of FOL, as they may express a wide range of problems (and meta-problems) that are not FOL-expressible. (See Appendix A for an exposition of the expressivity potential of FOL+DP and FOL+SDP.)

10.1 Open problems

Recall that FOL+DP is an extension of the *separator logic* FOL+conn, introduced in [22, 201]. The combinatorial condition given in [188] for this logic is having bounded Hajós number. As minor excluding graphs classes are also topological-minor excluding classes, the combinatorial condition of [188] is more general that the one that we give for FOL+DP in this paper. This makes the AMT of [188] non-comparable to ours. Moreover it is shown in [188] that, under certain complexity assumptions, the bounded Hajós number demand is actually demarking the combinatorial horizon of FOL+conn. The open question is whether having bounded Hajós number is also the combinatorial horizon of the more expressive FOL+DP. We are not in position to make a positive or negative conjecture on this. We wish only to comment that the algorithmic/combinatorial tools that where used in [188] are quite different than the ones used in this paper.

Another open question is to what extend one may further strengthen the expressibility $\mathsf{dp}_k(\cdot)$ (resp. $s\text{-}\mathsf{sdp}_k(\cdot)$) predicate, while maintaining the combinatorial condition of bounded Hadwiger number (resp. bounded Euler genus). A possible candidate might be to ask for paths of *guided disjointness*, that is to consider the predicate $\mathsf{gdp}_H(x_1, y_1, \ldots, x_k, y_k)$ where H is a graph where $V(H) = \{1, \ldots, k\}$ and where we ask that, for every edge $(i, j) \in E(H)$, the (x_i, y_i)-path and the (x_j, y_j)-path are disjoint. Clearly, $\mathsf{dp}_k(x_1, y_1, \ldots, x_k, y_k) = \mathsf{gdp}_{K_k}(x_1, y_1, \ldots, x_k, y_k)$, therefore this would provide a more general logic than FOL+DP. To our knowledge, even the parameterized complexity of the evaluation of $\mathsf{dp}_H(x_1, y_1, \ldots, x_k, y_k)$, when parameterized by $k = |H|$, is an interesting open problem.

10.2 Limitations

In the beginning of Appendix A we comment that FOL+DP, on multicolored (by z colors) graphs, may express the predicate $\mathsf{dp}_{k,\lambda}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$ equipped with a list function $\lambda : [k] \to 2^{[z]}$, where we demand that, for every $i \in [k]$, $\lambda(i)$ is a subset of the set of all colors assigned to the vertices of the path between the (valuations) of s_1 and t_1. Interestingly this permits us to *demand* certain colors to be traversed by the disjoint paths. However, on the negative side, we may not expect that FOL+DP may *exclude* colors. From the empirical point of view, such a demand obstructs the application of the irrelevant vertex technique. Moreover, we may have more solid evidence of this by picking a typical example of such a problem. In Subsection C.2, we present a colored variant of the Topological Minor Containment problem, namely the Monochromatic Path Topological Minor problem that we prove (Theorem 6) that is W[1]-hard on planar graphs.

Acknowledgements. We would like to thank Anuj Dawar and the anonymous reviewers of previous versions of this paper. Their comments led to an improvement of the presentation of our results.

References

[1] Isolde Adler, Martin Grohe, and Stephan Kreutzer. Computing excluded minors. In *Proc. of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 641–650, 2008. URL: http://portal.acm.org/citation.cfm?id=1347082.1347153. 4

[2] Isolde Adler, Stavros G. Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Irrelevant vertices for the planar disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 122:815–843, 2017. doi:10.1016/j.jctb.2016.10.001. 4

[3] Isolde Adler, Stavros G. Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Irrelevant vertices for the planar Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B*, 122:815–843, 2017. doi:10.1016/j.jctb.2016.10.001. 47

[4] Akanksha Agrawal, Lawqueen Kanesh, Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Elimination distance to topological-minor-free graphs is FPT, 2021. arXiv:2104.09950. 77

[5] Akanksha Agrawal, Lawqueen Kanesh, Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Deleting, eliminating and decomposing to hereditary classes are all fpt-equivalent. In *Proc. of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1976–2004. SIAM, 2022. doi:10.1137/1.9781611977073.79. 5, 77

[6] Akanksha Agrawal, Lawqueen Kanesh, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. An FPT Algorithm for Elimination Distance to Bounded Degree Graphs. In *Proc. of the 38th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 187 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:11, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2021.5. 77

[7] Akanksha Agrawal and M. S. Ramanujan. On the parameterized complexity of clique elimination distance. In *Proc. of the 15th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 180 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:13, Dagstuhl, Germany, 2020. doi:10.4230/LIPIcs.IPEC.2020.1. 77

[8] Robert E. L. Aldred, Sheng Bau, Derek A. Holton, and Brendan D. McKay. Cycles through 23 vertices in 3-connected cubic planar graphs. *Graphs and Combinatorics*, 15(4):373–376, 1999. doi:10.1007/s003730050046.

[9] Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991. doi:10.1016/0196-6774(91)90006-K. 1

[10] Jakub Balabán and Petr Hlinený. Twin-width is linear in the poset width. In *Proc. of the 16th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 214 of *LIPIcs*, pages 6:1–6:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.IPEC.2021.6. 1

[11] Valentin Bartier, Nicolas Bousquet, Clément Dallard, Kyle Lomer, and Amer E. Mouawad. On girth and the parameterized complexity of token sliding and token jumping. *Algorithmica*, 83(9):2914–2951, 2021. doi:10.1007/s00453-021-00848-1. 78

[12] Valentin Bartier, Nicolas Bousquet, and Amer E. Mouawad. Galactic token sliding, 2022. arXiv:2204.05549. 78

[13] Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. IV. an optimal algorithm, 2019. arXiv:1907.04442. 47

[14] Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary. In *Proc. of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 951–970, 2020. doi:10.1137/1.9781611975994.57. 4, 26

[15] Rémy Belmonte, Tesshu Hanaka, Michael Lampis, Hirotaka Ono, and Yota Otachi. Independent set reconfiguration parameterized by modular-width. *Algorithmica*, 82(9):2586–2605, 2020. doi:10.1007/s00453-020-00700-y. 78

[16] Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, Yota Otachi, and Florian Sikora. Token sliding on split graphs. *Theory of Computing Systems*, 65(4):662–686, 2021. doi:10.1007/s00224-020-09967-8. 78

[17] Pierre Bergé, Édouard Bonnet, and Hugues Déprés. Deciding twin-width at most 4 is np-complete, 2021. arXiv:2112.08953. 1

[18] Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016. doi:10.1137/130947374. 49

[19] Hans L. Bodlaender, Carla Groenland, Jesper Nederlof, and Céline M. F. Swennenhuis. Parameterized problems complete for nondeterministic FPT time and logarithmic space. In *Proc. of the 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 193–204. IEEE, 2021. doi:10.1109/FOCS52979.2021.00027. 78

[20] Hans L. Bodlaender, Carla Groenland, and Céline M. F. Swennenhuis. Parameterized Complexities of Dominating and Independent Set Reconfiguration. In *Proc. of the 16th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 214 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:16, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.IPEC.2021.9. 78

[21] Mikolaj Bojanczyk and Michal Pilipczuk. Definability equals recognizability for graphs of bounded treewidth. In *Proc. of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 407–416. ACM, 2016. doi:10.1145/2933575.2934508. 1

[22] Mikołaj Bojańczyk. Separator logic and star-free expressions for graphs, 2021. arXiv:2107.13953. 2, 52

[23] Béla Bollobás and Andrew Thomason. Highly linked graphs. *Combinatorica*, 16(3):313–320, 1996. doi:10.1007/BF01261316. 72

[24] Marthe Bonamy, Nicolas Bousquet, Marc Heinrich, Takehiro Ito, Yusuke Kobayashi, Arnaud Mary, Moritz Mühlenthaler, and Kunihiro Wasa. The perfect matching reconfiguration problem. In *Proc. of the 44th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 138 of *LIPIcs*, pages 80:1–80:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.MFCS.2019.80. 78

[25] Marthe Bonamy, Paul Dorbec, and Paul Ouvrard. Dominating sets reconfiguration under token sliding. *Discrete Applied Mathematics*, 301:6–18, 2021. doi:https://doi.org/10.1016/j.dam.2021.05.014. 78

[26] Édouard Bonnet, Dibyayan Chakraborty, Eun Jung Kim, Noleen Köhler, Raul Lopes, and Stéphan Thomassé. Twin-width VIII: delineation and win-wins, 2022. arXiv:2204.00722. 1

[27] Édouard Bonnet, Jan Dreier, Jakub Gajarský, Stephan Kreutzer, Nikolas Mählmann, Pierre Simon, and Szymon Torunczyk. Model checking on interpretations of classes of bounded local cliquewidth, 2022. arXiv:2202.13014. 2

[28] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: small classes. In *Proc. of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1977–1996. SIAM, 2021. doi:10.1137/1.9781611976465.118. 1

[29] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width III: max independent set, min dominating set, and coloring. In *Proc. of the 48th International Colloquium on Automata, Languages, and Programming, (ICALP)*, volume 198 of *LIPIcs*, pages 35:1–35:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.35. 1

[30] Édouard Bonnet, Colin Geniet, Romain Tessera, and Stéphan Thomassé. Twin-width VII: groups, 2022. arXiv:2204.12330. 1

[31] Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Torunczyk. Twin-width IV: ordered graphs and matrices. In *Proc. of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 924–937. ACM, 2022. doi:10.1145/3519935.3520037. 1, 2

[32] Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, and Stéphan Thomassé. Twin-width VI: the lens of contraction sequences. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proc. of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1036–1056. SIAM, 2022. doi:10.1137/1.9781611977073.45. 1

[33] Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, Stéphan Thomassé, and Rémi Watrigant. Twin-width and polynomial kernels. In *Proc. of the 16th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 214 of *LIPIcs*, pages 10:1–10:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.IPEC.2021.10. 1

[34] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. *Journal of the ACM*, 69(1):3:1–3:46, 2022. doi:10.1145/3486655. 1, 7

[35] Édouard Bonnet, O-joung Kwon, and David R. Wood. Reduced bandwidth: a qualitative strengthening of twin-width in minor-closed classes (and beyond), 2022. arXiv:2202.11858. 1

[36] Édouard Bonnet, Jaroslav Nesetril, Patrice Ossona de Mendez, Sebastian Siebertz, and Stéphan Thomassé. Twin-width and permutations, 2021. arXiv:2102.06880. 1

[37] Paul Bonsma and Luis Cereceda. Finding paths between graph colourings: Pspace-completeness and superpolynomial distances. *Theoretical Computer Science*, 410(50):5215–5226, 2009. Mathematical Foundations of Computer Science (MFCS 2007). doi:10.1016/j.tcs.2009.08.023. 78

[38] Richard B. Borie, R. Gary Parker, and Craig A. Tovey. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica*, 7(5-6):555–581, 1992. doi:10.1007/BF01758777. 1

[39] Nicolas Bousquet, Tatsuhiko Hatanaka, Takehiro Ito, and Moritz Mühlenthaler. Shortest reconfiguration of matchings. In *Proc. of the 45th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 11789 of *Lecture Notes in Computer Science*, pages 162–174. Springer, 2019. doi:10.1007/978-3-030-30786-8_13. 78

[40] Nicolas Bousquet, Takehiro Ito, Yusuke Kobayashi, Haruka Mizuta, Paul Ouvrard, Akira Suzuki, and Kunihiro Wasa. Reconfiguration of spanning trees with many or few leaves. In *Proc. of the 28th Annual European Symposium on Algorithms (ESA)*, volume 173 of *LIPIcs*, pages 24:1–24:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.24. 78

[41] Nicolas Bousquet, Takehiro Ito, Yusuke Kobayashi, Haruka Mizuta, Paul Ouvrard, Akira Suzuki, and Kunihiro Wasa. Reconfiguration of Spanning Trees with Degree Constraint or Diameter Constraint. In *Proc. of the 39th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 219 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:21, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2022.15. 78

[42] Nicolas Bousquet and Alice Joffard. Ts-reconfiguration of dominating sets in circle and circular-arc graphs. In *Proc. of the 23rd International Symposium on Fundamentals of Computation Theory (FCT)*, pages 114–134. Springer International Publishing, 2021. doi:10.1007/978-3-030-86593-1_8. 78

[43] Nicolas Bousquet, Amer E. Mouawad, Naomi Nishimura, and Sebastian Siebertz. A survey on the parameterized complexity of the independent set and (connected) dominating set reconfiguration problems, 2022. arXiv:2204.10526. 78

[44] A. E. Brouwer and H. J. Veldman. Contractibility and NP-completeness. *Journal of Graph Theory*, 11(1):71–79, 1987. 5, 71

[45] Jannis Bulian and Anuj Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. *Algorithmica*, 75(2):363–382, 2016. doi:10.1007/s00453-015-0045-3. 77

[46] Jannis Bulian and Anuj Dawar. Fixed-parameter tractable distances to sparse graph classes. *Algorithmica*, 79(1):139–158, 2017. doi:10.1007/s00453-016-0235-7. 77

[47] Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Finding shortest non-trivial cycles in directed graphs on surfaces. *Journal of Computational Geometry*, 7(1):123–148, 2016. doi:10.20382/jocg.v7i1a7. 49

[48] Leizhen Cai, Siu Man Chan, and Siu On Chan. Random separation: A new method for solving fixed-cardinality optimization problems. In *Proc. of the 2nd International Workshop on Parameterized and Exact Computation (IPWEC)*, volume 4169 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2006. 5

[49] Kevin Cattell, Michael J. Dinneen, Rodney G. Downey, Michael R. Fellows, and Michael A. Langston. On computing graph minor obstruction sets. *Theoretical Computer Science*, 233:107–127, 2000. doi:10.1016/S0304-3975(97)00300-9. 4

[50] Rajesh Chitnis, Marek Cygan, Mohammad Taghi Hajiaghayi, Marcin Pilipczuk, and Michal Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. *SIAM Journal on Computing*, 45(4):1171–1229, 2016. doi:10.1137/15M1032077. 5

[51] Julia Chuzhoy. Improved Bounds for the Flat Wall Theorem. In *Proc. of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 256–275, 2015. doi:10.1137/1.9781611973730.20. 86

[52] Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H. 1, 6

[53] Bruno Courcelle. The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues. *RAIRO - Theoretical Informatics and Applications*, 26:257–286, 1992. doi:10.1051/ita/1992260302571. 6

[54] Bruno Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*, pages 313–400. World Scientific, 1997. 6

[55] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3. 1

[56] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3. 92

[57] Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, and Jakub Onufry Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. *SIAM Journal on Discrete Mathematics*, 27(1):290–309, 2013. doi:10.1137/110843071. 74

[58] Mateus de Oliveira Oliveira. Graph amalgamation under logical constraints. In *Proc. of the 44th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 11159 of *Lecture Notes in Computer Science*, pages 152–163. Springer, 2018. doi:10.1007/978-3-030-00256-5_13. 75

[59] Erik D. Demaine, Martin L. Demaine, Eli Fox-Epstein, Duc A. Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada. Linear-time algorithm for sliding tokens on trees. *Theoretical Computer Science*, 600:132–142, 2015. doi:10.1016/j.tcs.2015.07.037. 78

[60] Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *Journal of the ACM*, 52(6):866–893, 2005. doi:10.1145/1101821.1101823. 49

[61] Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. The bidimensional theory of bounded-genus graphs. In *Proc. of the 29th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 3153 of *Lecture Notes in Computer Science*, pages 191–203, 2004. doi:10.1007/978-3-540-28629-5_12. 49

[62] Erik D. Demaine and MohammadTaghi Hajiaghayi. Linearity of grid minors in treewidth with applications through bidimensionality. *Combinatorica*, 28(1):19–36, 2008. doi:10.1007/s00493-008-2140-4. 49

[63] Reinhard Diestel. *Graph Theory*, volume 173. Springer-Verlag, 5th edition, 2017. doi:10.1007/978-3-662-53622-3. 12

[64] Öznur Yaşar Diner, Archontia C. Giannopoulou, Giannos Stamoulis, and Dimitrios M. Thilikos. Block elimination distance. *Graphs and Combinatorics*, 38(5):133, 2022. doi:10.1007/s00373-022-02513-y. 78

[65] Gabriel Andrew Dirac. In abstrakten graphen vorhandene vollständige 4-graphen und ihre unterteilungen. *Mathematische Nachrichten*, 22(1-2):61–85, 1960. doi:10.1002/mana.19600220107. 72

[66] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1. 89

[67] Jan Dreier, Jakub Gajarský, Yiting Jiang, Patrice Ossona de Mendez, and Jean-Florent Raymond. Twin-width and generalized coloring numbers. *Discrete Mathematics*, 345(3):112746, 2022. doi:10.1016/j.disc.2021.112746. 1

[68] Jan Dreier and Peter Rossmanith. Approximate evaluation of first-order counting queries. In Dániel Marx, editor, *Proc. of the 32nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1720–1739. SIAM, 2021. doi:10.1137/1.9781611976465.104. 1, 2

[69] Zdenek Dvorák, Daniel Král, and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *Journal of the ACM*, 60(5):36:1–36:24, 2013. doi:10.1145/2499483. 1, 2

[70] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Perspectives in Mathematical Logic. Springer, 1995. 3

[71] Yoshimi Egawa, Ralph J. Faudree, Ervin Györi, Yoshiyasu Ishigami, Richard H. Schelp, and Hong Wang. Vertex-disjoint cycles containing specified edges. *Graphs and Combinatorics*, 16(1):81–92, 2000. doi:10.1007/s003730050005. 72

[72] Kord Eickmeyer, Michael Elberfeld, and Frederik Harwath. Succinctness of order-invariant logics on depth-bounded structures. *ACM Transactions on Computational Logic*, 18(4):33:1–33:25, 2017. doi:10.1145/3152770. 2

[73] M. N. Ellingham, Michael D. Plummer, and Gexin Yu. Linkage for the diamond and the path with four vertices. *Journal of Graph Theory*, 70(3):241–261, 2012. doi:10.1002/jgt.20612. 72

[74] Hiroshi Eto, Takehiro Ito, Yasuaki Kobayashi, Yota Otachi, and Kunihiro Wasa. Reconfiguration of regular induced subgraphs. In *Proc. of the 16th International Conference and Workshops on Algorithms and Computation (WALCOM)*, volume 13174 of *Lecture Notes in Computer Science*, pages 35–46. Springer, 2022. doi:10.1007/978-3-030-96731-4_4. 78

[75] Luerbio Faria, Celina M. Herrera de Figueiredo, and Candido F.X. Mendonça. On the complexity of the approximation of nonplanarity parameters for cubic graphs. *Discrete Applied Mathematics*, 141(1):119–134, 2004. doi:10.1016/S0166-218X(03)00370-6. 79

[76] Michael R. Fellows, Jan Kratochvíl, Matthias Middendorf, and Frank Pfeiffer. The complexity of induced minors and related problems. *Algorithmica*, 13(3):266–282, 1995. doi:10.1007/BF01190507. 5, 71

[77] Michael Ferrara, Ronald Gould, Gerard Tansey, and Thor Whalen. On H-linked graphs. *Graphs and Combinatorics*, 22(2):217–224, 2006. doi:10.1007/s00373-006-0651-6.

[78] Evelyne Flandrin, Hao Li, Antoni Marczyk, and Mariusz Woniak. A generalization of Dirac's Theorem on cycles through k vertices in k-connected graphs. *Discrete Mathematics*, 307(7–8):878–884, apr 2007. doi:10.1016/j.disc.2005.11.052. 72

[79] Jörg Flum and Martin Grohe. Fixed-parameter tractability, definability, and model-checking. *SIAM Journal on Computing*, 31(1):113–145, 2001. doi:10.1137/S0097539799360768. 10, 35

[80] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X. 1

[81] Fedor V. Fomin, Petr A. Golovach, Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. A compound logic for modification problems: Big kingdoms fall from within, 2021. arXiv:2111.02755. 2, 4, 10

[82] Fedor V. Fomin, Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. An algorithmic meta-theorem for graph modification to planarity and FOL. In *Proc. of the 28th Annual European Symposium on Algorithms (ESA)*, volume 173 of *LIPIcs*, pages 51:1–51:17, 2020. doi:10.4230/LIPIcs.ESA.2020.51. 4, 74

[83] Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Contraction obstructions for treewidth. *Journal of Combinatorial Theory, Series B*, 101(5):302–314, 2011. doi:10.1016/j.jctb.2011.02.008. 49

[84] Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Modification to Planarity is Fixed Parameter Tractable. In *Proc. of the 36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:17, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.STACS.2019.28. 4, 76, 77

[85] Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Parameterized complexity of elimination distance to first-order logic properties. In *Proc. of the 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, 2021. doi:10.1109/LICS52264.2021.9470540. 2, 77

[86] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar $\mathcal{F}$-Deletion: Approximation, Kernelization and Optimal FPT Algorithms. In *Proc. of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 470–479, 2012. doi:10.1109/FOCS.2012.62. 73

[87] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Hitting topological minors is FPT. In *Proc. of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1317–1326. ACM, 2020. doi:10.1145/3357713.3384318. 4, 5

[88] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Hitting topological minors is FPT. In *Proc. of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 1317–1326, 2020. doi:10.1145/3357713.3384318.

[89] Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Subexponential algorithms for partial cover problems. *Information Processing Letters*, 111(16):814–818, 2011. doi:10.1016/j.ipl.2011.05.016. 4

[90] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Linear kernels for (connected) dominating set on H-minor-free graphs. In *Proc. of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 82–93, 2012. doi:10.1137/1.9781611973099.7. 4

[91] Jakub Gajarský, Maximilian Gorsky, and Stephan Kreutzer. Differential games, locality, and model checking for FO logic of graphs. In *Proceedings of the 30th EACSL Annual Conference on Computer Science Logic (CSL)*, volume 216 of *LIPIcs*, pages 22:1–22:18, 2022. doi:https://doi.org/10.4230/LIPIcs.CSL.2022.22. 7

[92] Jakub Gajarský, Stephan Kreutzer, Jaroslav Nesetril, Patrice Ossona de Mendez, Michal Pilipczuk, Sebastian Siebertz, and Szymon Torunczyk. First-order interpretations of bounded expansion classes. *ACM Transactions on Computational Logic*, 21(4):29:1–29:41, 2020. doi:10.1145/3382093. 2

[93] Jakub Gajarský, Michal Pilipczuk, and Szymon Torunczyk. Stable graphs of bounded twin-width, 2021. arXiv:2107.03711. 1

[94] Robert Ganian, Petr Hlinený, Alexander Langer, Jan Obdrzálek, Peter Rossmanith, and Somnath Sikdar. Lower bounds on the complexity of mso_1 model-checking. *Journal of Computer and System Sciences*, 80(1):180–194, 2014. doi:10.1016/j.jcss.2013.07.005. 1

[95] Robert Ganian, Filip Pokrývka, André Schidler, Kirill Simonov, and Stefan Szeider. Weighted model counting with twin-width, 2022. arXiv:2206.01706. 1

[96] James F. Geelen, R. Bruce Richter, and Gelasio Salazar. Embedding grids in surfaces. *European Journal of Combinatorics*, 25(6):785–792, 2004. doi:10.1016/j.ejc.2003.07.007. 49

[97] Archontia C. Giannopoulou and Dimitrios M. Thilikos. Optimizing the graph minors weak structure theorem. *SIAM Journal on Discrete Mathematics*, 27(3):1209–1227, 2013. doi:10.1137/110857027. 86

[98] Tatsuya Gima, Takehiro Ito, Yasuaki Kobayashi, and Yota Otachi. Algorithmic meta-theorems for combinatorial reconfiguration revisited. In *Proc. of the 30th Annual European Symposium on Algorithms (ESA)*, volume 244 of *LIPIcs*, pages 61:1–61:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ESA.2022.61. 78

[99] Petr A. Golovach, Marcin Kaminski, Spyridon Maniatis, and Dimitrios M. Thilikos. The parameterized complexity of graph cyclability. *SIAM Journal on Discrete Mathematics*, 31(1):511–541, 2017. doi:10.1137/141000014. 4, 72

[100] Petr A. Golovach, Marcin Kamiński, Daniël Paulusma, and Dimitrios M. Thilikos. Induced packing of odd cycles in planar graphs. *Theoretical Computer Science*, 420:28–35, 2012. doi:https://doi.org/10.1016/j.tcs.2011.11.004. 4

[101] Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. Hitting topological minor models in planar graphs is fixed parameter tractable. In *Proc. of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 931–950, 2020. doi:10.1137/1.9781611975994.56. 4, 9, 25, 27

[102] Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. Hitting Topological Minor Models in Planar Graphs is Fixed Parameter Tractable, 2020. arXiv:1907.02919. 49

[103] Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. Combing a linkage in an annulus, 2022. arXiv:2207.04798. 4, 9, 12, 25, 27, 50

[104] Parikshit Gopalan, Phokion G. Kolaitis, Elitza Maneva, and Christos H. Papadimitriou. The connectivity of boolean satisfiability: Computational and structural dichotomies. *SIAM Journal on Computing*, 38(6):2330–2355, 2009. doi:10.1137/07070440X. 78

[105] Ronald Gould and Thor Whalen. Subdivision extendibility. *Graphs and Combinatorics*, 23(2):165–182, 2007. doi:10.1007/s00373-006-0665-0. 72

[106] Ronald J. Gould, Alexandr Kostochka, and Gexin Yu. On minimum degree implying that a graph is h-linked. *SIAM Journal on Discrete Mathematics*, 20(4):829–840, 2006. doi:10.1137/050624662. 72

[107] Erich Grädel, Wolfgang Thomas, and Thomas Wilke. *Automata, logics, and infinite games: a guide to current research*, volume 2500. Springer, 2003. 7

[108] Julien Grange. Successor-invariant first-order logic on classes of bounded degree. *Log. Methods Comput. Sci.*, 17(3), 2021. doi:10.46298/lmcs-17(3:20)2021. 2

[109] Martin Grohe. Logic, graphs, and algorithms. In *Logic and Automata: History and Perspectives, in Honor of Wolfgang Thomas*, volume 2 of *Texts in Logic and Games*, pages 357–422. Amsterdam University Press, 2008. URL: https://eccc.weizmann.ac.il/report/2007/091/. 1

[110] Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. In *Proc. of the 43rd ACM Symposium on Theory of Computing (STOC)*, pages 479–488. ACM, 2011. doi:10.1145/1993636.1993700. 4, 5, 71

[111] Martin Grohe and Stephan Kreutzer. Methods for algorithmic meta theorems. In *Model Theoretic Methods in Finite Combinatorics*, pages 181 – 206. Contemporary Mathematics,

2011. URL: https://www.cs.technion.ac.il/~janos/RESEARCH/AMS-Book-files/pdfs/05_GroheK.pdf. 1

[112] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM*, 64(3):17:1–17:32, 2017. doi:10.1145/3051095. 1

[113] Martin Grohe and Nicole Schweikardt. First-order query evaluation with cardinality conditions. In *Proc. of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 253–266. ACM, 2018. doi:10.1145/3196959.3196970. 2

[114] Martin Grohe and Thomas Schwentick. Locality of order-invariant first-order formulas. *ACM Transactions on Computational Logic*, 1(1):112–130, 2000. doi:10.1145/343369.343386. 2

[115] Jonathan L Gross. Genus distribution of graph amalgamations: self-pasting at root-vertices. *Australasian Journal of Combinatorics*, 49:19–38, 2011. URL: https://ajc.maths.uq.edu.au/pdf/49/ajc_v49_p019.pdf. 74

[116] Marshall Hall. An existence theorem for Latin squares. *Bulletin of the American Mathematical Society*, 51:387–388, 1945. doi:10.1090/S0002-9904-1945-08361-X. 90

[117] Robert A. Hearn and Erik D. Demaine. Pspace-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science*, 343(1):72–96, 2005. Game Theory Meets Theoretical Computer Science. doi:10.1016/j.tcs.2005.05.008. 78

[118] Pinar Heggernes, Pim Van 'T Hof, Daniel Lokshtanov, and Christophe Paul. Obtaining a bipartite graph by contracting few edges. *SIAM Journal on Discrete Mathematics*, 27(4):2143–2156, 2013. doi:10.1137/130907392. 4

[119] Jan van den Heuvel. *The complexity of change*, page 127–160. London Mathematical Society Lecture Note Series. Cambridge University Press, 2013. doi:10.1017/CBO9781139506748.005. 78

[120] A. J. W. Hilton, Matthew Johnson, C. A. Rodger, and E. B. Wantland. Amalgamations of connected k-factorizations. *Journal of Combinatorial Theory, Series B*, 88(2):267–279, jul 2003. doi:10.1016/S0095-8956(03)00030-3. 74

[121] Petr Hlinený and Filip Pokrývka. Twin-width and limits of tractability of FO model checking on geometric graphs, 2022. arXiv:2204.13742. 1

[122] Eva-Maria C. Hols, Stefan Kratsch, and Astrid Pieterse. Elimination Distances, Blocking Sets, and Kernels for Vertex Cover. In *Proc. of the 37th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 154 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:14, Dagstuhl, Germany, 2020. doi:10.4230/LIPIcs.STACS.2020.36. 77

[123] Ken ichi Kawarabayashi and Yusuke Kobayashi. Fixed-parameter tractability for the subset feedback set problem and the s-cycle packing problem. *Journal of Combinatorial Theory, Series B*, 102(4):1020–1034, 2012. doi:10.1016/j.jctb.2011.12.001. 74

[124] Takehiro Ito, Erik D. Demaine, Nicholas J.A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12):1054–1065, 2011. doi:10.1016/j.tcs.2010.12.005. 78

[125] Takehiro Ito, Marcin Kamiński, and Erik D. Demaine. Reconfiguration of list edge-colorings in a graph. In *Algorithms and Data Structures*, pages 375–386, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. doi:10.1007/978-3-642-03367-4_33. 78

[126] Takehiro Ito, Marcin Kamiński, Daniël Paulusma, and Dimitrios M. Thilikos. Parameterizing cut sets in a graph by the number of their components. *Theoretical Computer Science*, 412(45):6340–6350, 2011. doi:10.1016/j.tcs.2011.07.005. 4

[127] Hugo Jacob and Marcin Pilipczuk. Bounding twin-width for bounded-treewidth graphs, planar graphs, and bipartite graphs, 2022. arXiv:2201.09749. 1

[128] Bart M. P. Jansen and Jari J. H. de Kroon. FPT algorithms to compute the elimination distance to bipartite graphs and more. In *Proc. of the 47th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 12911 of *Lecture Notes in Computer Science*, pages 80–93. Springer, 2021. doi:10.1007/978-3-030-86838-3_6. 77

[129] Bart M. P. Jansen, Jari J. H. de Kroon, and Michal Włodarczyk. Vertex deletion parameterized by elimination distance and even less. In *Proc. of the 53rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 1757–1769, 2021. doi:10.1145/3406325.3451068. 4, 77

[130] Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A near-optimal planarization algorithm. In *Proc. of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1802–1811, 2014. doi:10.1137/1.9781611973402.130.

[131] H. A. Jung. Eine verallgemeinerung desn-fachen zusammenhangs für graphen. *Mathematische Annalen*, 187(2):95–103, 1970. doi:10.1007/BF01350174. 72

[132] Marcin Kamiński and Naomi Nishimura. Finding an induced path of given parity in planar graphs in polynomial time. In *Proc. of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 656–670. ACM, 2012. URL: https://dl.acm.org/doi/10.5555/2095116.2095171. 4

[133] Marcin Kamiński and Dimitrios M. Thilikos. Contraction checking in graphs on surfaces. In *Proc. of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 182–193, 2012. doi:10.4230/LIPIcs.STACS.2012.182. 4, 71, 73

[134] Ken-ichi Kawarabayashi. Half integral packing, Erdős-Pósa-property and graph minors. In *Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1187–1196. Society for Industrial and Applied Mathematics, 2007. URL: http://dl.acm.org/citation.cfm?id=1283383.1283511. 4

[135] Ken-ichi Kawarabayashi. Planarity allowing few error vertices in linear time. In *Proc. of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 639–648, 2009. doi:10.1109/FOCS.2009.45. 4

[136] Ken-ichi Kawarabayashi and Yusuke Kobayashi. The induced disjoint paths problem. In *Proc. of the 13th International Conference on Integer Programming and Combinatorial Optimization*

(IPCO), volume 5035 of *Lecture Notes in Computer Science*, pages 47–61. Springer, 2008. doi:10.1007/978-3-540-68891-4_4. 5, 71

[137] Ken-ichi Kawarabayashi and Yusuke Kobayashi. An improved algorithm for the half-disjoint paths problem. *SIAM Journal on Discrete Mathematics*, 25(3):1322–1330, 2011. doi:10.1137/100808812. 4

[138] Ken-ichi Kawarabayashi and Yusuke Kobayashi. A linear time algorithm for the induced disjoint paths problem in planar graphs. *Journal of Computer and System Sciences*, 78(2):670–680, 2012. doi:10.1016/j.jcss.2011.10.004. 12

[139] Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce A. Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102(2):424–435, 2012. doi:10.1016/j.jctb.2011.07.004. 4, 5

[140] Ken-ichi Kawarabayashi, Alexandr Kostochka, and Gexin Yu. On sufficient degree conditions for a graph to be k-linked. *Combinatorics, Probability and Computing*, 15(5):685–694, 2006. doi:10.1017/S0963548305007479.

[141] Ken-ichi Kawarabayashi, Stephan Kreutzer, and Bojan Mohar. Linkless and flat embeddings in 3-space and the unknot problem. In *Proc. of the 2010 Annual Symposium on Computational Geometry (SoCG)*, pages 97–106. ACM, 2010. doi:10.1145/1810959.1810975. 4

[142] Ken-ichi Kawarabayashi, Zhentao Li, and Bruce A. Reed. Recognizing a totally odd K_4-subdivision, parity 2-disjoint rooted paths and a parity cycle through specified elements. In *Proc. of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 318–328, 2010. URL: https://dl.acm.org/doi/10.5555/1873601.1873628. 4

[143] Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In *Proc. of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 771–780, 2008. doi:10.1109/FOCS.2008.53. 4

[144] Ken-ichi Kawarabayashi and Bruce A. Reed. Hadwiger's conjecture is decidable. In *Proc. of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 445–454, 2009. URL: http://doi.acm.org/10.1145/1536414.1536476. 4

[145] Ken-ichi Kawarabayashi and Bruce A. Reed. Odd cycle packing. In *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 695–704, 2010. doi:10.1145/1806689.1806785. 4

[146] Ken-ichi Kawarabayashi, Robin Thomas, and Paul Wollan. A new proof of the flat wall theorem. *Journal of Combinatorial Theory, Series B*, 129:204–238, 2018. doi:10.1016/j.jctb.2017.09.006. 8, 9, 82, 83, 86

[147] Ken-ichi Kawarabayashi and Paul Wollan. A shorter proof of the graph minor algorithm: The unique linkage theorem. In *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC)*, page 687–694, New York, NY, USA, 2010. ACM. doi:10.1145/1806689.1806784. 4

[148] Ken-ichi Kawarabayashi and Paul Wollan. A shorter proof of the graph minor algorithm: the unique linkage theorem. In *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 687–694, 2010. doi:10.1145/1806689.1806784. 47

[149] Yusuke Kobayashi and Ken-ichi Kawarabayashi. Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In *Proc. of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1146–1155. ACM-SIAM, 2009. URL: https://dl.acm.org/doi/10.5555/1496770.1496894. 4

[150] Tomasz Kociumaka and Marcin Pilipczuk. Deleting vertices to graphs of bounded genus. *Algorithmica*, 81(9):3655–3691, 2019. doi:10.1007/s00453-019-00592-7. 73

[151] Alexandr Kostochka and Gexin Yu. An extremal problem for h-linked graphs. *Journal of Graph Theory*, 50(4):321–339, 2005. doi:10.1002/jgt.20115. 72

[152] Stefan Kratsch, Florian Nelles, and Alexandre Simon. On triangle counting parameterized by twin-width, 2022. arXiv:2202.06708. 1

[153] Stephan Kreutzer. Algorithmic meta-theorems. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:147, 2009. URL: https://www.cs.ox.ac.uk/people/stephan.kreutzer/Publications/amt-survey.pdf. 1

[154] Stephan Kreutzer and Anuj Dawar. Parameterized complexity of first-order logic. *Electronic Colloquium on Computational Complexity*, page 131, 2009. URL: https://eccc.weizmann.ac.il/report/2009/131. 1

[155] Stephan Kreutzer, Roman Rabinovich, and Sebastian Siebertz. Polynomial kernels and wideness properties of nowhere dense graph classes. *ACM Transactions on Algorithms*, 15(2):24:1–24:19, 2019. doi:10.1145/3274652. 1

[156] Stephan Kreutzer and Siamak Tazari. Lower bounds for the complexity of monadic second-order logic. In *Proc. of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 189–198. IEEE Computer Society, 2010. doi:10.1109/LICS.2010.39. 1

[157] Dietrich Kuske and Nicole Schweikardt. First-order logic with counting. In *Proc. of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005133. 2

[158] Dietrich Kuske and Nicole Schweikardt. Gaifman normal forms for counting extensions of first-order logic. In *Proc. of the 45th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107 of *LIPIcs*, pages 133:1–133:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.133. 2

[159] D. G. Larman and P. Mani. On the existence of certain configurations within graphs and the 1-skeletons of polytopes. *Proceedings of the London Mathematical Society*, s3-20(1):144–160, 1970. doi:https://doi.org/10.1112/plms/s3-20.1.144. 72

[160] C. D. Leach and C. A. Rodger. Hamilton decompositions of complete multipartite graphs with any 2-factor leave. *Journal of Graph Theory*, 44(3):208–214, 2003. doi:10.1002/jgt.10142. 74

[161] Benjamin Lévêque, David Y. Lin, Frédéric Maffray, and Nicolas Trotignon. Detecting induced subgraphs. *Discrete Applied Mathematics*, 157(17):3540–3551, 2009. doi:10.1016/j.dam.2009.02.015. 5, 71

[162] Asaf Levin, Daniël Paulusma, and Gerhard J. Woeginger. The computational complexity of graph contractions I: Polynomially solvable and NP-complete cases. *Networks*, 51(3):178–189, 2008. doi:10.1002/net.20214. 5, 71

[163] Asaf Levin, Daniël Paulusma, and Gerhard J. Woeginger. The computational complexity of graph contractions II: Two tough polynomially solvable cases. *Networks*, 52(1):32–56, 2008. doi:10.1002/net.20249. 5, 71

[164] Leonid Libkin. *Elements of Finite Model Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004. URL: http://www.cs.toronto.edu/%7Elibkin/fmt, doi:10.1007/978-3-662-07003-1. 3

[165] Bingkai Lin. Constant approximating k-clique is w[1]-hard. In *Proc. of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1749–1756. ACM, 2021. doi:10.1145/3406325.3451016. 89, 91

[166] Alexander Lindermayr, Sebastian Siebertz, and Alexandre Vigny. Elimination distance to bounded degree on planar graphs. In *Proc. of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 170 of *LIPIcs*, pages 65:1–65:12, 2020. doi:10.4230/LIPIcs.MFCS.2020.65. 4, 77

[167] Daniel Lokshtanov, Amer E. Mouawad, Fahad Panolan, M.S. Ramanujan, and Saket Saurabh. Reconfiguration on sparse graphs. *Journal of Computer and System Sciences*, 95:122–131, 2018. doi:https://doi.org/10.1016/j.jcss.2018.02.004. 78, 79

[168] Daniel Lokshtanov, Amer E. Mouawad, Fahad Panolan, and Sebastian Siebertz. On the parameterized complexity of reconfiguration of connected dominating sets. *Algorithmica*, 84(2):482–509, 2022. doi:10.1007/s00453-021-00909-5. 78

[169] Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Reducing CMSO Model Checking to Highly Connected Graphs. In *Proc. of the 45th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 135:1–135:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2018.135. 5

[170] Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012. doi:10.1007/s00453-010-9484-z. 4, 73

[171] Frédéric Mazoit. A single exponential bound for the redundant vertex theorem on surfaces, 2013. arXiv:1309.7820. 4, 12, 47

[172] Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, and Sebastian Siebertz. Vertex cover reconfiguration and beyond. *Algorithms*, 11(2):20, 2018. doi:10.3390/a11020020. 78

[173] Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, 78(1):274–297, 2017. doi:10.1007/s00453-016-0159-2. 78

[174] Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, and Marcin Wrochna. Reconfiguration over tree decompositions. In *Parameterized and Exact Computation*, pages 246–257. Springer International Publishing, 2014. doi:10.1007/978-3-319-13524-3_21. 78

[175] Mouawad, Amer. *On Reconfiguration Problems: Structure and Tractability.* PhD thesis, University of Waterloo, 2015. URL: http://hdl.handle.net/10012/9183. 78

[176] C. M. Mynhardt and S. Nasserasr. Reconfiguration of colourings and dominating sets in graphs: a survey, 2020. arXiv:2003.05956. 78

[177] Jaroslav Nesetril and Patrice Ossona de Mendez. The grad of a graph and classes with bounded expansion. *Electronic Notes in Discrete Mathematics*, 22:101–106, 2005. doi:10.1016/j.endm.2005.06.018. 1

[178] Jaroslav Nesetril and Patrice Ossona de Mendez. On nowhere dense graphs. *European Journal of Combinatorics*, 32(4):600–617, 2011. doi:10.1016/j.ejc.2011.01.006. 1

[179] Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4. 1

[180] Jaroslav Nesetril, Patrice Ossona de Mendez, Michal Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. Rankwidth meets stability. In *Proc. of the 32nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2014–2033. SIAM, 2021. doi:10.1137/1.9781611976465.120. 2

[181] Jaroslav Nesetril, Patrice Ossona de Mendez, and Sebastian Siebertz. Structural properties of the first-order transduction quasiorder. In *Proc. of the 30th EACSL Annual Conference on Computer Science Logic (CSL)*, volume 216 of *LIPIcs*, pages 31:1–31:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CSL.2022.31. 2

[182] Jaroslav Nesetril, Roman Rabinovich, Patrice Ossona de Mendez, and Sebastian Siebertz. Linear rankwidth meets stability. In *Proc. of the 31st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1180–1199. SIAM, 2020. doi:10.1137/1.9781611975994.72. 2

[183] J. Nešetřil. Amalgamation of graphs and its applications. *Annals of the New York Academy of Sciences*, 319(1):415–428, 1979. doi:10.1111/j.1749-6632.1979.tb32819.x. 74

[184] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006. doi:10.1093/ACPROF:OSO/9780198566076.001.0001. 1

[185] Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4), 2018. doi:10.3390/a11040052. 78

[186] William Pettersson and John Sylvester. Bounds on the twin-width of product graphs, 2022. arXiv:2202.11556. 1

[187] Michal Pilipczuk, Patrice Ossona de Mendez, and Sebastian Siebertz. Transducing paths in graph classes with unbounded shrubdepth, 2022. arXiv:2203.16900. 2

[188] Michal Pilipczuk, Nicole Schirrmacher, Sebastian Siebertz, Szymon Torunczyk, and Alexandre Vigny. Algorithms and data structures for first-order logic with connectivity under vertex failures. In *Proc. of the 49th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 229 of *LIPIcs*, pages 102:1–102:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.102. 2, 52, 77, 78

[189] Michal Pilipczuk, Sebastian Siebertz, and Szymon Torunczyk. On the number of types in sparse graphs. In *Proc. of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 799–808. ACM, 2018. doi:10.1145/3209108.3209178. 1

[190] Michal Pilipczuk and Marek Sokolowski. Graphs of bounded twin-width are quasi-polynomially χ-bounded, 2022. arXiv:2202.07608. 1

[191] Michal Pilipczuk, Marek Sokolowski, and Anna Zych-Pawlewicz. Compact representation for matrices of bounded twin-width. In *Proc. of the 39th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 219 of *LIPIcs*, pages 52:1–52:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.STACS.2022.52. 1

[192] Michael D. Plummer and Ervin Győri. A nine vertex theorem for 3-connected claw- free graphs. *Studia Scientiarum Mathematicarum Hungarica*, 38:233–244, 2001. doi:10.1556/sscmath.38.2001.1-4.16. 72

[193] Neil Robertson and Paul D. Seymour. Graph minors . XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995. doi:10.1006/jctb.1995.1006. 3, 4, 5, 8, 71, 72, 82, 83, 86

[194] Neil Robertson and Paul D. Seymour. Graph minors. XXI. Graphs with unique linkages. *Journal of Combinatorial Theory, Series B*, 99(3):583–616, 2009. doi:10.1016/j.jctb.2008.08.003. 47

[195] Neil Robertson and Paul D. Seymour. Graph Minors. XXII. Irrelevant vertices in linkage problems. *Journal of Combinatorial Theory, Series B*, 102(2):530–563, 2012. doi:10.1016/j.jctb.2007.12.007. 47

[196] Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. An FPT-Algorithm for Recognizing k-Apices of Minor-Closed Graph Classes. In *Proc. of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 168 of *LIPIcs*, pages 95:1–95:20, 2020. doi:10.4230/LIPIcs.ICALP.2020.95. 4

[197] Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. A more accurate view of the Flat Wall Theorem, 2021. arXiv:2102.06463. 4, 6, 8, 9, 81, 82, 83, 84, 85, 86, 87

[198] Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. k-apices of minor-closed graph classes. I. Bounding the obstructions, 2021. arXiv:2103.00882. 4, 85

[199] Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. k-apices of minor-closed graph classes. II. Parameterized algorithms. *ACM Transactions on Algorithms*, 18(3), 2022. doi:10.1145/3519028. 4, 73

[200] André Schidler and Stefan Szeider. A SAT approach to twin-width. In *Proc. of the Symposium on Algorithm Engineering and Experiments (ALENEX21)*, pages 67–77. SIAM, 2022. doi:10.1137/1.9781611977042.6. 1

[201] Nicole Schirrmacher, Sebastian Siebertz, and Alexandre Vigny. First-Order Logic with Connectivity Operators. In *Proc. of the 30th EACSL Annual Conference on Computer Science Logic (CSL)*, volume 216 of *LIPIcs*, pages 34:1–34:17, 2022. doi:10.4230/LIPIcs.CSL.2022.34. 2, 52, 77

[202] Sebastian Siebertz. Reconfiguration on nowhere dense graph classes. *The Electronic Journal of Combinatorics*, 25(3):3, 2018. doi:10.37236/7458. 78

[203] Dimitrios M. Thilikos. Graph minors and parameterized algorithm design. In *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, pages 228–256, 2012. URL: https://dx.doi.org/10.1007/978-3-642-30891-8_13. 4

[204] Robin Thomas and Paul Wollan. An improved linear edge bound for graph linkages. *European Journal of Combinatorics*, 26(3):309–324, 2005. Topological Graph Theory and Graph Minors, second issue. doi:10.1016/j.ejc.2004.02.013. 72

[205] Paul Turán. Eine Extremalaufgabe aus der Graphentheorie. *Matematikai és Fizikai Lapok*, 48:436–452, 1941. 91

[206] Jan van den Heuvel, Stephan Kreutzer, Michal Pilipczuk, Daniel A. Quiroz, Roman Rabinovich, and Sebastian Siebertz. Model-checking for successor-invariant first-order formulas on graph classes of bounded expansion. In *Proc. of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–11. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005115. 2

[207] M. Watkins and D. Mesner. Cycles and connectivity in graphs. *Canadian Journal of Mathematics*, 19:1319–1328, 1967. doi:10.4153/cjm-1967-121-2. 72

[208] Mark E. Watkins. On the existence of certain disjoint arcs in graphs. *Duke Mathematical Journal*, 35(2):231 – 246, 1968. doi:10.1215/S0012-7094-68-03523-0. 72

[209] Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *Journal of Computer and System Sciences*, 93:1–10, 2018. doi:10.1016/j.jcss.2017.11.003. 78

[210] Takeshi Yamada and Ryuhei Uehara. Shortest reconfiguration of sliding tokens on subclasses of interval graphs. *Theoretical Computer Science*, 863:53–68, 2021. doi:10.1016/j.tcs.2021.02.019. 78

[211] Yan Yang and Yichao Chen. The thickness of amalgamations and cartesian product of graphs. *Discussiones Mathematicae Graph Theory*, 37(3):561–572, 2017. doi:doi:10.7151/dmgt.1942. 74

A Problems expressible in FOL+DP and in FOL+SDP

In this section we present some (families) of parameterized problems where Theorem 1 and Theorem 2 are applicable. In all cases, we consider the standard parameterization (by the integer k of the input). Also for each (meta)-problem we comment on their general parameterized complexity status, their possible classification on the families A, B, and C, given in the introduction.

First of all observe that DISJOINT PATHS is the prototypical problem of this category as it is expressed by the "trivial" sentence $\varphi_k = \mathsf{dp}_k(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$. This problem is FPT because of [193]. If instead we consider the INDUCED DISJOINT PATHS, that is $\varphi_k = \mathsf{sdp}_k(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$, then the corresponding parameterized problem becomes para-NP-hard because checking whether $(G, \mathsf{s}_1, \mathsf{t}_1, \mathsf{s}_2, \mathsf{t}_2) \models \varphi_2$ is already NP-complete [136]. More generaly, we may also consider problems where the input graph is accompanied with a fixed number of colors $X_1, \ldots, X_z$, i.e., we consider structures of the form $(G, X_1, \ldots, X_z)$, and we may consider the more general predicates $(\mathsf{s})\mathsf{dp}_{k,\lambda}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$ equiped with a list function $\lambda : [k] \to 2^{[z]}$, where we demand that, for every $i \in [k]$, $\lambda(i)$ is a subset of the set of colors of the vertices of the path between the (valuations) of s_1 and t_1. For instance we may demand that all disjoint paths are *colorful*, i.e., they contain vertices of all available colors.

A.1 Graph containment problems

We first consider families of problems expressible by some $\varphi_k = \exists \overline{\mathsf{x}}\, \psi(\overline{\mathsf{x}}) \in$ FOL+SDP. Such problems are defined by some partial ordering relation $\preceq$ on graphs. We say that H is a *contraction* of G if H can be obtained from G after contracting edges, H is an (*induced*) *minor* of G if H is the contraction of an (induced) subgraph of G. Finally we say that H is an (*induced*) *topological minor* of G if G contains a subdivision of H is an (induced) subgaph.

The general setting is the following.

$\preceq$-CONTRAINMENT
Input: two graphs G and H where $k = |H|$.
Question: $H \preceq G$?

Notice that if $\preceq$ is the minor or the topological minor relation, $\preceq$-CONTRAINMENT is definable in FOL+DP and yields MINOR CONTAINMENT and TOPOLOGICAL MINOR CONTAINMENT respectively. MINOR CONTAINMENT belongs in the Category A as it is FPT in general graphs because no-instances are trivially excluding a $K_{|H|}$ minor. TOPOLOGICAL MINOR CONTAINMENT belongs in the category C as it its FPT in general graphs however to deal with the question "*what to do with a clique*" it needs extra arguments [110]. In the case where $\preceq$ is the induced minor or the induced topological minor relation, $\preceq$-CONTRAINMENT is definable in FOL+SDP and Theorem 2 yields that INDUCED MINOR and INDUCED TOPOLOGICAL MINOR CONTAINMENT are FPT on bounded genus graphs. Moreover as observed in [133], it is possible ro reduce the CONTRACTION CONTAINMENT problem on bounded genus graphs to TOPOLOGICAL MINOR CONTAINMENT. These last three problems belong in category C because they are all NP-hard for particular instantiations of H due to the results of [44, 76, 161–163] (using the parameterized complexity terminology, their standard parmeterizatins are para-NP-hard). Certain rooted variants of all these problems can be also expressed by the corresponding logics if we ask that the "models" certifying each of the aforementioned relations meet certain vertices or sets of vertices (colors) of the input graph.

A.2 Linkability problems

We now consider families of problems expressible by some $\varphi_k = \forall \mathsf{x}\, \psi(\bar{\mathsf{x}}) \in \mathsf{FOL{+}SDP}$. Such problems involve disjoint path queries for every choice of terminals in the graph.

Unordered Linkability problems. Given a graph G, a set $R \subseteq V(G)$, and a $k \in \mathbb{N}$, we say that R is *k-cyclable* in G if every k vertices of R belong in some cycle of G (see [8, 65, 78, 192, 207] for the combinatorial properties of k-cyclable sets). The CYCLABILITY problem asks, given a triple G, R, and k as above, whether R is k-cyclable in G. The algorithmic properties of CYCLABILITY have been studied in [99] where it was proven that the standard parametrization of CYCLABILITY is FPT for planar graphs, while the general problem is co-W[1]-hard.

For this, given a graph H, we say that R is *H-linkable* if for every subset $S \in \binom{R}{k}$ there is a $\mathcal{P} \in \binom{S}{2}$ and a collection of internally vertex disjoint paths in G between the pairs in $\mathcal{P}$ that, when contracted to single edges, give a graph that is isomorphic to H. We now consider the following problem.

> UNORDERED LINKABILITY
> *Input*: a graph G, $R \subseteq V(G)$, and a graph H where $k = |H|$.
> *Question*: is R H-linkable in G?

Notice that above problem is expressible by a sentence $\varphi_H = \forall \mathsf{x}\ \psi(\bar{\mathsf{x}})$, where ψ consists of $k!$ disjunctions of the $\mathsf{dp}_{|E(H)|}$ predicate. Moreover, when H is a cycle of k vertices the above problem yields the CYCLABILITY problem, that is already co-W[1]-hard. Theorem 1 automatically implies that the standard parameterization of UNORDERED LINKABILITY is FPT, when restricted to graphs of bounded Hadwiger number.

Ordered Linkability problems. Let G be a graph and let $R \subseteq V(G)$. Given a $k \in \mathbb{N}$ we say that *R is k-linked in G* if for every (ordered) set $\{s_1, \ldots, s_k, t_1, \ldots, t_k\}$ of $2k$ distinct vertices in R there are k vertex disjoint paths in G joining the pairs $(s_i, t_i), i \in [k]$. This notion has introduced by [208] in the late 60s and its graph-theoretical properties have been extensively studied in [23, 71, 131, 140, 159, 193]. For instance, Thomas and Wollan proved in [204] that if a graph G is $10k$-connected, then $V(G)$ is k-linked in G. This notion has been extended to the one of a *H-linked set* as follows: given some graph H with $V(H) = \{x_1, \ldots, x_k\}$, we say that G is *H-linked in* R if, *for every sequence* $v_1, \ldots, v_k$ of vertices in R there is a $\mathcal{P} \in \binom{\{v_1, \ldots, v_k\}}{2}$ and a collection of internally vertex disjoint paths in G between the pairs in $\mathcal{P}$ that, when contracted to single edges, give a graph with vertex set S that is isomorphic to H via the isomorphism that maps x_i to v_i, $i \in [k]$. The combinatorics of H-linked sets has been studied in [73, 77, 105, 106, 151]. However, to our knowledge, nothing is known about the algorithmic properties of k-linked sets or the more general concept of H-linked sets. For this, we consider the following general problem.

> ORDERED LINKABILITY
> *Input*: a graph G, $R \subseteq V(G)$, and a graph H where $k = |H|$.
> *Question*: is R H-linked in G?

It is easy to verify that the above problem is expressible by a sentence $\varphi_H = \forall \mathsf{x}\ \psi(\bar{\mathsf{x}})$, for some suitable choice of the quantification-free formula ψ. In Subsection C.1 we prove that the ORDERED LINKABILITY problem, even for the case where H is the disjoint union of k edges, is not FPT, unless $\mathsf{FPT} = \mathsf{W}[1]$ (Theorem 5). Theorem 1 automatically implies that the standard parameterization of ORDERED LINKABILITY is FPT, when restricted to graphs of bounded Hadwiger number.

Clearly both Unordered Linkability and Ordered Linkability belong category B and it is an open question whether it is FPT for other graph classes more general (or different) than those of bounded Hadwiger number. Notice that we may further consider that the disjoint paths in the definition of being H-linked and H-linkable are *induced* paths. This would define the Induced Unordered Linkability and Induced Ordered Linkability problems whose standard parameterizations are not expected to be FPT (by easy reductions from the problems Cyclability and Ordered Linkability) and they are FPT in bounded genus graphs, because of Theorem 2.

A.3 Vertex deletion problems

We give now a wide variety of $\mathsf{FOL}+\mathsf{SDP}$-expressible problems, typically correspond to formulas of the type $\varphi_k = \exists \bar{\mathsf{x}}_1 \forall \bar{\mathsf{x}}_2\ \psi(\bar{\mathsf{x}}_1, \bar{\mathsf{x}}_2)$. We present below those that we consider more relevant.

Vertex deletion to exclusion. This family of problems can be seen as a natural extension of those mentioned in Subsection A.1. Let $\preceq$ be a partial ordering relation on graphs and let $\mathcal{F}$ be a finite set of graphs. Given a graph G, we say that $\mathcal{F} \preceq G$ if for some $F \in \mathcal{F}$ it holds that $F \preceq G$. We define the following meta-problem.

> $\mathcal{F}$-$\preceq$-Deletion
> *Input*: a graph G and a $k \in \mathbb{N}$.
> *Question*: is there an $S \in \binom{V(G)}{k}$ such that $\mathcal{F} \npreceq G \setminus S$?

Notice that the above yields the problems $\mathcal{F}$-Minor-Deletion and $\mathcal{F}$-Topological Minor-Deletion are expressible by some $\varphi_k = \exists \mathsf{x}_1, \ldots, \mathsf{x}_k\ \neg\xi(\mathsf{x}_1, \ldots, \mathsf{x}_k)$ where $\xi(\mathsf{x}_1, \ldots, \mathsf{x}_k)$ is obtained if, in the corresponding sentences of Subsection A.1, we replace every $\mathsf{dp}_k(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k)$ by $\mathsf{dp}_k(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_k, \mathsf{t}_k, \mathsf{x}_1, \mathsf{x}_1, \cdots, \mathsf{x}_k, \mathsf{x}_k)$. It is easy to see that yes-instance of $\mathcal{F}$-Minor-Deletion have bounded Hadwiger number, therefore it belongs in Category A. The standard parameterization of $\mathcal{F}$-Minor-Deletion is known to be FPT in general because of [86, 130, 150, 170, 199]. On the other hand, the standard parameterization of $\mathcal{F}$-Topological Minor Deletion was proved to be FPT in general [88] and belongs to Category C, as extra machinery was used for 1st phase of the irrelevant vertex technique.

By applying Theorem 1 on bounded genus graphs and combining it with the duality trick of [133] it is possible to give an FPT-reduction of the standard parameterization of $\mathcal{F}$-Contraction Deletion on bounded genus graphs to the standard parameterization of $\mathcal{F}$-Topological Minor Deletion. Furthermore, $\mathcal{F}$-Induced Minor Deletion and $\mathcal{F}$-Induced Topological Minor Deletion are expressible in $\mathsf{FOL}+\mathsf{SDP}$ using sdp_k instead of dp_k in the above expressibility argument. Therefore both parameterized problems are in FPT on graphs of bounded genus because of Theorem 2. It is easy to see that these last three parameterized problems are para-NP-hard in general, therefore they are classified in Category B.

Annotation and subset variants. Another direction is to consider Annotated $\mathcal{F}$-$\preceq$-Deletion where the input comes with an annotated set of vertices $R \subseteq V(G)$ and we further demand that $S \subseteq R$. Another variant is the Subset $\mathcal{F}$-$\preceq$-Deletion where again the input comes an annotated set R of vertices but now we ask that for every $Z \subseteq V(G)$ where $\mathcal{F} \preceq G[Z]$ and $Z \cap R \neq \emptyset$, it holds that $Z \cap S \neq \emptyset$. Intuitively, we demand that *only* the certificates of the containment of a graph in $\mathcal{F}$ that intersect R are required to be intersected by the solution S.

All results mentioned above for the five aforementioned partial relations on graphs hold also for the corresponding Annotated $\mathcal{F}$-$\preceq$-Deletion and the Subset $\mathcal{F}$-$\preceq$-Deletion meta-problems when restricted either to graphs of bounded Hadwiger number or of bounded genus. For general graphs, the Subset $\mathcal{F}$-Minor-Deletion problem has been treated in [57, 123] for $\mathcal{F} = \{K_3\}$.

General vertex deletion. Deviating from the $\exists\,\forall$ scheme of this subsection we wish to mention that all above problems can be seen as *modification* problems where we want to achieve some particular target property by removing vertices. The target property is the exclusion of some pattern graphs (the graphs in $\mathcal{F}$) under some partial relation (the relation $\preceq$). The application of Theorem 1 and Theorem 2 was made possible because because the $\preceq$-exclusion is, depending on the choice of $\preceq$, either $\mathsf{FOL{+}DP}$ or $\mathsf{FOL{+}SDP}$-expressible. In fact we can see all above results as special cases of the following meta-problem defined given a $\varphi \in \mathsf{FOL{+}SDP}$.

φ-Deletion
Input: a graph G and a $k \in \mathbb{N}$.
Question: is there an $S \in \binom{V(G)}{k}$ such that $G \setminus S \models \varphi$?

According to Theorem 1 and Theorem 2 if $\varphi \in \mathsf{FOL{+}DP}$ (resp. $\varphi \in \mathsf{FOL{+}SDP}$), then φ-Deletion is FPT on graphs of bounded Hadwiger number (resp. bounded genus). The special case where φ corresponds to the property of being planar and satisfying some FOL property is treated in [82].

A.4 Amalgamation problems

The input of an amalgamation problem consists of two graphs and asks for a way to identify their vertices so that the new graph satisfies some particular property. The notion of amalgamation dates back to [183] and its combinatorial study includes [115, 120, 160, 211].

Given two graphs G_1 and G_2 we define $G_1 \otimes_k G_2$ as the set containing every graph obtained if for some $S_i \in \binom{V(G_i)}{k}, i \in [2]$ and a bijection $\sigma : S_1 \to S_2$ we take the disjoint union of G_1 and G_2 and then identify each vertex $v \in S_1$ with $\sigma(v)$. Consider the following problem, defined given a $\varphi \in \mathsf{FOL{+}DP}$ ($\varphi \in \mathsf{FOL{+}SDP}$).

φ-Amalgamation
Input: two graphs G_1, G_2 and a $k \in \mathbb{N}$.
Question: is there a graph in $J \in G_1 \otimes_k G_2$ where $J \models \varphi$?

Expressing φ-Amalgamation. Let $\varphi \in \mathsf{FOL{+}DP}$ (resp. $\varphi \in \mathsf{FOL{+}SDP}$). We claim that there is a sentence $\chi_k \in \mathsf{FOL{+}DP}$ ($\chi_k \in \mathsf{FOL{+}SDP}$) such that (G_1, G_2, k) is a yes-instance of φ-Amalgamation iff the tuple $(G, V_1, V_2) \models \chi_k$ where G is the disjoint union of G_1 and G_2 and $V_i = V(G_i), i \in [2]$. To see this, consider the sentence

$$\chi_k = \exists \mathsf{v}_1^1, \ldots, \mathsf{v}_k^1, \mathsf{v}_1^2, \ldots, \mathsf{v}_k^2 \; \Big(\bigwedge_{i \in [k], j \in [2]} \mathsf{v}_i^j \in V_j \wedge \varphi^\star(\mathsf{v}_1^1, \ldots, \mathsf{v}_k^1, \mathsf{v}_1^2, \ldots, \mathsf{v}_k^2) \Big),$$

where $\varphi^\star$ is a formula with $\mathsf{v}_1^1, \ldots, \mathsf{v}_k^1, \mathsf{v}_1^2, \ldots, \mathsf{v}_k^2$ as free variables obtained from φ after replacing each of its atomic formulas as follows:

- Each atomic formula $\mathsf{x} = \mathsf{y}$, is replaced by the formula $\zeta_=(\mathsf{x}, \mathsf{y})$, defined as

$$(\mathsf{x} = \mathsf{y}) \vee \bigvee_{i \in [k]} \big((\mathsf{x} = \mathsf{v}_i^1 \wedge \mathsf{y} = \mathsf{v}_i^2) \vee (\mathsf{y} = \mathsf{v}_i^1 \wedge \mathsf{x} = \mathsf{v}_i^2) \big).$$

- Each atomic formula $\mathsf{E}(\mathsf{x},\mathsf{y})$ is replaced by the formula $\zeta_{\mathsf{E}}(\mathsf{x},\mathsf{y})$, defined as

$$\mathsf{E}(\mathsf{x},\mathsf{y}) \vee \bigvee_{i\in[k]} \Big(\big(\mathsf{x} =^{\star} \mathsf{v}_i^1 \wedge \mathsf{E}(\mathsf{v}_i^2,\mathsf{y})\big) \vee \big(\mathsf{y} =^{\star} \mathsf{v}_i^1 \wedge \mathsf{E}(\mathsf{x},\mathsf{v}_i^2)\big)\Big),$$

- Each atomic formula $(\mathsf{s})\mathsf{dp}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_t,\mathsf{t}_t)$ is replaced by the formula $\zeta_{(\mathsf{s})\mathsf{dp}}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_t,\mathsf{t}_t)$ defined in Subsection 6.1, after doing some local replacements in $\zeta_{(\mathsf{s})\mathsf{dp}}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_t,\mathsf{t}_t)$ as follows.
  - replace every atomic formula of the form "$\mathsf{y}_j \in \mathsf{C}_i$" by $\zeta_{\mathsf{E}}(\mathsf{x}_i,\mathsf{y}_j)$, and
  - replace all atomic formulas $\mathsf{x} = \mathsf{y}$ and $\mathsf{E}(\mathsf{x},\mathsf{y})$ by $\zeta_{=}(\mathsf{x},\mathsf{y})$ and $\zeta_{\mathsf{E}}(\mathsf{x},\mathsf{y})$, respectively.

We make clear that we assume that the collection $\mathbf{c}$ in the definitions in Subsection 6.1 is replaced by $(\mathsf{v}_1^1,\ldots,\mathsf{v}_k^1,\mathsf{v}_1^2,\ldots,\mathsf{v}_k^2)$.

Intuitively, in the sentence χ_k, "$\exists \mathsf{v}_1^1,\ldots,\mathsf{v}_k^1,\mathsf{v}_1^2,\ldots,\mathsf{v}_k^2$" asks for the existence of sets S_1 (corresponding to the interpretations of $\mathsf{v}_1^1,\ldots,\mathsf{v}_k^1$) and S_2 (corresponding to the interpretations of $\mathsf{v}_1^2,\ldots,\mathsf{v}_k^2$) each of size k, and a bijection $\sigma : S_1 \to S_2$, given by the ordering of the variables (i.e., mapping the interpretation of v_i^1 to the interpretation of v_i^2 for each $i \in [k]$). Also, for each $i \in [2]$, we demand S_i to be a subset of V_i ("$\bigwedge_{i\in[k],j\in[2]} \mathsf{v}_i^j \in V_j$"). Then, to express the identification of each $v \in S_1$ to $\sigma(v)$ and the satisfaction of φ from the graph $G_1 \otimes_k G_2$, we define the formula $\varphi^{\star}$ with $\mathsf{v}_1^1,\ldots,\mathsf{v}_k^1,\mathsf{v}_1^2,\ldots,\mathsf{v}_k^2$ as free variables. For this formula, we use the idea from Subsection 6.1 for the definition of the apex-projection of a formula, in order to deal with S_1 and S_2 separately and ask a modified version of φ in the resulting graph. First feature of $\varphi^{\star}$ is to consider the interpretations of v_i^1 and v_i^2 as the same vertex. To incorporate this, we "re-define" equality as $\zeta_{=}(\mathsf{x},\mathsf{y})$. The respective configuration has to be done to the adjacency predicate E and all atomic formulas $(\mathsf{s})\mathsf{dp}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_t,\mathsf{t}_t)$ in φ. For this reason, we define $\zeta_{\mathsf{E}}(\mathsf{x},\mathsf{y})$ in a way that, for example, v_i^1 is adjacent to y adjacent if and only if v_i^2 is adjacent to y. Then, each atomic formula $(\mathsf{s})\mathsf{dp}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_t,\mathsf{t}_t)$ is "splitted" to the part that concerns $\mathsf{v}_1^1,\ldots,\mathsf{v}_k^1,\mathsf{v}_1^2,\ldots,\mathsf{v}_k^2$ and the rest of the graph, and "guessing" where the supposed paths should enter or exit the sets S_1 and S_2. This idea is the same as the one in Subsection 6.1 and for this reason we use the formula $\zeta_{(\mathsf{s})\mathsf{dp}}(\mathsf{s}_1,\mathsf{t}_1,\ldots,\mathsf{s}_t,\mathsf{t}_t)$. However, here we *do not* need to remove the edges between the apex-tuple and the rest of the graph, so we may just ask for adjacencies between the "apex set" and the rest of the graph. This is why we can write $\zeta_{\mathsf{E}}(\mathsf{x}_i,\mathsf{y}_j)$ instead of "$\mathsf{y}_j \in \mathsf{C}_i$" and avoid using the toolbox of "apex-projection" and "backwards-translation". Also, to be consistent to the identification given by σ, we have to replace all atomic formulas $\mathsf{x} = \mathsf{y}$ and $\mathsf{E}(\mathsf{x},\mathsf{y})$ by $\zeta_{=}(\mathsf{x},\mathsf{y})$ and $\zeta_{\mathsf{E}}(\mathsf{x},\mathsf{y})$, respectively. Observe that if $\varphi \in \mathsf{FOL{+}DP}$ (resp. $\varphi \in \mathsf{FOL{+}DP}$), then $\chi_k \in \mathsf{FOL{+}DP}$ (resp. $\chi_k \in \mathsf{FOL{+}DP}$)

According to the above, if $\varphi \in \mathsf{FOL{+}DP}$ (resp. $\varphi \in \mathsf{FOL{+}DP}$), then φ-Amalgamation is FPT on graphs of bounded Hadwiger number (resp. bounded genus). In [58], de Oliveira Oliveira considered, given a pattern graph H on k vertices, the alternative amalgamation operation $G_1 \otimes_H G_2$ where the subgraphs of G_1 and G_2 induced by S_1 and S_2 are also asked to be isomorphic to H. Clearly, this operation can also be treated by above machinery by introducing the isomorphism of $G_1[S_1]$ and $G_2[S_2]$ in the formula χ_k. For this operation, de Oliveira Oliveira [58, Theorem 4.3] proves that there is an algorithm that, given a sentence $\varphi \in \mathsf{CMSOL}$, three (connected) graphs G_1, G_2, H of treewidth at most t and maximum degree at most Δ, reports whether $G_1 \otimes_H G_2 \models \varphi$ in time $\mathcal{O}_{|\varphi|,t,\Delta}(n^{\mathcal{O}(t)})$, where $n = |G_1| + |G_2|$. This result is incomparable with our results.

Note that if the models of φ have bounded Hadwiger number, then we define problems of Category A that are in FPT in general. As an example of such a problem we mention Planar Amalgamation.

A.5 Actions and replacements

In [84] a general local graph modification framework was defined where we consider a set of ways, called *actions*, that locally replace small size subgraph patterns of a graph. The problem treated in [84] is whether such a replacement (or a sequence of such replacements) may modify the graph so to satisfy some graph property. In [84] this property was planarity (and some modifications of it). Here we will consider a way more general setting.

Replacement actions. We start with some necessary definitions. We use the notation $\mathbf{inj}([k], G)$ for all the injections of $[k]$ to the set of vertices of G. A *k-numbered-graph* is any graph H where $V(H) = [k]$, i.e., the vertices of H are the numbers $\{1, \ldots, k\}$. We denote the set of all k-numbered graphs by $\mathcal{H}_k$ and we set $\mathcal{H} = \bigcup_{k\in\mathbb{N}} \mathcal{H}_k$. A *replacement action* is any function $\mathcal{L} : \mathcal{H} \to \mathcal{H}$, where for every $H \in \mathcal{H}$, $|\mathcal{L}(H)| = |H|$, i.e., graphs in $\mathcal{H}$ are mapped to same-size graphs.

Let G be a graph and let $\mu \in \mathbf{inj}([k], G)$. We set $\mu^{-1}(G) = ([k], \{\mu^{-1}(e) \mid e \in E(G[\mu([k])])\})$, i.e., we see $\mu^{-1}(G)$ is the graph in $\mathcal{H}_k$ that is isomorphic, via μ, to the subgraph of G where μ applies.

Let G be a graph and let G' be an other graph where $V(G') \subseteq V(G)$. We denote $G \sqcup G' = (G \setminus \binom{V(G')}{2}) \cup G'$, i.e., $G \sqcup G'$ occurs if we remove from G the edges between vertices in G' and then add all edges of G'. Given a graph G, a $\mu \in \mathbf{inj}([k], V(G))$, and a $H \in \mathcal{H}_k$, we define $\mu(H) = \{\mu([k]), \{\mu(e) \mid e \in E(H)\}\}$. Given a replacement action $\mathcal{L} : \mathcal{H} \to \mathcal{H}$, we set $\mathcal{L}_\mu(G) = G \sqcup \mu(\mathcal{L}(\mu^{-1}(G)))$, in other words, we consider the part of G that is delimited by μ and then we replace this part by its image via $\mathcal{L}$.

An action $\mathcal{L}$ may be seen as prescribed way to locally change a graph. It might be the complementation of the edges of a subgraph G' of G, their removal, or the addition of a clique on the vertices of G', or the removal of a matching from G'.

We now have all ingredients we need for defining a general local replacement problem. Let $\mathcal{L}$ be an action and let $\varphi \in \mathsf{FOL{+}DP}$ ($\varphi \in \mathsf{FOL{+}SDP}$). We define the following problem.

$\mathcal{L}$-φ-Replacement
Input: a graph G and a $k \in \mathbb{N}$.
Question: Is there a $\mu \in \mathbf{inj}([k], V(G))$ such that $\mathcal{L}_\mu(G) \models \varphi$?

Expressing $\mathcal{L}$-φ-Replacement. We claim that, for every action $\mathcal{L}$ and every sentence $\varphi \in \mathsf{FOL{+}DP}$ ($\varphi \in \mathsf{FOL{+}SDP}$), there is a sentence $\xi_k \in \mathsf{FOL{+}DP}$ ($\xi_k \in \mathsf{FOL{+}SDP}$) such that (G, k) is a yes-instance of $\mathcal{L}$-φ-Replacement if and only if $G \models \xi_k$ To see this, consider the sentence

$$\xi_k = \exists \mathsf{v}_1, \ldots, \mathsf{v}_k\ \hat{\varphi}(\mathsf{v}_1, \ldots, \mathsf{v}_k),$$

where $\hat{\varphi}$ is the formula obtained from $\varphi^\star$ of Subsection A.4, after replacing each atomic formula $\zeta_=(\mathsf{x}, \mathsf{y})$ by $\mathsf{x} = \mathsf{y}$, each atomic formula $\zeta_\mathsf{E}(\mathsf{x}, \mathsf{y})$ by the formula

$$\xi_\mathsf{E}(\mathsf{x}, \mathsf{y}) = \Bigg(\Big(\mathsf{E}(\mathsf{x}, \mathsf{y}) \wedge \bigwedge_{(i,j)\in\binom{[k]}{2}} (\mathsf{x} \neq \mathsf{v}_i \vee \mathsf{y} \neq \mathsf{v}_j)\Big) \vee \bigwedge_{H\in\mathcal{H}_k} \Big(\bigwedge_{(i,j)\in E(H)} (\mathsf{x} = \mathsf{v}_i \wedge \mathsf{y} = \mathsf{v}_j) \implies \bigwedge_{(i,j)\in E(\mathcal{L}(H))} (\mathsf{x} = \mathsf{v}_i \wedge \mathsf{y} = \mathsf{v}_j)\Big)\Bigg).$$

Intuitively, the sentence ξ_k asks the existence of an $\mu \in \mathbf{inj}([k], V(G))$ ("$\exists \mathsf{v}_1, \ldots, \mathsf{v}_k$") such that the modified graph $\mathcal{L}_\mu(G)$ satisfies φ. We use $v_1, \ldots, v_k$ to denote the interpretation of $\mathsf{v}_1, \ldots, \mathsf{v}_k$. To express the replacement action, we define $\hat{\varphi}$ to be the formula obtained from φ after "forgetting" all edges between $v_1, \ldots, v_k$ ("$\bigwedge_{(i,j)\in\binom{[k]}{2}}(\mathsf{x} \neq \mathsf{v}_i \vee \mathsf{y} \neq \mathsf{v}_j)$") and after "adding" all edges given by the replacement action $\mathcal{L}$, i.e., for each graph $H \in \mathcal{H}_k$, if $H = \mu^{-1}(G)$, then "add" the edges of $\mu(\mathcal{L}(\mu^{-1}(G)))$ in G. This is expressed using the formula "$\bigwedge_{H\in\mathcal{H}_k}\Big(\bigwedge_{(i,j)\in E(H)}(\mathsf{x} = \mathsf{v}_i \wedge \mathsf{y} = \mathsf{v}_j) \implies \bigwedge_{(i,j)\in E(\mathcal{L}(H))}(\mathsf{x} = \mathsf{v}_i \wedge \mathsf{y} = \mathsf{v}_j)\Big)$". Additionally, in order to deal with disjoint paths that pass through $v_1, \ldots, v_k$, we use the trick in Subsection A.4 to replace the atomic formulas $(\mathsf{s})\mathsf{dp}(\mathsf{s}_1, \mathsf{t}_1, \ldots, \mathsf{s}_t, \mathsf{t}_t)$. For this reason, we define the formula $\hat{\varphi}$ as a modified version of $\varphi^\star$ from Subsection A.4 by only replacing each atomic formula $\zeta_=(\mathsf{x}, \mathsf{y})$ by $\mathsf{x} = \mathsf{y}$ and each atomic formula $\zeta_\mathsf{E}(\mathsf{x}, \mathsf{y})$ (even the ones that appear in the "translation" of the atomic formulas $(\mathsf{s})\mathsf{dp}$) by $\xi_\mathsf{E}(\mathsf{x}, \mathsf{y})$. Observe that if $\varphi \in$ FOL+DP (resp. $\varphi \in$ FOL+SDP), then $\xi_k \in$ FOL+DP (resp. $\xi_k \in$ FOL+SDP).

According to the above, if $\varphi \in$ FOL+DP (resp. $\varphi \in$ FOL+SDP), then $\mathcal{L}$-φ-Replacement is FPT on graphs of bounded Hadwiger number (resp. bounded genus). Again in case where the models of φ have bounded Hadwiger number then we obtain problems of Category A that belong FPT in general and this already includes the results of [84] where the target property was planarity. In fact using the above setting we may extend the definition of replacement actions so to permit the substitution of subgraphs of the input graph with new graphs of different (but still bounded) sizes. This would make it possible to define more flexible types of modifications such as edge contractions or Δ-Y transformations.

A.6 Elimination distance problems

Given a graph G, we use $\mathsf{cc}(G)$ to denote the connected components of G. We say that a graph class is *non-trivial* if it contains at least one non-empty graph and does not contain all graphs. Given a graph class $\mathcal{G}$, we define the *connected closure* of $\mathcal{G}$ as $\mathcal{C}(\mathcal{G}) = \{G \mid \forall C \in \mathsf{cc}(G), C \in \mathcal{G}\}$. Also, we use $\mathcal{A}(\mathcal{G})$ to denote the set $\{G \mid \exists v \in V(G) G \setminus v \in \mathcal{G}\}$.

Let $\mathcal{G}$ be a non-trivial graph class. We say that a graph G has *elimination distance at most k* to $\mathcal{G}$ if

$$G \in \overbrace{\mathcal{C}(\mathcal{A}(\cdots\mathcal{C}(\mathcal{A}}^{k \text{ times}}(\mathcal{C}(\mathcal{G}))))).$$

The elimination distance from a graph class was defined by Bulian and Dawar in [45] as an alternative graph modification measure (see [4–7,122,128,129,166] for algorithmic results concerning elimination distance). Given a $\varphi \in$ FOL+DP (resp. $\varphi \in$ FOL+SDP) we define the following problem:

φ-Elimination distance *Input*: a graph G and a $k \in \mathbb{N}$. *Question*: is the elimination distance of G from $\mathrm{Mod}(\varphi)$ at most k?

Bulian and Dawar in [46] considered the above problem for the case where φ expresses the minor-exclusion of some finite set of graphs and they proved that this problem is (constructively) FPT. In [85] considered φ-Elimination distance when $\varphi \in$ FOL and proved that for particular instantiations of φ the problem, parameterized by k, is W[2]-hard. According to the recent meta-algorithmic results in [188], when $\varphi \in$FOL+conn, φ-Elimination distance is FPT for graphs of bounded Hajós number (see [201]). According to our results if $\varphi \in$ FOL+SDP (resp. $\varphi \in$ FOL+DP), then φ-Elimination distance is FPT on graphs of bounded Hadwiger number (resp. bounded genus).

The problem φ-Block Elimination Distance. Another parameter similar to elimination distance is the *block elimination distance*, introduced in [64], that is obtained if we replace the connected closure operator $\mathcal{C}$ in the above definition by the operator $\mathcal{B}$, defined as

$$\mathcal{B}(\mathcal{G}) = \{G \mid \forall B \in \mathsf{bc}(G),\ B \in \mathcal{B}\},$$

where $\mathsf{bc}(G)$ is the set of all blocks of G. Similar to φ-Elimination distance, we can define the problem φ-Block Elimination Distance. According to our results, if $\varphi \in \mathsf{FOL}{+}\mathsf{DP}$ (resp. $\varphi \in \mathsf{FOL}{+}\mathsf{SDP}$), then φ-Block Elimination Distance is FPT on graphs of bounded Hadwiger number (resp. bounded genus). We remark, that when $\varphi \in$FOL+conn, φ-Elimination distance is FPT for graphs of bounded Hajós number because of the results in [188].

A.7 Reconfiguration problems

Intuitively, reconfiguration problems ask, given two feasible solutions S and T of a problem, whether there is a step-by-step transformation between S and T where all intermediate sets are also feasible solutions.

Reconfiguration sequences. Let φ be a sentence that is satisfied in structures of the form (G, S), i.e., structures of the colored-graph vocabulary $\{\mathsf{E}, \mathsf{S}\}$. A sequence $S_1, \ldots, S_\ell, \ell \in \mathbb{N}$ of subsets of $V(G)$ is called a *φ-reconfiguration sequence* if

- for every $i \in [\ell]$, $(G, S_i) \models \varphi$ and
- for every $i \in [\ell]$, there is a $v \in S_i$ and a $u \in V(G) \setminus S_i$ such that $S_{i+1} = (S_i \setminus \{v\}) \cup \{u\}$.

We call ℓ the *lenght* of the φ-reconfiguration sequence.

Given a $\varphi \in \mathsf{FOL}{+}\mathsf{DP}$ (resp. $\varphi \in \mathsf{FOL}{+}\mathsf{SDP}$), whose models are of the form (G, S), we define the following problem:

φ-Reconfiguration
Input: a graph G, two sets $S, T \subseteq V(G)$, and an $\ell \in \mathbb{N}$.
Question: is there a φ-reconfiguration sequence $S, \ldots, T$ of length at most ℓ?

Reconfiguration problems have received a lot of attention in the literature [37, 43, 104, 117, 119, 124, 125, 167, 168, 173, 173, 175, 176, 185, 202]). A vibrant branch of research on reconfiguration problems deals with other reconfiguration models (apart from removing/adding vertices) like token sliding, (perfect) matching flipping, spanning tree flipping [11, 12, 16, 24, 25, 39–42, 59, 74, 210]. Also, the tractability of reconfiguration problems has been studied under different structural parameterizations of the input graph [15, 19, 20, 172, 209].

Known AMTs for reconfiguration problems. Also, there are some konwn algorithmic metatheorems for reconfiguration problems. In fact, [174] proved that for every $\varphi \in \mathsf{MSO}_2$, the problem φ-Reconfiguration is FPT parameterized by $\mathsf{tw} + \ell + |\varphi|$, where tw is the treewidth of the input graph. Also, their framework can be used to derive an FPT algorithm for the parameterization of the problem φ-Reconfiguration for $\varphi \in \mathsf{MSO}_1$ by $\ell + \mathsf{cw} + |\varphi|$, where cw is the cliquewidth of the input graph. For formulas $\varphi \in \mathsf{MSOL}$, in [98], they considered two parameterizations of φ-Reconfiguration: the first is by the neighborhood diversity of the input graph and the second is, when restricted to feasible solutions of size k, by the treewidth of the input graph and k. For these two parameterized (meta)problems, they give an FPT algorithm.

For $\varphi \in \mathsf{FOL}$, the variant of φ-Reconfiguration where the size of all sets in the reconfiguration sequence is k, is FPT parameterized by $\ell + k + |\varphi|$ on nowhere dense classes [167]. This result also holds for any reconfiguration model (for example, token sliding) that can be expressed by a formula in FOL.

Expressing φ-Reconfiguration. Let $\varphi \in \mathsf{FOL+DP}$ (resp. $\varphi \in \mathsf{FOL+SDP}$). We claim that there is a sentence $\tilde{\varphi}_\ell \in \mathsf{FOL+DP}$ ($\tilde{\varphi}_\ell \in \mathsf{FOL+SDP}$) such that (G, S, T, ℓ) is a yes-instance of φ-Reconfiguration if and only if $(G, S, T) \models \tilde{\varphi}_\ell$. To see this, first, we consider the formula $\psi_{\mathsf{S}}^{(0)}(\mathsf{z}) = (\mathsf{z} \in \mathsf{S})$ and, for every $i \in [\ell - 1]$, we consider the formula

$$\psi_{\mathsf{S}}^{(i)}(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_i, \mathsf{y}_i, \mathsf{z}) = \Big(\psi_{\mathsf{S}}^{(i-1)}(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_{i-1}, \mathsf{y}_{i-1}, \mathsf{z}) \wedge \mathsf{z} \neq \mathsf{x}_i\Big) \vee \mathsf{z} = \mathsf{y}_i.$$

Intuitively, for every $i \in \mathbb{N}$, given a graph G, a set $S \subseteq V(G)$ that interprets S, and some vertices $x_1, y_1, \ldots, x_i, y_i, z \in V(G)$ that interpret the variables $\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_i, \mathsf{y}_i, \mathsf{z}$, $\psi_{\mathsf{S}}^{(i)}(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_i, \mathsf{y}_i, \mathsf{z})$ expresses that $z \in S_i$, where $S_0 = S$ and for every $j \in [i]$, $S_j = (S_{j-1} \setminus \{x_j\}) \cup \{y_j\}$.

Also, for every $i \in \mathbb{N}$, we define $\varphi_{\mathsf{S}}^{(i)}(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_i, \mathsf{y}_i)$ to be the formula obtained from φ after replacing each atomic term $\mathsf{x} \in \mathsf{S}$ with the formula $\psi_{\mathsf{S}}^{(i)}(\mathsf{x}_1, \mathsf{y}_1, \ldots, \mathsf{x}_i, \mathsf{y}_i, \mathsf{x})$. Intuitively, the formula $\varphi_{\mathsf{S}}^{(i)}$ demands that the variables of φ are picked inside the set S_i, instead of the set S, where S_i is defined as above. We define $\tilde{\varphi}_\ell$ as follows.

$$\tilde{\varphi}_\ell = \exists \mathsf{v}_1, \mathsf{u}_1, \ldots, \mathsf{v}_\ell, \mathsf{u}_\ell \bigwedge_{i \in [\ell]} \Big(\psi_{\mathsf{S}}^{(i-1)}(\mathsf{v}_1, \mathsf{u}_1, \ldots, \mathsf{v}_{i-1}, \mathsf{u}_{i-1}, \mathsf{v}_i) \wedge \neg\psi_{\mathsf{S}}^{(i-1)}(\mathsf{v}_1, \mathsf{u}_1, \ldots, \mathsf{v}_{i-1}, \mathsf{u}_{i-1}, \mathsf{u}_i)\Big) \wedge \varphi_{\mathsf{S}}^{(i)}(\mathsf{v}_1, \mathsf{u}_1, \ldots, \mathsf{v}_i, \mathsf{u}_i).$$

Observe that if $\varphi \in \mathsf{FOL+DP}$ (resp. $\varphi \in \mathsf{FOL+SDP}$), then $\tilde{\varphi}_\ell \in \mathsf{FOL+DP}$ (resp. $\tilde{\varphi}_\ell \in \mathsf{FOL+SDP}$). According to our results, if $\varphi \in \mathsf{FOL+DP}$ (resp. $\varphi \in \mathsf{FOL+SDP}$), then φ-Reconfiguration is FPT, parameterized by ℓ and $|\varphi|$, on graphs of bounded Hadwiger number (resp. bounded genus).

A.8 Planarizer game

We would like to finish this section with a new problem that we consider worth mentioning here. We call it Planarizer Game and it is played by two players, the *blocker* and the *planarizer*. The two players play in rounds. In each round each player places tokens on the vertices of the graph and the blocker plays first. None of the players can move his/her token on a vertex that is already occupied by a token. The planarizer wins if the removal from G of the vertices that are occupied by his/her tokens yields a planar graph. The Planarizer Game problem asks, given a graph G and a non-negative integer k, whether the planarizer has a victory strategy against the blocker that uses at most k rounds. To prove that Planarizer Game is NP-hard we conside the Edge Planarizer problem asking, give a graph G and a non-negative integer k, whether there is set of at most k edges of G whose removal produces a planar graph. In [75], Fariaa, Herrera de Figueiredo, Mendonça, proved that this problem is NP-hard for graphs of maximum degree three. We reduce this restricted version of Edge Planarizer to Planarizer Game as follows.

Let (G, k) be an input of Edge Planarizer. We transform (G, k) to an input (G', k) of Planarizer Game by replacing each edge by a path of length k and by removing every vertex v of degree three and making $N_G(v)$ a clique (i.e., a triangle). Then (G, k) is a yes instance of Edge Planarizer iff (G', k) is a yes instance of Planarizer Game because each edge-choice of Edge Planarizer corresponds to a choice of a vertex of a joining path and the length of joining paths is big enough to "neutralize" the potential of the blocker during the game.

According to Theorem 1 and given that yes-instances of Planarizer Game have bounded Hadwiger number, the problem belongs in Category A and, when parameterized by the number of rounds, is FPT. Of course the same result can be generalized if, instead of planarity, the planarizer pursues some other graph property whose graphs have bounded Hadwiger number (resp. genus) and is expressible in FOL+DP (resp. FOL+SDP).

B Flat walls and flat annuli framework

Here we present the framework on flat walls that was introduced in [197]. In Subsection B.1 we define walls, subwalls, and other notions related to walls. Next, in Subsection B.2, we give the definitions of renditions and paintings, that are used in Subsection B.3 to define flatness pairs. In Subsection B.3, apart from the definition of flatness pairs, we present notions like *influence*, *regularity*, and *tilts*.

B.1 Walls and subwalls

Dissolutions and subdivisions. Given a vertex $v \in V(G)$ of degree two with neighbors u and w, we define the *dissolution* of v to be the operation of deleting v and, if u and w are not adjacent, adding the edge $\{u, w\}$. Given an edge $e = \{u, v\} \in E(G)$, we define the *subdivision* of e to be the operation of deleting e, adding a new vertex w and making it adjacent to u and v. Given two graphs H, G, we say that H is a *subdivision* of G if H can be obtained from G after subdividing edges of G.

Walls. Let $k, r \in \mathbb{N}$. The $(k \times r)$-*grid* is the graph whose vertex set is $[k] \times [r]$ and two vertices (i, j) and (i', j') are adjacent if and only if $|i - i'| + |j - j'| = 1$. An *elementary r-wall*, for some odd integer $r \geq 3$, is the graph obtained from a $(2r \times r)$-grid with vertices $(x, y) \in [2r] \times [r]$, after the removal of the "vertical" edges $\{(x, y), (x, y + 1)\}$ for odd $x + y$, and then the removal of all vertices of degree one. Notice that, as $r \geq 3$, an elementary r-wall is a planar graph that has a unique (up to topological isomorphism) embedding in the plane such that all its finite faces are incident to exactly six edges. The *perimeter* of an elementary r-wall is the cycle bounding its infinite face, while the cycles bounding its finite faces are called *bricks*. Also, the vertices in the perimeter of an elementary r-wall that have degree two are called *pegs*, while the vertices $(1, 1), (2, r), (2r-1, 1), (2r, r)$ are called *corners* (notice that the corners are also pegs).

An *r-wall* is any graph W obtained from an elementary r-wall $\bar{W}$ after subdividing edges. A graph W is a *wall* if it is an r-wall for some odd $r \geq 3$ and we refer to r as the *height* of W. Given a graph G, a *wall of G* is a subgraph of G that is a wall. We insist that, for every r-wall, the number r is always odd.

We call the vertices of degree three of a wall W *3-branch vertices*. A cycle of W is a *brick* (resp. the *perimeter*) of W if its 3-branch vertices are the vertices of a brick (resp. the perimeter) of $\bar{W}$. We denote by $\mathcal{C}(W)$ the set of all cycles of W. We use $D(W)$ in order to denote the perimeter of the wall W. A brick of W is *internal* if it is disjoint from $D(W)$.

Subwalls. Given an elementary r-wall $\bar{W}$, some odd $i \in \{1, 3, \ldots, 2r - 1\}$, and $i' = (i + 1)/2$, the *i'-th vertical path* of $\bar{W}$ is the one whose vertices, in order of appearance, are $(i, 1), (i, 2), (i + 1, 2), (i + 1, 3), (i, 3), (i, 4), (i + 1, 4), (i + 1, 5), (i, 5), \ldots, (i, r - 2), (i, r - 1), (i + 1, r - 1), (i + 1, r)$. Also, given some $j \in [2, r - 1]$ the *j-th horizontal path* of $\bar{W}$ is the one whose vertices, in order of appearance, are $(1, j), (2, j), \ldots, (2r, j)$.

A *vertical* (resp. *horizontal*) path of W is one that is a subdivision of a vertical (resp. horizontal) path of $\bar{W}$. Notice that the perimeter of an r-wall W is uniquely defined regardless of the choice of the elementary r-wall $\bar{W}$. An *r'-subwall* (or simply *subwall*) of W is any subgraph W' of W that is an r'-wall, with $r' \leq r$, and such the vertical (resp. horizontal) paths of W' are subpaths of the vertical (resp. horizontal) paths of W.

Layers. The *layers* of an r-wall W are recursively defined as follows. The first layer of W is its perimeter. For $i = 2, \dots, (r-1)/2$, the i-th layer of W is the $(i-1)$-th layer of the subwall W' obtained from W after removing from W its perimeter and removing recursively all occurring vertices of degree one. We refer to the $(r-1)/2$-th layer as the *inner layer* of W. The *central vertices* of an r-wall W are its two branch vertices that do not belong to any of its layers and are connected by a path of W that does not intersect any layers of W.

Central walls. Given an r-wall W and an odd $q \in \mathbb{N}_{\geq 3}$ where $q \leq r$, we define the *central q-subwall* of W, denoted by $W^{(q)}$, to be the q-wall obtained from W after removing its first $(r-q)/2$ layers and all occurring vertices of degree one. Given an $h \in [(r-1)/2]$, a subwall W' of W is called *h-internal* if it is a subwall of $W^{(r-2h)}$.

Tilts. The *interior* of a wall W is the graph obtained from W if we remove from it all edges of $D(W)$ and all vertices of $D(W)$ that have degree two in W. Given two walls W and $\tilde{W}$ of a graph G, we say that $\tilde{W}$ is a *tilt* of W if $\tilde{W}$ and W have identical interiors.

B.2 Paintings and renditions

In this subsection we present the notions of renditions and paintings, originating in the work of Robertson and Seymour [193]. The definitions presented here were introduced by Kawarabayashi, Thomas, and Wollan [146] (see also [197]).

Paintings. Let Δ be a closed annulus or a closed disk. A *Δ-painting* is a pair $\Gamma = (U, N)$ where

- N is a finite set of points of Δ,
- $N \subseteq U \subseteq \Delta$, and
- $U \setminus N$ has finitely many arcwise-connected components, called *cells*, where, for every cell c,
  - the closure $\bar{c}$ of c is a closed disk and
  - $|\tilde{c}| \leq 3$, where $\tilde{c} := \mathsf{bd}(c) \cap N$.

We use the notation $U(\Gamma) := U$, $N(\Gamma) := N$ and denote the set of cells of Γ by $C(\Gamma)$. For convenience, we may assume that each cell of Γ is an open disk of Δ. Notice that, given a Δ-painting Γ, the pair $(N(\Gamma), \{\tilde{c} \mid c \in C(\Gamma)\})$ is a hypergraph whose hyperedges have cardinality at most three and Γ can be seen as a plane embedding of this hypergraph in Δ.

Disk and annulus renditions. Let G be a graph and let Ω be a cyclic permutation of a subset of $V(G)$ that we denote by $V(\Omega)$. By a *disk Ω-rendition* of G we mean a triple (Γ, σ, π), where

(a) Γ is a Δ-painting for some closed disk Δ,

(b) $\pi : N(\Gamma) \to V(G)$ is an injection, and

(c) σ assigns to each cell $c \in C(\Gamma)$ a subgraph $\sigma(c)$ of G, such that

  (1) $G = \bigcup_{c \in C(\Gamma)} \sigma(c)$,
  (2) for distinct $c, c' \in C(\Gamma)$, $\sigma(c)$ and $\sigma(c')$ are edge-disjoint,
  (3) for every cell $c \in C(\Gamma)$, $\pi(\tilde{c}) \subseteq V(\sigma(c))$,

(4) for every cell $c \in C(\Gamma)$, $V(\sigma(c)) \cap \bigcup_{c' \in C(\Gamma) \setminus \{c\}} V(\sigma(c')) \subseteq \pi(\tilde{c})$, and
(5) $\pi(N(\Gamma) \cap \mathsf{bd}(\Delta)) = V(\Omega)$, such that the points in $N(\Gamma) \cap \mathsf{bd}(\Delta)$ appear in $\mathsf{bd}(\Delta)$ in the same ordering as their images, via π, in Ω.

Similarly, we define *annulus* (Ω_1, Ω_2)-*renditions* as follows. Let G be a graph and let X_1, X_2 be two subsets of $V(G)$, and let Ω_1 (resp. Ω_2) be a cyclic permutation of X_1 (resp. X_2). By an (Ω_1, Ω_2)-*rendition* of G we mean a triple (Γ, σ, π), where (Γ, σ, π), is defined as for disk Ω-renditions but Γ is a Δ-painting for some closed annulus Δ (instead of a closed disk) and as for item (5), $\pi(N(\Gamma) \cap \mathsf{bd}(\Delta)) = X_1 \cup X_2$, such that, if $\mathsf{bd}(\Delta) = B_1 \cup B_2$, then the points in $N(\Gamma) \cap B_i$ appear in B_i in the same ordering as their images, via π, in Ω_i, for $i \in \{1, 2\}$.

B.3 Flatness pairs

In this subsection we define the notion of a flat wall, originating in the work of Robertson and Seymour [193] and later used in [146]. Here, we define flat walls as in [197].

Flat walls. Let G be a graph and let W be an r-wall of G, for some odd integer $r \geq 3$. We say that a pair $(P, C) \subseteq D(W) \times D(W)$ is a *choice of pegs and corners for* W if W is the subdivision of an elementary r-wall $\bar{W}$ where P and C are the pegs and the corners of $\bar{W}$, respectively (clearly, $C \subseteq P$). To get more intuition, notice that a wall W can occur in several ways from the elementary wall $\bar{W}$, depending on the way the vertices in the perimeter of $\bar{W}$ are subdivided. Each of them gives a different selection (P, C) of pegs and corners of W.

We say that W is a *flat r-wall* of G if there is a separation (X, Y) of G and a choice (P, C) of pegs and corners for W such that:

- $V(W) \subseteq Y$,
- $P \subseteq X \cap Y \subseteq V(D(W))$, and
- if Ω is the cyclic ordering of the vertices $X \cap Y$ as they appear in $D(W)$, then there exists an Ω-rendition (Γ, σ, π) of $G[Y]$.

We say that W is a *flat wall* of G if it is a flat r-wall for some odd integer $r \geq 3$.

Flatness pairs. Given the above, we say that the choice of the 7-tuple $\mathfrak{R} = (X, Y, P, C, \Gamma, \sigma, \pi)$ *certifies that W is a flat wall of G*. We call the pair $(W, \mathfrak{R})$ a *flatness pair* of G and define the *height* of the pair $(W, \mathfrak{R})$ to be the height of W. We use the term *cell of* $\mathfrak{R}$ in order to refer to the cells of Γ.

We call the graph $G[Y]$ the $\mathfrak{R}$-*compass* of W in G, denoted by $\mathsf{Compass}_{\mathfrak{R}}(W)$. It is easy to see that there is a connected component of $\mathsf{Compass}_{\mathfrak{R}}(W)$ that contains the wall W as a subgraph. We can assume that $\mathsf{Compass}_{\mathfrak{R}}(W)$ is connected, updating $\mathfrak{R}$ by removing from Y the vertices of all the connected components of $\mathsf{Compass}_{\mathfrak{R}}(W)$ except of the one that contains W and including them in X (Γ, σ, π can also be easily modified according to the removal of the aforementioned vertices from Y). We define the *flaps* of the wall W in $\mathfrak{R}$ as $\mathsf{flaps}_{\mathfrak{R}}(W) := \{\sigma(c) \mid c \in C(\Gamma)\}$. Given a flap $F \in \mathsf{flaps}_{\mathfrak{R}}(W)$, we define its *base* as $\partial F := V(F) \cap \pi(N(\Gamma))$.

Flat railed annuli. Let G be a graph and let $\mathcal{A}$ be an (r, q)-railed annulus of G, for some odd integer $r \geq 3$ and $q \in \mathbb{N}_{\geq 3}$. We say that $\mathcal{A}$ is a *flat (r, q)-railed annulus* of G if there are two laminar separations $(X_1, Y_1), (X_2, Y_2)$ of G, a set Z_1 of degree-two vertices in $V(C_1)$, and a set Z_2 of degree-two vertices in $V(C_2)$ such that:

- $V(\mathcal{A}) \subseteq Y_1 \cap X_2$,
- $Z_1 \subseteq X_1 \cap Y_1 \subseteq V(C_1)$,
- $Z_2 \subseteq X_2 \cap Y_2 \subseteq V(C_r)$, and
- if Ω_1 (resp. Ω_2) is the cyclic ordering of the vertices $X_1 \cap Y_1$ (resp. $X_2 \cap Y_2$) as they appear in C_1 (resp. C_r), then there is an (Ω_1, Ω_2)-rendition of $G[X_1 \cap Y_2]$.

We say that $\mathcal{A}$ is a *flat railed annulus* of G if it is a flat (r, q)-railed annulus for some odd integer $r \geq 3$ and some $q \in \mathbb{N}_{\geq 3}$.

Railed annuli flatness pairs. Given the above, we say that the choice of the 9-tuple $\mathfrak{R} = (X_1, Y_1, X_2, Y_2, Z_1, Z_2, \Gamma, \sigma, \pi)$ *certifies that $\mathcal{A}$ is a flat railed annulus of* G. We call the pair $(\mathcal{A}, \mathfrak{R})$ an *railed annulus flatness pair* of G. We use the term *cell of* $\mathfrak{R}$ in order to refer to the cells of Γ.

We call the graph $G[X_1 \cap Y_2]$ the $\mathfrak{R}$-*compass* of $\mathcal{A}$ in G, denoted by $\mathsf{Compass}_{\mathfrak{R}}(\mathcal{A})$. It is easy to see that there is a connected component of $\mathsf{Compass}_{\mathfrak{R}}(\mathcal{A})$ that contains the cycles and the paths of $\mathcal{A}$ as subgraphs. We can assume that $\mathsf{Compass}_{\mathfrak{R}}(\mathcal{A})$ is connected, updating $\mathfrak{R}$ by removing from $X_1 \cap Y_2$ the vertices of all the connected components of $\mathsf{Compass}_{\mathfrak{R}}(\mathcal{A})$ except of the one that contains $\mathcal{A}$ and including them in X_2 (Γ, σ, π can also be easily modified according to the removal of the aforementioned vertices from $X_1 \cap Y_2$). We define the *flaps* of the railed annulus $\mathcal{A}$ in $\mathfrak{R}$ as $\mathsf{flaps}_{\mathfrak{R}}(\mathcal{A}) := \{\sigma(c) \mid c \in C(\Gamma)\}$. Given a flap $F \in \mathsf{flaps}_{\mathfrak{R}}(\mathcal{A})$, we define its *base* as $\partial F := V(F) \cap \pi(N(\Gamma))$.

B.4 Influence of cycles in flat walls and flat railed annuli

Let G be a graph and let $(W, \mathfrak{R})$ be either a flatness pair or a railed annulus flatness pair of G.

A cell c of $\mathfrak{R}$ is *untidy* if $\pi(\tilde{c})$ contains a vertex $x \in V(W)$ such that two of the edges in $E(W)$ that are incident to x are edges of $\sigma(c)$. Notice that if c is untidy then $|\tilde{c}| = 3$. A cell c of $\mathfrak{R}$ is *tidy* if it is not untidy. The notion of tidy/untidy cell as well as the notions that we present in the rest of this subsection have been introduced in [197].

Cell classification. Given a graph G and a set $X \subseteq V(G)$, we denote by $\partial_G(X)$ the set of vertices in X that are adjacent to vertices of $G \setminus X$.

Given a cycle C of $\mathsf{Compass}_{\mathfrak{R}}(W)$, we say that C is $\mathfrak{R}$-*normal* if it is *not* a subgraph of a flap $F \in \mathsf{flaps}_{\mathfrak{R}}(W)$. Given an $\mathfrak{R}$-normal cycle C of $\mathsf{Compass}_{\mathfrak{R}}(W)$, we call a cell c of $\mathfrak{R}$ *C-perimetric* if $\sigma(c)$ contains some edge of C. Since every C-perimetric cell c contains some edge of C and $|\partial\sigma(c)| \leq 3$, we observe the following.

Observation 13. *For every pair (C, C') of $\mathfrak{R}$-normal cycles of $\mathsf{Compass}_{\mathfrak{R}}(W)$ such that $V(C) \cap V(C') = \emptyset$, there is no cell of $\mathfrak{R}$ that is both C-perimetric and C'-perimetric.*

Notice that if c is C-perimetric, then $\pi(\tilde{c})$ contains two points $p, q \in N(\Gamma)$ such that $\pi(p)$ and $\pi(q)$ are vertices of C where one, say P_c^{in}, of the two $(\pi(p), \pi(q))$-subpaths of C is a subgraph of $\sigma(c)$ and the other, denoted by P_c^{out}, $(\pi(p), \pi(q))$-subpath contains at most one internal vertex of $\sigma(c)$, which should be the (unique) vertex z in $\partial\sigma(c) \setminus \{\pi(p), \pi(q)\}$. We pick a (p, q)-arc A_c in $\hat{c} := c \cup \tilde{c}$ such that $\pi^{-1}(z) \in A_c$ if and only if P_c^{in} contains the vertex z as an internal vertex.

We consider the circle $K_C = \bigcup\{A_c \mid c \text{ is a } C\text{-perimetric cell of } \mathfrak{R}\}$ and we denote by Δ_C the closed disk bounded by K_C that is contained in Δ (in the case of a railed annulus flatness pair, Δ_C the closed annulus bounded by K_C and B_1 that is contained in Δ). A cell c of $\mathfrak{R}$ is called *C-internal* if $c \subseteq \Delta_C$ and is called *C-external* if $\Delta_C \cap c = \emptyset$. Notice that the cells of $\mathfrak{R}$ are partitioned into C-internal, C-perimetric, and C-external cells.

Let c be a tidy C-perimetric cell of $\mathfrak{R}$ where $|\tilde{c}| = 3$. Notice that $c \setminus A_c$ has two arcwise-connected components and one of them is an open disk D_c that is a subset of Δ_C. If the closure $\overline{D}_c$ of D_c contains only two points of $\tilde{c}$ then we call the cell c *C-marginal*. We refer the reader to [197] for figures illustrating the above notions.

Influence. For every $\mathfrak{R}$-normal cycle C of $\mathsf{Compass}_{\mathfrak{R}}(W)$ we define the set

$$\mathsf{influence}_{\mathfrak{R}}(C) = \{\sigma(c) \mid c \text{ is a cell of } \mathfrak{R} \text{ that is not } C\text{-external}\}.$$

We conclude this subsection by presenting a corollary of [198, Lemma 12].

Proposition 6. *There exists a function $f_{17} : \mathbb{N}^3 \to \mathbb{N}$ such that if $z \in \mathbb{N}_{\geq 1}$, $x \in \mathbb{N}_{\geq 3}$ is an odd integer, G is a graph, and $(W, \mathfrak{R})$ is a flatness pair of G of height at least $f_{17}(z, x)$, then there is a collection $\mathcal{W} = \{W_1, \ldots, W_z\}$ of x-subwalls of W such that*

- *for every $i \in [z]$, $\bigcup \mathsf{influence}_{\mathfrak{R}}(D(W_i))$ is a subgraph of $\mathsf{Compass}_{\mathfrak{R}}(W)$ and*
- *for every $i, j \in [z]$, with $i \neq j$, $V(\bigcup \mathsf{influence}_{\mathfrak{R}}(D(W_i)))$ and $V(\bigcup \mathsf{influence}_{\mathfrak{R}}(D(W_j)))$ are pairwise disjoint.*

Moreover, $f_{17}(z, x, p) = \mathcal{O}(\sqrt{z} \cdot x)$ and $\mathcal{W}$ can be constructed in linear time.

B.5 Regular flatness pairs and tilts

Let $(W, \mathfrak{R})$ be a flatness pairs of a graph G. A wall W' of $\mathsf{Compass}_{\mathfrak{R}}(W)$ is *$\mathfrak{R}$-normal* if $D(W')$ is $\mathfrak{R}$-normal. Notice that every wall of W (and hence every subwall of W) is an $\mathfrak{R}$-normal wall of $\mathsf{Compass}_{\mathfrak{R}}(W)$. We denote by $\mathcal{S}_{\mathfrak{R}}(W)$ the set of all $\mathfrak{R}$-normal walls of $\mathsf{Compass}_{\mathfrak{R}}(W)$. Given a wall $W' \in \mathcal{S}_{\mathfrak{R}}(W)$ and a cell c of $\mathfrak{R}$, we say that c is *W'-perimetric/internal/external/marginal* if c is $D(W')$-perimetric/internal/external/marginal, respectively. We also use $K_{W'}$, $\Delta_{W'}$, $\mathsf{influence}_{\mathfrak{R}}(W')$ as shortcuts for $K_{D(W')}$, $\Delta_{D(W')}$, $\mathsf{influence}_{\mathfrak{R}}(D(W'))$, respectively.

Regular flatness pairs. We call a flatness pair $(W, \mathfrak{R})$ of a graph G *regular* if none of its cells is W-external, W-marginal, or untidy.

Tilts of flatness pairs. Let $(W, \mathfrak{R})$ and $(\tilde{W}', \tilde{\mathfrak{R}}')$ be two flatness pairs of a graph G and let $W' \in \mathcal{S}_{\mathfrak{R}}(W)$. We assume that $\mathfrak{R} = (X, Y, P, C, \Gamma, \sigma, \pi)$ and $\tilde{\mathfrak{R}}' = (X', Y', P', C', \Gamma', \sigma', \pi')$. We say that $(\tilde{W}', \tilde{\mathfrak{R}}')$ is a *W'-tilt* of $(W, \mathfrak{R})$ if

- $\tilde{\mathfrak{R}}'$ does not have $\tilde{W}'$-external cells,
- $\tilde{W}'$ is a tilt of W',
- the set of $\tilde{W}'$-internal cells of $\tilde{\mathfrak{R}}'$ is the same as the set of W'-internal cells of $\mathfrak{R}$ and their images via σ' and σ are also the same,
- $\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')$ is a subgraph of $\bigcup \mathsf{influence}_{\mathfrak{R}}(W')$, and
- if c is a cell in $C(\Gamma') \setminus C(\Gamma)$, then $|\tilde{c}| \leq 2$.

The next observation follows from the third item above and the fact that the cells corresponding to flaps containing a central vertex of W' are all internal (recall that the height of a wall is always at least three).

Observation 14. *Let* $(W, \mathfrak{R})$ *be a flatness pair of a graph* G *and* $W' \in \mathcal{S}_{\mathfrak{R}}(W)$. *For every* W'*-tilt* $(\tilde{W}', \tilde{\mathfrak{R}}')$ *of* $(W, \mathfrak{R})$, *the central vertices of* W' *belong to the vertex set of* $\mathsf{Compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')$.

Also, given a regular flatness pair $(W, \mathfrak{R})$ of a graph G and a $W' \in \mathcal{S}_{\mathfrak{R}}(W)$, for every W'-tilt $(\tilde{W}', \tilde{\mathfrak{R}}')$ of $(W, \mathfrak{R})$, by definition, none of its cells is $\tilde{W}'$-external, $\tilde{W}'$-marginal, or untidy – thus, $(\tilde{W}', \tilde{\mathfrak{R}}')$ is regular. Therefore, regularity of a flatness pair is a property that its tilts "inherit".

Observation 15. *If* $(W, \mathfrak{R})$ *is a regular flatness pair of a graph* G, *then for every* $W' \in \mathcal{S}_{\mathfrak{R}}(W)$, *every* W'*-tilt of* $(W, \mathfrak{R})$ *is also regular.*

We next present one of the two main results of [197] (see [197, Theorem 5]).

Proposition 7. *There exists an algorithm that given a graph* G, *a flatness pair* $(W, \mathfrak{R})$ *of* G, *and a wall* $W' \in \mathcal{S}_{\mathfrak{R}}(W)$, *outputs a* W'*-tilt of* $(W, \mathfrak{R})$ *in time* $\mathcal{O}(n + m)$.

We conclude this subsection with the Flat Wall theorem and, in particular, the version proved by Chuzhoy [51], restated in our framework (see [197, Proposition 7]).

Proposition 8. *There exist two functions* $f_{18} : \mathbb{N} \to \mathbb{N}$ *and* $f_{19} : \mathbb{N} \to \mathbb{N}$, *where the images of* f_{18} *are odd numbers, such that if* $r \in \mathbb{N}_{\geq 3}$ *is an odd integer,* $t \in \mathbb{N}_{\geq 1}$, G *is a graph that does not contain* K_t *as a minor, and* W *is an* $f_{18}(t) \cdot r$*-wall of* G, *then there is a set* $A \subseteq V(G)$ *with* $|A| \leq f_{19}(t)$ *and a flatness pair* $(\tilde{W}', \tilde{\mathfrak{R}}')$ *of* $G \setminus A$ *of height* r. *Moreover,* $f_{18}(t) = \mathcal{O}(t^2)$ *and* $f_{19}(t) = t - 5$.

B.6 Flat walls with compasses of bounded treewidth

The following result was proved in [197, Theorem 8]. It is a version of the Flat Wall theorem, originally proved in [193]. The proof in [197, Theorem 8] is strongly based on the proof of an improved version of the Flat Wall theorem given by of Kawarabayashi, Thomas, and Wollan [146] (see also [51, 97]).

Proposition 9. *There is a function* $f_{20} : \mathbb{N} \to \mathbb{N}$ *and an algorithm that receives as input a graph* G, *an odd integer* $r \geq 3$, *and a* $t \in \mathbb{N}_{\geq 1}$, *and outputs, in time* $2^{\mathcal{O}_t(r^2)} \cdot n$, *one of the following:*

- *a report that* K_t *is a minor of* G,
- *a tree decomposition of* G *of width at most* $f_{20}(t) \cdot r$, *or*
- *a set* $A \subseteq V(G)$, *where* $|A| \leq f_{19}(t)$, *a regular flatness pair* $(W, \mathfrak{R})$ *of* $G \setminus A$ *of height* r, *and a tree decomposition of the* $\mathfrak{R}$*-compass of* W *of width at most* $f_{20}(t) \cdot r$. *(Here* $f_{19}(t)$ *is the function of Proposition 8 and* $f_{20}(t) = 2^{\mathcal{O}(t^2 \log t)}$.*)*

Given graphs H and G, we say that a subgraph M of G is a *minor-model* of H in G if there is a partition of the vertex set of M to sets $V_1, \ldots, V_{|V(H)|}$ such that for every $i \in [|V(H)|]$, $G[V_i]$ is connected and the graph obtained from G after contracting the edges of each $G[V_i]$ is isomorphic to H. Following the version of the Flat Wall theorem in [146], Proposition 9 can be modified so as when it reports that K_t is a minor of G, it also outputs a minor-model of K_t in G.

B.7 Levelings and well-aligned flatness pairs

Let G be a graph and let $(W, \mathfrak{R})$ be either a flatness pair or a railed annulus flatness pair of G. If $(W, \mathfrak{R})$ is a flatness pair of G, let $\mathfrak{R} = (X, Y, P, C, \Gamma, \sigma, \pi)$, where (Γ, σ, π) is an Ω-rendition of $G[Y]$ and $\Gamma = (U, N)$ is a Δ-painting. If $(W, \mathfrak{R})$ is a railed annulus flatness pair of G, let $\mathfrak{R} = (X_1, Y_1, X_2, Y_2, Z_1, Z_2, \Gamma, \sigma, \pi)$, where (Γ, σ, π) is an (Ω_1, Ω_2)-rendition of $G[Y_1 \cap X_2]$ and $\Gamma = (U, N)$ is a Δ-painting, for some closed annulus Δ. In both cases, we define the *ground set* of W in $\mathfrak{R}$ to be the set $\mathsf{ground}_{\mathfrak{R}}(W) := \pi(N(\Gamma))$ and we refer to the vertices of this set as the *ground vertices* of the $\mathfrak{R}$-compass of W in G. Notice that $\mathsf{ground}_{\mathfrak{R}}(W)$ may contain vertices of $\mathsf{compass}_{\mathfrak{R}}(W)$ that are not necessarily vertices in $V(W)$.

Levelings. We define the $\mathfrak{R}$-*leveling* of W in G, denoted by $W_{\mathfrak{R}}$, as the bipartite graph where one part is the ground set of W in $\mathfrak{R}$, the other part is a set $\mathsf{vflaps}_{\mathfrak{R}}(W) = \{v_F \mid F \in \mathsf{flaps}_{\mathfrak{R}}(W)\}$ containing one new vertex v_F for each flap F of W in $\mathfrak{R}$, and, given a pair $(x, F) \in \mathsf{ground}_{\mathfrak{R}}(W) \times \mathsf{flaps}_{\mathfrak{R}}(W)$, the set $\{x, v_F\}$ is an edge of $W_{\mathfrak{R}}$ if and only if $x \in \partial F$. We call the vertices of $\mathsf{ground}_{\mathfrak{R}}(W)$ (resp. $\mathsf{vflaps}_{\mathfrak{R}}(W)$) *ground-vertices* (resp. *flap-vertices*) of $W_{\mathfrak{R}}$. Notice that the incidence graph of the plane hypergraph $(N(\Gamma), \{\tilde{c} \mid c \in C(\Gamma)\})$ is isomorphic to $W_{\mathfrak{R}}$ via an isomorphism that extends π and, moreover, bijectively corresponds cells to flap-vertices. If $(W, \mathfrak{R})$ is a flatness pair of G, this permits us to treat $W_{\mathfrak{R}}$ as a D-embedded graph, for some closed disk D, where $\mathsf{bd}(D) \cap W_{\mathfrak{R}}$ is the set $X \cap Y$. If $(W, \mathfrak{R})$ is a railed annulus flatness pair of G, we can treat $\mathcal{A}_{\mathfrak{R}}$ as a Δ-embedded graph, for some closed annulus Δ with boundaries B_1 and B_2, where $B_1 \cap \mathcal{A}_{\mathfrak{R}}$ is the set $X_1 \cap Y_1$ and $B_2 \cap \mathcal{A}_{\mathfrak{R}}$ is the set $X_2 \cap Y_2$.

Representations in flatness pairs and railed annulus flatness pairs. We denote by $W^{\bullet}$ the graph obtained from W if we subdivide *once* every edge of W that is short in $\mathsf{compass}_{\mathfrak{R}}(W)$. The graph $W^{\bullet}$ is a "slightly richer variant" of W that is necessary for our definitions and proofs, namely to be able to associate every flap-vertex of an appropriate subgraph of $W_{\mathfrak{R}}$ (that we will denote by R_W) with a non-empty path of $W^{\bullet}$, as we proceed to formalize. We say that $(W, \mathfrak{R})$ is *well-aligned* if the following holds:

> $W_{\mathfrak{R}}$ contains as a subgraph an r-wall R_W where $D(R_W) = D(W_{\mathfrak{R}})$ and $W^{\bullet}$ is isomorphic to some subdivision of R_W via an isomorphism that maps each ground vertex to itself.

Suppose now that the flatness pair $(W, \mathfrak{R})$ is well-aligned. We call the wall R_W in the above condition a *representation* of W in $W_{\mathfrak{R}}$.

Proposition 10 ([197]). *If a flatness pair $(W, \mathfrak{R})$ is regular, then it is also well-aligned. Moreover, there is an $\mathcal{O}(n)$-time algorithm that, given G and such a $(W, \mathfrak{R})$, outputs a representation R_W of W in $W_{\mathfrak{R}}$.*

Well-aligned railed annulus flatness pairs. We denote by $\mathcal{A}^{\bullet}$ the graph obtained from $\mathcal{A}$ if we subdivide *once* every edge in $E(\mathcal{A})$ that is short in $\mathsf{compass}_{\mathfrak{R}}(\mathcal{A})$. The graph $\mathcal{A}^{\bullet}$ is a "slightly richer variant" of $\mathcal{A}$ that is necessary for our definitions and proofs, namely to be able to associate every flap-vertex of an appropriate subgraph of $\mathcal{A}_{\mathfrak{R}}$ (that we will denote by $R_{\mathcal{A}}$) with a non-empty path of $\mathcal{A}^{\bullet}$, as we proceed to formalize. We say that $(\mathcal{A}, \mathfrak{R})$ is *well-aligned* if the following holds:

> $\mathcal{A}_{\mathfrak{R}}$ contains as a subgraph an (r, q)-railed annulus $R_{\mathcal{A}}$ where the first (resp. last) cycle of $\mathcal{A}_W$ is the same as the first (resp. last) cycle of $\mathcal{A}_{\mathfrak{R}}$ and $\mathcal{A}^{\bullet}$ is isomorphic to some subdivision of $R_{\mathcal{A}}$ via an isomorphism that maps each ground vertex to itself.

Suppose now that the railed annulus flatness pair $(\mathcal{A}, \mathfrak{R})$ is well-aligned. We call the wall $R_{\mathcal{A}}$ in the above condition a *representation* of $\mathcal{A}$ in $\mathcal{A}_{\mathfrak{R}}$. Note that, as $R_{\mathcal{A}}$ is a subgraph of $\mathcal{A}_{\mathfrak{R}}$, it is bipartite as well. The above property gives us a way to represent a flat wall by a wall of its leveling in a way that ground vertices are not altered.

Notice that both $\mathcal{A}_{\mathfrak{R}}$ and its subgraph $R_{\mathcal{A}}$ can be seen as Δ-embedded graphs where $B_1 \cap \mathcal{A}_{\mathfrak{R}} = B_1 \cap R_{\mathcal{A}}$ and these are both subsets of the vertex set of the first cycle of $\mathcal{A}_{\mathfrak{R}}$ (resp. of $R_{\mathcal{A}}$) and $B_2 \cap \mathcal{A}_{\mathfrak{R}} = B_2 \cap R_{\mathcal{A}}$ and these are both subsets of the vertex set of the last cycle of $\mathcal{A}_{\mathfrak{R}}$ (resp. of $R_{\mathcal{A}}$). This establishes a bijection δ from the set of cycles of $\mathcal{A}$ to the set of cycles of $R_{\mathcal{A}}$.

Let G be a graph, let $(\mathcal{A}, \mathfrak{R})$ be a well-aligned railed annulus flatness pair of G. We set $\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G)$ to be the graph obtained from G after replacing $\mathsf{Compass}_{\mathfrak{R}}(\mathcal{A})$ with $\mathcal{A}_{\mathfrak{R}}$. We now show the next result.

Lemma 12. *Let G, H be graphs and let $(\mathcal{A}, \mathfrak{R})$ be a well-aligned railed annulus flatness pair of G. Also, let $s_1, t_1, \ldots, s_k, t_k \in V(G)$ such that for every $i \in [k]$, $s_i, t_i \notin \mathsf{Compass}_{\mathfrak{R}}(\mathcal{A})$. Then there is a linkage L in G such that $T(L) = \{s_1, t_1, \ldots, s_k, t_k\}$ if and only if there is a linkage L' in $\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G)$ such that $L \equiv L'$.*

Proof. Let $P_1, \ldots, P_k$ be the paths of L in G. Since for every $i \in [k]$, $s_i, t_i \notin \mathsf{Compass}_{\mathfrak{R}}(\mathcal{A})$, no flap $F \in \mathsf{flaps}_{\mathfrak{R}}(\mathcal{A})$ does contain any endpoint of $P_1, \ldots, P_k$. Therefore, for every flap $F \in \mathsf{flaps}_{\mathfrak{R}}(\mathcal{A})$ there is at most one $i \in [k]$ such that P_i contains some vertex of $F \setminus \partial F$. For every $i \in [k]$, let $\mathcal{Q}_i = \{P_i \cap (F \setminus \partial F) \mid F \in \mathsf{flaps}_{\mathfrak{R}}(\mathcal{A})\}$. Observe that each $Q \in \mathcal{Q}_i$ corresponds to a single vertex v_Q of $\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G)$. Therefore, by replacing, for each $i \in [k]$, each $Q \in \mathcal{Q}_i$ by v_Q, we get a linkage L' of paths $P'_1, \ldots, P'_k$ in $\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G)$ that is equivalent to L. For the reverse implication, notice that every collection $P'_1, \ldots, P'_k$ of disjoint paths in $\mathsf{Leveling}_{(\mathcal{A},\mathfrak{R})}(G)$ corresponds to a collection $P_1, \ldots, P_k$ of disjoint paths in G, where for every $i \in [k]$, P_i is obtained by replacing each vertex v_F that is contained in P'_i by a path in F that connects the two neighbors of v_F in P'_i (these neighbors belong to ∂F). $\square$

C Missing complexity proofs

C.1 Hardness of Ordered Linkability

In this subsection we treat the parameterized complexity of Ordered Linkability which we restate below.

Ordered Linkability
Input: a graph G, $R \subseteq V(G)$, and a graph H where $k = \lvert H \rvert$.
Question: is R H-linked in G?

We consider the above problem in the case where the graph H is the 1-regular graph on $2k$ vertices. In this case, the task is, given a graph G, a subset of vertices $R \subseteq V(G)$, and an integer $k \geq 1$, decide whether for every sequence $s_1, \ldots, s_k, t_1, \ldots, t_k$ of distinct vertices of R, the graph G has pairwise disjoint vertex-disjoint paths between s_i and t_i, for $i \in [k]$.

Theorem 5. Ordered Linkability *cannot be solved in time* $\mathcal{O}_k(n^{\mathcal{O}(1)})$ *unless* $\mathsf{FPT} = \mathsf{W}[1]$.

Proof. Given a graph G, we use $\omega(G)$ to denote the maximum r such that G contains K_r as a subgraph. Recall that the Clique problem asks, given a graph G and a positive integer k, whether $\omega(G) \geq k$. It is well-known that Clique is $\mathsf{W}[1]$-hard [66] when parameterized by k. Furthermore, the optimization version is hard from the parameterized approximation viewpoint. In particular, by the breakthrough result of Lin [165], for any positive constant $c \geq 1$, no algorithm running in time $\mathcal{O}_k(n^{\mathcal{O}(1)})$ for a computable function $f(k)$ can distinguish between the cases $\omega(G) \geq k$ and $\omega(G) < k/c$, unless $\mathsf{FPT} = \mathsf{W}[1]$. We use this result for our reduction.

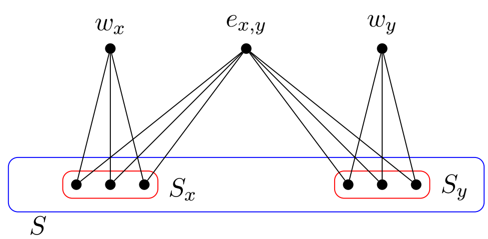

Figure 11: Construction of G'.

Consider an instance (G, k) of Clique. Without loss of generality, we assume that $k = 4r + 1$ for an integer $r \geq 2$. We construct the graph G' as follows (see Figure 11).

- For every vertex $x \in V(G)$, construct a set S_x of $(k-1)/2$ vertices (note that $k-1$ is even) and then form a clique from $S = \bigcup_{x \in V(G)} S_x$ by making the vertices pairwise adjacent.
- For every vertex $x \in V(G)$, construct a vertex w_x and make it adjacent to the vertices of S_x; denote $W = \{w_x : x \in V(G)\}$.
- For every edge $\{x, y\} \in E(G)$, construct a vertex e_{xy} and make it adjacent to the vertices of S_x and S_y; denote $L = \{e_{xy} : \{x, y\} \in E(G)\}$.

We set $R = W \cup L$ and $k' = \frac{1}{2}\binom{k}{2} + 1$; note that k' is an integer because k is odd and $k-1$ is divisible by 4.

First, we show that if G contains a clique of size k as a subgraph, then there are disjoint k'-tuples $(s_1, \ldots, s_{k'})$ and $(t_1, \ldots, t_{k'})$ of vertices of R such that G' has no vertex-disjoint paths between s_i and t_i, for $i \in [k']$.

We use the following observation.

Claim 4. *Let H be a complete graph with $k = 4r + 1$ vertices for $r \geq 1$. Then there is a partition of $E(H)$ into $\ell = (4r+1)r$ pairs $\{e_i, e'_i\}$ for $i \in [\ell]$ such that e_i and e'_i have no common endpoints.*

Proof of Claim 4. The proof is by induction on r. If $r = 1$, then the partition is shown in Figure 12 (a). If $r \geq 2$, then we select four arbitrary vertices v_1, v_2, v_3, v_4 of H and partition the edges incident to them using the pattern shown in Figure 12 (b). The remaining edges are partitioned into pairs using the inductive assumption.

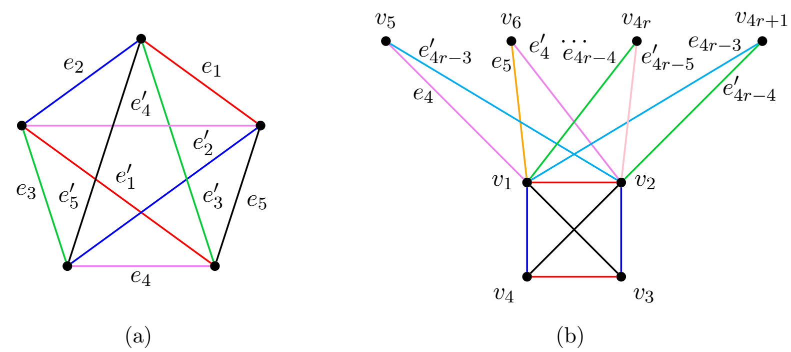

Figure 12: Pairing edges of H; the edges incident to v_3 and v_4 are not shown and are partitioned in pairs in the same pattern as the edges incident to v_1 and v_2.

$\diamond$

Suppose that G contains a clique H of size k as a subgraph. By Claim 4, the set of edges of H can be partitioned into $\ell = (4r+1)r = k' - 1$ pairs $\{e_i, e'_i\}$ for $i \in [\ell]$ such that e_i and e'_i have no common endpoints. Let $e_i = \{x_i, y_i\}$ and $e'_i = \{x'_i, y'_i\}$ for $i \in [\ell]$. We consider the vertices $s_i = e_{x_i y_i}$ and $t_i = e_{x'_i y'_i}$ of G' for $i \in [\ell]$. We also define $s_{k'} = w_x$ and $t_{k'} = w_y$ for arbitrary distinct $x, y \in V(H)$. The k'-tuples $(s_1, \ldots, s_{k'})$ and $(t_1, \ldots, t_{k'})$ are disjoint and consist of distinct vertices of R. Because e_i and e'_i have no common endpoints for every $i \in [\ell]$, any path between s_i and t_i in G' contains two edges incident to s_i and t_i, respectively. Because x and y are distinct, the same holds for any path in G' between $s_{k'}$ and $t_{k'}$. Suppose that G' has vertex-disjoint paths between s_i and t_i, for $i \in [k']$. Then the edges of the paths incident to the vertices $s_1, \ldots, s_{k'}$ and $t_1, \ldots, t_{k'}$ should form a matching. However, $N_{G'}(\{s_1, \ldots, s_{k'}\} \cup \{t_1, \ldots, t_{k'}\}) = \bigcup_{x \in V(H)} S_x$ and $|\bigcup_{x \in V(H)} S_x| = k(k-1)/2 < 2k'$. Therefore, by Hall's theorem [116], G' has no matching saturating the vertices of $\{s_1, \ldots, s_{k'}\} \cup \{t_1, \ldots, t_{k'}\}$, i.e., G' has no matching M such that the set of the endpoints of the edges in M contains $\{s_1, \ldots, s_{k'}\} \cup \{t_1, \ldots, t_{k'}\}$). Thus, there are no vertex-disjoint paths between s_i and t_i for $i \in [k']$ in G'.

Now we show that if G has no clique with at least $k/5$ vertices, then for all pairs of disjoint k'-tuples of distinct vertices $(s_1, \ldots, s_{k'})$ and $(t_1, \ldots, t_{k'})$, the graph G has vertex-disjoint paths between s_i and t_i, for all $i \in [k']$. Consider arbitrary $(s_1, \ldots, s_{k'})$ and $(t_1, \ldots, t_{k'})$ and set $X = \{s_1, \ldots, s_{k'}\} \cup \{t_1, \ldots, t_{k'}\}$. We prove the following claim.

Claim 5. *The graph G' has a matching M saturating every vertex of X.*

Proof of Claim 5. We set $Y = X \cap L$. For every $x \in V(G)$, let $S'_x \subset S_x$ be an arbitrary subset of S_x of size $(k-1)/2 - 1$. We define $S' = \bigcup_{x \in V(G)} S'_x$. We claim that $H = G[Y \cup S']$ has a matching M' saturating every vertex of Y. Because Y is an independent set, we can apply Hall's theorem [116] and observe that it is sufficient to show that for every $Z \subseteq Y$, $|N_H(Z)| \geq |Z|$. Consider a set $Z \subseteq Y$. Let $F = \{\{x, y\} \in E(G) \colon e_{xy} \in Z\}$ and let U be the set of vertices of G incident to the edges of F.

By Turan's theorem [205], $|F| \leq \left(1 - \frac{5}{k}\right)\frac{|U|^2}{2}$, because G has no clique with at least $k/5$ vertices. Therefore, $\sqrt{\frac{2|F|k}{k-5}} \leq |U|$. By the construction of H, we have that $|N_H(Z)| = |U|\frac{k-3}{2}$, because $|S'_x| = (k-3)/2$ for every $x \in X$. We obtain that $|N_H(Z)| \geq \sqrt{\frac{2|F|k}{k-5}} \cdot \frac{k-3}{2}$. Because $|Z| = |F|$, it is sufficient to show that $|Z| \leq \sqrt{\frac{2|Z|k}{k-5}} \cdot \frac{k-3}{2}$. This inequality is equivalent to $|Z| \leq \frac{k(k-3)^2}{2(k-5)}$ and holds because $\frac{k(k-3)^2}{2(k-5)} \geq \binom{k}{2} + 2 = 2k' \geq |Z|$. We conclude that H has a matching M' saturating every vertex of Y.

Note that for every $w_x \in X \cap W$, there is an adjacent vertex $v \in S_x \setminus S'_x$. By adding $\{w_x, v\}$ to M' for each $w_x \in X \cap W$, we obtain a matching M in G' saturating every vertex of X. This concludes the proof of the claim. ◇

Using Claim 5, we construct the paths between s_i and t_i in G' as follows. Let $i \in [k']$. The matching M contains the edges $\{s_i, u\}$ and $\{t_i, v\}$, for some $u, v \in S$. Because S is a clique, (s_i, u, v, t_i) is an (s_i, t_i)-path. Since M is a matching, all these paths are vertex-disjoint.

To conclude the proof, note that the existence of an algorithm solving Ordered Linkability in time $\mathcal{O}_k(n^{\mathcal{O}(1)})$ would imply that there is an algorithm distinguishing the cases $\omega(G) \geq k$ and $\omega(G) < k/5$ in time $f(\frac{1}{2}\binom{k}{2} + 1) \cdot n^{\mathcal{O}(1)}$, contradicting the result of Lin [165]. □

C.2 Monochromatic Path Topological Minor is W[1]-hard on planar graphs

Topological minor models. Given a graph G and a graph H, we say that H is a *topological minor* of G if there is an injection $\varphi : V(H) \to V(G)$ and a function ψ mapping the edges of H to paths of G such that

- For every distinct e_1, e_2, $\psi(e_1)$ and $\psi(e_2)$ are internally vertex disjoint paths of G and
- For every $e = \{x, y\} \in E(G)$, $\psi(e)$ is a path joining $\varphi(x)$ and $\varphi(y)$.

Given the above, we say that H is a *topological minor* of G *via the pair* (φ, ψ).

We consider the following problem:

> Monochromatic Path Topological Minor
> *Input*: multicolored graph $(G, X_1, \ldots, X_z)$ a graph H and a coloring function $\lambda : V(H) \to [z]$.
> *Question*: does G contain H as a topological minor via a pair (φ, ψ) where
> - for every $x \in V(H)$, $\varphi(x) \in X_{\lambda(x)}$ (i.e., the image of x carries the color of x) and
> - every $e \in E(H)$, there is some $i \in [z]$ such that $V(\psi(e)) \subseteq V_i$ (i.e., the path $\psi(e)$ is monochromatic)

Theorem 6. Monochromatic Path Topological Minor*, when parameterized by* $k = |H|$*, is* W[1]*-hard on planar graphs, even when* $z = 4$ *and* H *is the* $(k \times k)$*-grid.*

Proof. We give a parameterized reduction from the following problem:

> Grid Tiling
> *Input*: two integers $D, k \in \mathbb{N}$ and a function $M : [k]^2 \to 2^{[D]^2}$.
> *Question*: is there a function $s : [k]^2 \to [D]^2$ such that
> - for every $(i, j) \in [k]$, $s(i, j) \in M(i, j)$,
> - for every $i \in [k]$, all first coordinates of the pairs in $\{s(i, j) \mid j \in [k]\}$ are equal, and
> - for every $j \in [k]$, all second coordinates of the pairs in $\{s(i, j) \mid i \in [k]\}$ are equal

Intuitively, one may see the input of the Grid Tiling problem as the assignment of pairs in $[D]^2$ to the cells of a $k \times k$-matrix an the question is whether it is possible to choose one pair from each cell so that, in the occurring $k \times k$-matrix, vertical pairs agree in the 1st coordinate and horizontal pairs agree in the 2nd coordinate. It is known (see [56]) that Grid Tiling,when parameterized by k, is W[1]-hard. Given an instance (D, k, M) of Grid Tiling, we build an instance $(G, B, C, O, T, H, \lambda)$ of Monochromatic Path Topological Minor as follows.

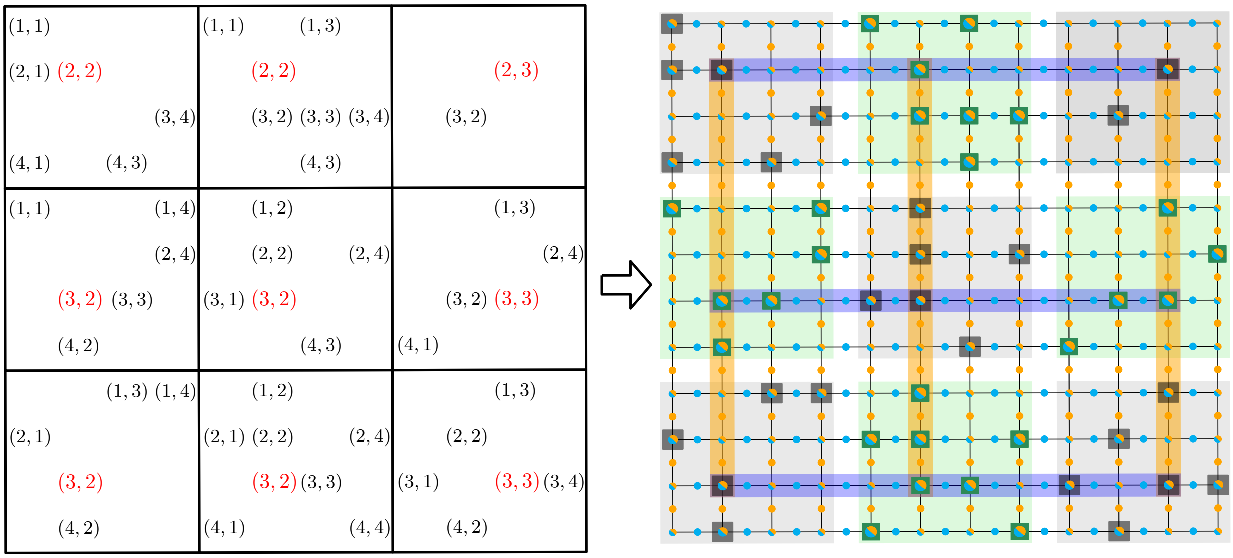

Figure 13: An example of the reduction of Theorem 6. On the left an instance of Grid Tiling is depicted where $D = 4$ and $k = 3$. On the right the corresponding instance $(G, B, C, O, T, H, \lambda)$ of Monochromatic Path Topological Minor is depicted, where H is the $(k \times k)$-grid. The solution for the Grid Tiling, depicted by the red pairs, corresponds to a realization of the $(k \times k)$-grid on the right, depicted by the (long) orange and blue rectangles.

Consider a $(kD \times kD)$-grid Γ. Also define $\{\Gamma_{i,j} \mid (i, j) \in [k]^2\}$ to be the (unique) collection of k^2 pairwise vertex disjoint $(D \times D)$-grids of Γ. We define the black and green color sets, namely B and C so that for every $(i, j) \in [k]^2$ we include, for every pair $(x, y) \in M(i, j)$, the intersection vertex of the x-th column and the y-th row of $\Gamma_{i,j}$ in the set B (if $i + j = 0 \pmod 2$) or in the set C (if $i + j = 1 \pmod 2$).

The graph G is obtained by subdividing every vertex of Γ once. Clearly, G consists of kD vertical paths and kD horizontal paths. The orange and the turquoise color sets, namely O anf T are defined so that O contains every vertex of a vertical path and T contains every vertex of an horizontal path. We now consider the $(k \times k)$-grid H and λ is a proper two coloring of H in black and green.

It now remains to see that (D, k, M) is a yes-instance of Grid Tiling if and only if the tuple $(G, B, C, O, T, H, \lambda)$ is a yes-instance of Monochromatic Path Topological Minor. Just observe that the orange (resp. turquoise) colors force all vertical (resp. horizontal) paths of H to be mapped to vertical (resp. horizontal) paths and that the black and green colors force each edge of H to be mapped to a path in some neighboring $\Gamma_{i,j}$'s joining a black vertex and a green vertex of G. □