Model Checking Disjoint-Paths Logic on Topological-Minor-Free Graph Classes

Nicole Schirrmacher
Bremen Universität
Bremen, Germany
schirrmacher@uni-bremen.de

Sebastian Siebertz
Bremen Universität
Bremen, Germany
siebertz@uni-bremen.de

Giannos Stamoulis
University of Warsaw
Warsaw, Poland
giannos.stamoulis@mimuw.edu.pl

Dimitrios M. Thilikos
LIRMM, Univ Montpellier, CNRS
Montpellier, France
sedthilk@thilikos.info

Alexandre Vigny
Université Clermont Auvergne
Clermont-Ferrand, France
alexandre.vigny@uca.fr

ABSTRACT

Disjoint-paths logic, denoted FO + DP, extends first-order logic (FO) with atomic predicates $\mathrm{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$, expressing the existence of internally vertex-disjoint paths between x_i and y_i, for $1 \leqslant i \leqslant k$. We prove that for every graph class excluding some fixed graph as a topological minor, the model checking problem for FO + DP is fixed-parameter tractable. This extends the model checking algorithm of Golovach et al. [SODA 2023] for FO + DP for minor-closed graph classes. It also essentially settles the question of tractable model checking for this logic on subgraph-closed classes, since the problem is hard on subgraph-closed classes not excluding a topological minor (assuming a further mild condition on efficiency of encoding).

CCS CONCEPTS

• **Theory of computation** → **Finite Model Theory**; **Graph algorithms analysis**; **Fixed parameter tractability**.

KEYWORDS

Model checking, Topological-minor-free graphs, First-order logic, Disjoint paths, Parameterized algorithms

ACM Reference Format:
Nicole Schirrmacher, Sebastian Siebertz, Giannos Stamoulis, Dimitrios M. Thilikos, and Alexandre Vigny. 2024. Model Checking Disjoint-Paths Logic on Topological-Minor-Free Graph Classes. In *39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '24), July 8–11, 2024, Tallinn, Estonia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3661814.3662089

1 INTRODUCTION

The model checking problem for a logic $\mathcal{L}$ gets as input a structure and an $\mathcal{L}$-sentence and the question is to decide whether the sentence is true in the structure. Therefore, the model checking problem for $\mathcal{L}$ subsumes the decision problem for all $\mathcal{L}$-definable problems. For this reason, tractability results for model checking problems are often called *algorithmic meta theorems*, as they explain and unify tractability for all problems definable in the considered logic $\mathcal{L}$. A prime example of an algorithmic meta theorem is Courcelle's theorem [6] stating that every problem definable in monadic second-order logic (MSO) can be solved in linear time on every graph class with bounded treewidth. An algorithmic meta theorem not only provides a quick way to establish tractability of problems but in many cases, its proof distills the essence of the algorithmic techniques required to solve them. Courcelle's theorem captures the decomposability of MSO-definable problems and a corresponding dynamic programming approach over tree decompositions of small width. Courcelle's theorem was extended to graph classes with bounded cliquewidth [7] and it is known that these are essentially the most general graph classes on which we can expect efficient MSO model checking [24, 34].

Also, the first-order (FO) model checking problem has received considerable attention in the literature, see e.g. [4, 5, 10–12, 14, 16, 17, 20, 22, 23, 31, 33, 43]. Seese [43] was the first to study the FO model checking problem on classes of graphs with bounded maximum degree. The essence of his approach was to exploit the locality properties of FO, and, in some form, locality-based methods constitute the basis of all of the mentioned model checking results for FO. Grohe, Kreutzer, and Siebertz [31] showed that the FO model checking problem is fixed-parameter tractable on nowhere dense graph classes, and when considering subgraph-closed classes this result is optimal [14]. In a recent breakthrough, it was shown that the problem is fixed-parameter tractable on classes with bounded twin width [5] and, moreover, this result is optimal when considering classes of ordered graphs [4].

While FO can express many interesting algorithmic properties, it also has some shortcomings. In particular, it cannot count and it can express *only* local properties. The first shortcoming led to the study of counting extensions, see e.g. [2, 13, 32, 35, 36, 39, 45]. These meta theorems, in essence, build again on locality properties that are shared by FO with counting extensions. The second shortcoming classically led to the study of transitive-closure logics and fixed-point logics, see e.g. [15, 27, 29, 37]. However, even the model checking problem for the very restricted monadic transitive-closure logic TC^1 is most probably not fixed-parameter tractable even on planar graphs of maximum degree three [29]. Furthermore, these logics still fall short of being able to express many interesting algorithmic problems, involving "non-local" queries

LICS '24, July 8–11, 2024, Tallinn, Estonia

ACM ISBN 979-8-4007-0660-8/24/07
https://doi.org/10.1145/3661814.3662089

studied in contemporary algorithmics. A classic problem of this kind is the Disjoint Paths problem: *Given a graph G and a set $\{(s_1, t_1), \ldots, (s_k, t_k)\}$ of pairs of terminals, the question is whether G contains vertex-disjoint paths joining s_i and t_i for* $1 \leqslant i \leqslant k$. Clearly, asking for a path joining pairs of terminals is not a local query as the length of such a path is unbounded.

These shortcomings have recently led to the study of new logics whose expressive power lies between FO and MSO. The *compound logic* combines FO and MSO and is designed to express a wide range of graph modification problems [18]. Its model checking problem is fixed-parameter tractable on classes of graphs with excluded minors and at its core, it combines the locality method for FO with the irrelevant vertex technique to eventually reduce the problem to a graph of bounded treewidth. Another recently introduced logic is *separator logic*, which extends FO by connectivity after vertex deletions [3, 42] and which can express other interesting algorithmic problems such as elimination distance to FO-definable graph classes. It was proven in [40] that, for this logic, the model checking problem is fixed-parameter tractable on classes excluding a topological minor, and for subgraph-closed classes, this result cannot be extended to more general classes (assuming a further condition on the efficiency of encoding[1]). This meta theorem essentially combines classical FO model checking with dynamic programming over decompositions into unbreakable parts. The required decompositions are provided by a result of Cygan, Lokshtanov, Pilipczuk, Pilipczuk, and Saurabh [9]. A key observation is that over highly connected graphs, connectivity can be reduced to queries of paths of bounded length, and therefore becomes in fact FO expressible.

In this work, we study *disjoint-paths logic*, which was also introduced in [42] as an extension of separator logic. Disjoint-paths logic, denoted FO+DP, extends first-order logic with atomic predicates $\mathsf{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$ expressing the existence of internally vertex-disjoint paths between x_i and y_i, for $1 \leqslant i \leqslant k$. It can express many interesting algorithmic problems, such as the disjoint paths problem, minor containment, topological minor containment, $\mathcal{F}$-topological minor deletion, and many more (see the appendix of [25, 26] for several examples indicating the expressibility potential of FO+DP). It was already shown in [25, 26] that the model checking problem for disjoint-paths logic is fixed-parameter tractable on classes with excluded minors. The essence of the meta theorem of [25, 26] is again the irrelevant vertex technique.

Our results. In this work, we prove that for every graph class excluding a fixed graph as a topological minor, the model checking problem for FO+DP is fixed-parameter tractable. More precisely, we prove the following result.

Theorem 1.1. *Let $\mathscr{C}_H$ be the class of graphs excluding a fixed graph H as a topological minor. Then, there is an algorithm that, given $G \in \mathscr{C}_H$ and an* FO+DP *formula $\varphi(\bar{x})$ and $\bar{v} \in V(G)^{|\bar{x}|}$, decides whether $G \models \varphi(\bar{v})$ in time $f(\varphi) \cdot |V(G)|^3$, where f is a computable function depending on H.*

[1]We say that a class $\mathscr{C}$ *admits efficient encoding of topological minors* if for every graph H there exists $G \in \mathscr{C}$ such that H is a topological minor of G, and, given H, such G together with a suitable topological minor model can be computed in time polynomial in $|H|$.

This essentially settles the question of tractable model checking for FO+DP on subgraph-closed classes, since it is already known (see [42]) that the model checking problem for separator logic on subgraph-closed classes that do not exclude a topological minor and admit efficient encoding is as hard as model checking for FO on general graphs, i.e., it is AW[$\star$]-hard when parameterized by the formula size. Since disjoint-paths logic extends separator logic we conclude the same hardness result for disjoint-paths logic.

Beyond the direct application to classes that exclude a topological minor, Theorem 1.1 has further applications for the design of parameterized algorithms as follows. Suppose that Π is some FO+DP-expressible parameterized problem whose instance is a graph G and some integer k. Suppose also Π satisfies the following property: *The topological minor containment of some particular graph (whose size depends on k) in the input graph G directly certifies a* yes*- or a* no*-answer.* Then, because of Theorem 1.1, Π admits a (parameterized) $O(f(k) \cdot n^3)$ time algorithm, for some function f.

As a characteristic example of the above, we mention the $\mathcal{F}$-Topological Minor Deletion problem, defined for some finite collection $\mathcal{F}$ of graphs: *Given a graph G and an integer k, the question is whether G contains a set S of k vertices whose removal from G gives a graph excluding all graphs in $\mathcal{F}$ as topological minors.* Fomin, Lokshtanov, Panolan, Saurabh, and Zehavi proved in [19] that $\mathcal{F}$-Topological Minor Deletion admits a time $O(f(k) \cdot n^4)$ algorithm, for some function f.

As the presence of a big enough (as a function of k and $\mathcal{F}$) clique as a topological minor implies directly that the instance (G, k) is a no-instance, the problem reduces to graphs excluding some clique as a topological minor and can be solved, using Theorem 1.1, in time $O(f(k) \cdot n^3)$, for some function f.

Our techniques. Our meta theorem combines the approaches of both [25, 26] and [40]. We start by decomposing the input graph into unbreakable parts, using the decomposition of Cygan, Lokshtanov, Pilipczuk, Pilipczuk, and Saurabh [9]. On each part, we first apply a subroutine of the algorithm of [25, 26], which either reports the existence of a large clique-minor or solves the model checking problem. The emerging challenge is to answer the question

"what to do with a big clique-minor?"

on the meta-algorithmic level. For particular problems, the answer to this question was given by the "*generic folio technique*". This technique was introduced by Robertson and Seymour in [41] for the Disjoint Paths problem and was later used by Grohe, Kawarabayashi, Marx, and Wollan [30], for the Topological Minor Checking problem (see also [19]).

The combinatorial assumption of unbreakability implies that for every small-enough separator at most one side is large-enough. In the presence of a large clique minor, a big part of the clique should reside in the large side. This, in turn, permits the application of the generic folio technique, which guarantees that the separator vertices are "highly connected" in the large side.

In this paper, we generalize the above approach for every $\varphi(\bar{x})$ of FO+DP. In particular, we prove that every formula in this logic on unbreakable graphs with a large clique-minor is equivalent to a plain first-order formula $\psi(\bar{x})$. In other words, we prove that these two combinatorial conditions permit a syntactic use of the generic folio technique that implies an expressibility collapse of FO+DP to FO.

We next use a dynamic programming approach over tree-decomposable graphs to combine the solutions of the unbreakable parts into a global solution. Our dynamic programming approach deviates from the standard approach for MSO or FO, which is based on the computation and combination of types building on the Feferman-Vaught decomposition theorem for these logics. A similar decomposition also exists for disjoint-paths logic, however, when decomposing over larger separators, the number of disjoint paths that need to be queried increases. As a consequence, the decomposition theorem cannot be applied unboundedly often in the dynamic programming procedure. Instead, our approach uses the ideas of [38] and [30], where large structures are replaced by small structures of the same type. However, since the satisfiability problem already of plain FO is undecidable, it is not possible to find representative structures of the same type whose size is bounded by any computable function. However, since we can solve the model checking problem on each bag of the tree decomposition locally, using either the aforementioned exressibility collapse or the subroutine of the algorithm of [25, 26], we can compute a small representative of the game graph of the model checking game (enriched with further vertices to ensure the same connectivity for a fixed number of disjoint paths of the substructure). With an appropriate rewriting of formulas, this structure can serve as a small representative structure that can be maintained over the dynamic programming procedure. This generalizes the results of [42] for separator logic to disjoint-paths logic, in particular handling the case of topological-minor-free graphs that are unbreakable and contain large clique-minors. Again, as it is the case also in [42], working on topological-minor-free graphs appears to be the appropriate condition for such a dynamic programming procedure to work.

Let us also comment on why we failed to use the framework of [40] to combine the solutions of the unbreakable parts into a global solution, but need to fall back to the dynamic programming approach. The obstacle arises from the fact that the tree decomposition into unbreakable parts (with parts of unbounded size) builds on an underlying tree with unbounded branching degree. This requires a "simultaneous" dynamic programming step when progressing from the children of a node to the node itself. Such a step was possible in the case of separator logic, and in fact for all properties that can be encoded as FO formulas with MSO subformulas that are essentially restricted to speak about the tree order of the tree decomposition into unbreakable parts (this logic is called $\mathsf{FO}(\mathsf{MSO}(\preceq, A) \cup \Sigma)$ in [40]). We were not able to combine the disjoint-paths queries simultaneously over unboundedly many child nodes of a node due to the many possibilities to route disjoint paths through the children (compare with the mentioned problems with the Feferman-Vaught decomposition theorem). This is in contrast to the comparatively simple connectivity queries of separator logic, where the solution for the children is unique (since we ask simple connectivity queries) and can be encoded into the torso of a bag. Note also that these problems cannot be handled by going to nice tree decompositions with an underlying binary tree. This translation requires a copying of bags (which are unboundedly large) and does not allow an encoding of the decomposition accessible to the logic.

A result that is weaker than ours follows also from the work of Lokshtanov, Ramanujan, Saurabh, and Zehavi [38] who proved the following result: For every CMSO sentence φ, where CMSO is the extension of MSO with modulo counting predicates, if there is an $O(n^d)$ algorithm (for $d \geqslant 5$) to test the truth of φ over unbreakable graphs, then there is an $O(n^d)$ algorithm to test the truth of φ over all graphs. Since FO+DP is a fragment of MSO and our algorithm for FO+DP runs in time $O(n^3) \subseteq O(n^5)$, our results on unbreakable graphs together with the result of [38] implies the *existence* of an $O(n^5)$ model checking algorithm for every fixed FO+DP sentence φ on every class excluding a topological minor. The main caveat of the proof based on the result of [38] is that it is non-constructive and we can only conclude the existence of an efficient model checking algorithm. *Au contraire*, our algorithm is fully constructive.

Organization. We give background on graphs, minors and topological minors, unbreakability, and disjoint-paths logic in Section 2. We prove the collapse of FO+DP to plain FO on unbreakable graphs with large clique-minors in Section 3. To lift from unbreakable graphs to general graphs we show how to combine solutions and apply dynamic programming in Section 4.

2 GRAPHS, UNBREAKABILITY AND DISJOINT-PATHS LOGIC

All graphs in this paper are finite, undirected graphs without loops. We write $V(G)$ for the vertex set and $E(G)$ for the edge set of a graph G. We write $\|G\|$ for $|V(G)| + |E(G)|$. Let G be a graph and $u, v \in V(G)$. A *u-v-path* P in G is a sequence $v_1, \ldots, v_k$ of pairwise different vertices such that $\{v_i, v_{i+1}\} \in E(G)$ for all $1 \leqslant i < k$ and $v_1 = u$ and $v_k = v$. The vertices $v_2, \ldots, v_{k-1}$ are the *internal vertices* of P and the vertices u and v are its *endpoints*. Two vertices u, v are *connected* if there exists a path with endpoints u, v. A graph is connected if any two of its vertices are connected. Two paths P, Q are *internally vertex-disjoint* if no vertex of one path appears as an internal vertex of the other path. For a vertex subset $X \subseteq V(G)$, we write $G[X]$ for the subgraph of G induced by X.

An acyclic and connected graph T is a *tree*. Assigning a distinguished vertex r as a root of a tree, we define a tree order $\preceq_T$ on $V(T)$ by setting $x \preceq_T y$ if x lies on the unique path (possibly of length 0) from y to r (intuitively, x is "closer" to r than y). If $x \preceq_T y$ we call x an *ancestor* of y in T. Note that by this definition, every node is an ancestor of itself. We drop the subscript T if it is clear from the context. The *parent* of a non-root node x of T, is the (unique) node $w \in V(T)$ such that $w \preceq_T x$ and, for every $y \in V(T)$, $y \preceq_T w$. We write $\mathrm{parent}(x)$ for the parent of a non-root node x of T, and $\mathrm{children}(x)$ for the set of nodes $z \in V(T)$ such that $\mathrm{parent}(z) = x$. We define $\mathrm{parent}(r) = \bot$.

Minors and topological minors. A *minor model* of a graph H in a graph G is a mapping μ that maps vertices of H to connected subgraphs of G and edges of H to edges of G such that for every $v, u \in V(H)$, where $v \neq u$, the graphs $\mu(v)$ and $\mu(u)$ are vertex disjoint, and for every $\{u, v\} \in E(H)$, the edge $\mu(\{u, v\})$ is incident to a vertex $v' \in V(\mu(v))$ and a vertex $u' \in V(\mu(u))$. A graph H is a *minor* of G if there is a minor model of H in G. Given a minor model μ of H in G, we refer to the sets $V(\mu(v))$, for $v \in V(H)$, as the *branch sets* of μ in G.

A *topological minor model* of a graph H in a graph G is an injective mapping η that maps vertices of H to vertices of G and edges of H to pairwise internally vertex-disjoint paths in G so that for every $\{u, v\} \in E(H)$ the path $\eta(\{u, v\})$ has the endpoints $\eta(u)$ and $\eta(v)$. Given a topological minor model η of H in G, we refer to $\eta(v)$, for $v \in V(H)$, as *principal vertices* of η in G. A graph H is a *topological minor* of G if there is a topological minor model of H in G.

We call a graph G *H-minor-free*, and *H-topological-minor-free*, respectively, if H is not a minor, or topological minor of G. We call a class $\mathscr{C}$ of graphs *(topological-)minor-free* if there exists a graph H such that every member of $\mathscr{C}$ is H-(topological-)minor-free.

Folios. In our model checking algorithm, in order to deal with unbreakable parts with large clique-minors, as well as to perform our dynamic programming procedure via small representative replacement, we need to express the variety of graphs that can be routed through a given graph as topological minors. This is done by using the notion of *folio* of a graph G, which is defined as the set of all graphs that are topological minors of G.

We write $\mathrm{is}(H)$ for the number of isolated vertices of H. For $\delta \in \mathbb{N}$, the *δ-folio* of G is the set of topological minors H of G with $|E(H)| + \mathrm{is}(H) \leqslant \delta$. Note that every graph H in the δ-folio has at most 2δ vertices and that the size of the δ-folio of G depends only on δ (we identify isomorphic graphs). A *rooted graph* is an undirected graph G with a distinguished set $R(G) \subseteq V(G)$ of *root vertices* and an injective mapping $\rho\colon R(G) \to \mathbb{N}$ assigning a distinct positive integer label to each vertex of $R(G)$.

A rooted graph H is a *topological minor* of a rooted graph G if there is a topological minor model η of H in G such that $\rho_G(\eta(v)) = \rho_H(v)$, for all $v \in V(H)$, where ρ_G and ρ_H are the injective mappings assigning distinct positive integer labels to the vertices of $R(G)$ and $R(H)$, respectively.

There is a bounded number x of possible undirected graphs on $R(G)$. For each such graph X, we write $G + X$ for the graph obtained from G by setting $G[R(G)] = X$. For every different such graph X, the rooted graph $G + X$ may have a different δ-folio. The x-tuple of all possible $G + X$ δ-folios is the *extended δ-folio of G*.

Let G be a rooted graph and let w be a weight function assigning a positive integer to each vertex of $V(G)$. The *w-bounded* δ-folio of G is the subset of the δ-folio of G containing those graphs H that have a model η satisfying the additional requirement that for every $v \in R(H)$, the degree of v in H is at most $w(\eta(v))$.

A δ-folio of a graph is *generic* if it contains every rooted graph H with $E(H) + \mathrm{is}(H) \leqslant \delta$ and $\rho(R(H)) \subseteq \rho(R(G))$. We call it *rooted-generic* if it contains every such graph H with the additional requirement that every vertex of H is rooted. Note that hence a rooted-generic δ-folio is generic, but not necessarily every generic δ-folio is rooted-generic. We define the notions of generic and rooted-generic analogously for w-bounded folios.

Tree decompositions. A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \mathrm{bag})$, where T is a rooted tree and $\mathrm{bag}\colon V(T) \to 2^{V(G)}$ is a mapping assigning to each node x of T its *bag* $\mathrm{bag}(x)$, which is a subset of vertices of G such that the following conditions are satisfied:

(1) For every vertex $u \in V(G)$, the set of nodes $x \in V(T)$ with $u \in \mathrm{bag}(x)$ induces a nonempty and connected subtree of T.
(2) For every edge $\{u, v\} \in E(G)$, there exists a node $x \in V(T)$ with $\{u, v\} \subseteq \mathrm{bag}(x)$.

Recall that if r is the root of T, then we have $\mathrm{parent}(r) = \perp$. We define $\mathrm{bag}(\perp) = \emptyset$.

To define additional notions on tree decompositions and prove tools for the dynamic programming part of our model checking algorithm, we need to provide extra definitions concerning a node of the tree of tree decomposition. For a node $x \in V(T)$, we define the *adhesion* of x as $\mathrm{adh}(x) \coloneqq \mathrm{bag}(\mathrm{parent}(x)) \cap \mathrm{bag}(x)$; the *margin of x as* $\mathrm{mrg}(x) \coloneqq \mathrm{bag}(x) \setminus \mathrm{adh}(x)$; the *cone at x* as $\mathrm{cone}(x) \coloneqq \bigcup_{y \succeq_T x} \mathrm{bag}(y)$; and the *component at x* as $\mathrm{comp}(x) \coloneqq \mathrm{cone}(x) \setminus \mathrm{adh}(x)$. We refer to Figure 1 for a visualization of these definitions.

The *adhesion* of a tree decomposition $\mathcal{T} = (T, \mathrm{bag})$ is defined as the largest size of an adhesion, that is, $\max_{x \in V(T)} |\mathrm{adh}(x)|$.

A tree decomposition $\mathcal{T} = (T, \mathrm{bag})$ is *regular* if for every non-root node $x \in V(T)$

(1) the margin $\mathrm{mrg}(x)$ is nonempty;
(2) the graph $G[\mathrm{comp}(x)]$ is connected; and
(3) every vertex of $\mathrm{adh}(x)$ has a neighbor in $\mathrm{comp}(x)$.

Unbreakability. A *separation* in a graph G is a pair (G_1, G_2) of subgraphs $G_1, G_2 \subseteq G$ such that $V(G_1) \cup V(G_2) = V(G)$ and there is no edge with one endpoint in $V(G_1) \setminus V(G_2)$ and the other endpoint in $V(G_2) \setminus V(G_1)$. The *separator* of G is the intersection $V(G_1) \cap V(G_2)$ and the *order* of a separation is the size of its separator.

For $q, k \in \mathbb{N}$, a subgraph H in a graph G is *(q, k)-unbreakable* if for every separation (G_1, G_2) of G of order at most k, we have

$$|V(G_1) \cap V(H)| \leqslant q \qquad \text{or} \qquad |V(G_2) \cap V(H)| \leqslant q.$$

The notion of unbreakability can be lifted to tree decompositions by requiring it from every individual bag.

Definition 2.1. *Fix $q, k \in \mathbb{N}$. A tree decomposition $\mathcal{T} = (T, \mathrm{bag})$ of a graph G is* strongly (q, k)-unbreakable *if for every $x \in V(T)$, $G[\mathrm{bag}(x)]$ is (q, k)-unbreakable in $G[\mathrm{cone}(x)]$.*

We will use the following result from [9].

Theorem 2.2 ([9]). *There is a function $q(k) \in 2^{O(k)}$ such that for every graph G and $k \in \mathbb{N}$, there exists a strongly $(q(k), k)$-unbreakable tree decomposition of G of adhesion at most $q(k)$. Moreover, given G and k, such a tree decomposition can be computed in time $2^{O(k^2)} \cdot |G|^2 \cdot \|G\|$.*

Given any strongly (q, k)-unbreakable decomposition we can refine it so that it becomes regular. Hence, we may assume that the tree decompositions constructed by the algorithm of Theorem 2.2 are regular.

We will replace cones with small representative graphs and need to take care that we still exclude a topological minor. This is proved in the following lemma.

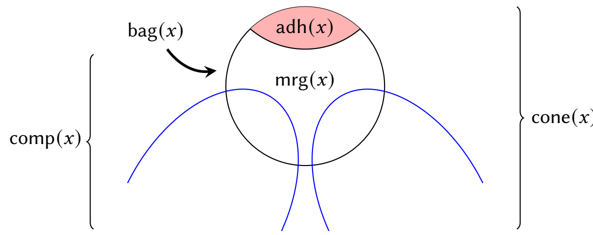

Figure 1: An illustration of the notions of adhesion, margin, cone, and component of a node $x \in V(T)$.

Lemma 2.3. *Let $\mathcal{T} = (T, \mathrm{bag})$ be a regular tree decomposition of a graph G, say of adhesion a. Suppose further that G is K_t-topological-minor-free. Let x be a non-root node in $V(T)$ and let $R = \mathrm{adh}(x)$. Let H with $R \subseteq V(H)$ be a graph with the same extended a-folio with respect to R as $G[\mathrm{cone}(x)]$ and assume $|V(H)| \leqslant c$, for some constant c. Let G' be the graph that is obtained from G by replacing $G[\mathrm{cone}(x)]$ with H. Then G' is $K_{t'}$-topological-minor-free, where $t' = \max(t, 2a+2, c)$.*

Proof. See Figure 2 for an illustration of the graphs and the vertex sets in the statement of Lemma 2.3. Assume we can find a topological minor model η of $K_{t'}$ in G'. We first show that all principal vertices of η must be completely in H or completely in $G' - H$. Assume towards a contradiction that this is not the case. Without loss of generality, we can assume that there is a principal vertex v in H, such that at least half of the other principal vertices lie in $G' - H$. Then at least $\lfloor t'/2 \rfloor \geqslant a+1$ disjoint paths from v to the other principal vertices must cross the adhesion. This is not possible, as the adhesion has size at most a.

Now, if all principal vertices lie in H, the model is trivially bounded in size by c, since H has at most c vertices. On the other hand, if all principal vertices lie in $G' - H$, then we cannot find a model with more than t vertices. The reason for this is that all paths connecting the principal vertices going through H can be replaced by paths in $G[\mathrm{cone}(y)]$, since H and $G[\mathrm{cone}(y)]$ have the same extended a-folio. Thus, the model also exists in G, a contradiction. □

Let $\mathcal{T} = (T, \mathrm{bag})$ be a regular tree decomposition of a graph G. For every $x \in V(T)$, let A_x be the set of all $z \in \mathrm{children}(x)$ such that there is no $z' \in A_x$ such that $z' \neq z$ and $\mathrm{adh}(z) \subseteq \mathrm{adh}(z')$.

Lemma 2.4. *There is some function $f : \mathbb{N}^2 \to \mathbb{N}$, such that for all $q, k \in \mathbb{N}$, if $\mathcal{T} = (T, \mathrm{bag})$ is a regular tree decomposition of a graph G, say of adhesion a, and, for every $x \in V(T)$, G'_x is the graph obtained from $G[\mathrm{bag}(x)]$ after attaching in every $\mathrm{adh}(z), z \in A_x$, a graph H_z of at most c vertices with the same a-folio as $\bigcup_{\mathrm{adh}(y)\subseteq \mathrm{adh}(z)} G[\mathrm{cone}(y)]$, the following holds: if $G[\mathrm{bag}(x)]$ is (q,k)-unbreakable in $G[\mathrm{cone}(x)]$, then G'_x is $(f(c,q),k)$-unbreakable in G'_x.*

Proof. We set $g(x) = \binom{x}{\lfloor \frac{x}{2} \rfloor}$ and $f(c,q) = q + c \cdot g(q)$. Assume that there is a separation (A, B) of G'_x of order at most k with $|V(A)|, |V(B)| > f(c,q)$. Let A' and B' be obtained from A and B by removing all vertices that do not belong to $\mathrm{bag}(x)$. These can be at most $c \cdot g(|V(A')|)$ and $c \cdot g(|V(B')|)$, respectively. Since $|V(A)| \leqslant |V(A')| + c \cdot g(|V(A')|)$ and $|V(B)| \leqslant |V(B')| + c \cdot g(|V(B')|)$ and $|V(A)|, |V(B)| > f(c,q)$, we have that $|V(A')|, |V(B')| > q$. We will show that there is a separation (A'', B'') of $G[\mathrm{cone}(x)]$ of order at most k with $A' \subseteq A''$ and $B' \subseteq B''$. Indeed, let (A'', B'') be a separation of $G[\mathrm{cone}(x)]$ of minimum order with $A' \subseteq A''$ and $B' \subseteq B''$. Since, $\bigcup_{\mathrm{adh}(y)\subseteq \mathrm{adh}(z)} G[\mathrm{cone}(y)]$ and H_z have the same a-folio, the size of the part of $V(A'' \cap B'')$ that is contained in $V(H_z) \setminus \mathrm{adh}(z)$ is at most $|V(A \cap B \cap \bigcup_{\mathrm{adh}(y)\subseteq \mathrm{adh}(z)} G[\mathrm{cone}(y)])|$. Therefore, the order of (A'', B'') in $G[\mathrm{cone}(x)]$ is at most the order of (A, B) in G'_x, that is k. This separation (A'', B'') of $G[\mathrm{cone}(x)]$ contradicts the (q,k)-unbreakability of $G[\mathrm{bag}(x)]$. □

Disjoint-paths queries. Let G be a graph, k a positive integer and $s_1, t_1, \ldots, s_k, t_k$ vertices of G. We say that $\mathrm{dp}_k[(s_1,t_1), \ldots, (s_k,t_k)]$ holds in G if and only if in G there are internally vertex-disjoint paths between s_i and t_i for $1 \leqslant i \leqslant k$.

Signatures. We will only consider finite relational signatures Σ consisting of unary (arity 1) and binary (arity 2) relation symbols. An *alphabet* is a signature consisting only of unary relation symbols. A Σ-structure $\mathfrak{A}$ consists of a finite universe $V(\mathfrak{A})$ and an interpretation $R(\mathfrak{A}) \subseteq V(\mathfrak{A})^m$ of each m-ary relation symbol $R \in \Sigma$. When there is no ambiguity, we will not distinguish between relation symbols and their interpretations.

Graphs are represented as Σ-structures where the universe is the vertex set and Σ consists of one binary relation symbol $E(\cdot,\cdot)$, interpreted as the edge relation; that is, as an irreflexive and symmetric relation.

First-order logic (FO). For a fixed signature Σ, formulas of first-order logic are constructed from atomic formulas of the form $x = y$, where x and y are variables, and $R(x_1, \ldots, x_m)$, where $R \in \Sigma$ is an m-ary relation symbol and $x_1, \ldots, x_m$ are variables, by applying the Boolean operators $\neg$, $\wedge$ and $\vee$, and existential and universal quantification $\exists x$ and $\forall x$.

A variable x not in the scope of a quantifier is a *free variable* (we will not consider formulas with free set variables). A formula without free variables is a sentence. We write $\varphi(\bar{x})$ to indicate that the free variables of a formula φ are contained in the set of variables $\bar{x}$. A *valuation* of $\bar{x}$ in a set A is a function $\bar{a} : \bar{x} \to A$. Let $A^{\bar{x}}$ denote the set of all valuations of $\bar{x}$ in A.

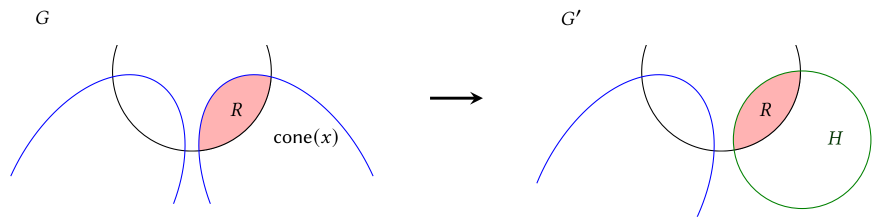

Figure 2: An illustration of the graphs in the statement of Lemma 2.3.

The satisfaction relation between Σ-structures and formulas is defined as usual by structural induction on the formula. When $\mathfrak{A}$ is a Σ-structure, $\varphi(\bar{x})$ is a formula with free variables contained in $\bar{x}$, and $\bar{a} \in V(\mathfrak{A})^{\bar{x}}$ is a valuation of $\bar{x}$, we write $(\mathfrak{A}, \bar{a}) \models \varphi(\bar{x})$ or $\mathfrak{A} \models \varphi(\bar{a})$ to denote that φ holds in $\mathfrak{A}$ when the variables are evaluated by $\bar{a}$. We let $\varphi(\mathfrak{A}) := \{\bar{a} \in A^{\bar{x}} \mid \mathfrak{A} \models \varphi(\bar{a})\}$.

Disjoint-paths logic ($\mathsf{FO+DP}$). Assume that Σ contains a distinguished binary relation symbol $E(\cdot,\cdot)$ that will always be interpreted as the edge relation of a graph. For a Σ-structure $\mathfrak{A}$, we write $G(\mathfrak{A})$ for the graph $(V(\mathfrak{A}), E(\mathfrak{A}))$. $\mathsf{FO+DP}$ is first-order logic extended by the disjoint-paths predicates $\mathsf{dp}_k[(x_1, y_1), \dots, (x_k, y_k)]$, for all $k \geqslant 1$, where $x_1, y_1, \dots, x_k, y_k$ are first-order variables. The satisfaction relation between Σ-structures and $\mathsf{FO+DP}$ formulas is as for first-order logic, where the new atomic formula $\mathsf{dp}_k[(x_1, y_1), \dots, (x_k, y_k)]$ holds in a Σ-structure $\mathfrak{A}$ with a valuation $\bar{a}$ of the variables $x_1, y_1, \dots, x_k, y_k$ to elements $s_1, t_1, \dots, s_k, t_k$ in $V(\mathfrak{A})$, if $G(\mathfrak{A}) \models \mathsf{dp}_k[(s_1, t_1), \dots, (s_k, t_k)]$.

3 COLLAPSE OF $\mathsf{FO+DP}$ ON UNBREAKABLE GRAPHS WITH LARGE CLIQUE MINORS

We now show that $\mathsf{FO+DP}$ collapses to plain first-order logic on unbreakable graphs that contain large clique minors. As mentioned in the introduction, we build on the generic folio technique, introduced in [41] and used in [19, 30]. In particular, our proof is based on the following result.

Proposition 3.1 (Lemma 4.2 from [30]). *Let G be a rooted graph. Let w be a positive weight function on $V(G)$. Let $t \geqslant \frac{3}{2} \cdot w(R(G))$ and $B_1, \dots, B_t$ be the branch sets of a minor model of K_t in G. Suppose that there is no separation (G_1, G_2) of G such that $w(V(G_1) \cap V(G_2)) < w(R(G))$, $R(G) \subseteq V(G_1)$, and $B_i \cap V(G_1) = \emptyset$, for some $i \in \{1, \dots, t\}$. Then, for every non-negative integer $k \leqslant \binom{|R(G)|}{2}$, the w-bounded k-folio of G is rooted-generic.*

We will apply this result as follows. Consider a predicate $\mathsf{dp}_k[(x_1, y_1), \dots, (x_k, y_k)]$. We will group the x_i and y_i that are close to each other into clusters $C_1, \dots, C_r$. We assign weight k to all vertices of $\{x_1, y_1, \dots, x_k, y_k\}$ and weight 1 to all other vertices. With this weight function chosen, we find small-weight separators S_i separating the C_i from the rest of the graph. Let $S = \bigcup_{i \in \{1,\dots,r\}} S_i$. By definition of the weight function, we have $|S| \leqslant p$ for an appropriately chosen value of p. Using the unbreakability of G, S separates the clusters from the rest of the graph, where the small part G_1 of the separation contains the clusters and the large part G_2 contains the minor model. We note that the small part contains the clusters since the branch sets of the minor model should be contained in the large part, due to the fact that they are disjoint sets that are more than the maximum size of the small part. By the choice of the weight function, we conclude with Proposition 3.1 that the w-weighted k-folio of G_2 rooted at S is rooted-generic, hence, that all pairs of vertices of S are connected by pairwise disjoint paths. Hence, any disjoint-paths query with elements of S is true. On the other hand, since the part containing the clusters is small (constant size), we can hardcode the disjoint-paths connectivity between the x_i, y_i and S by a first-order formula.

With this intuition at hand, let us more formally define this first-order formula, and then prove that it indeed has the desired properties. For every predicate $\mathsf{dp}_k[(x_1, y_1), \dots, (x_k, y_k)]$ we want to construct a formula $\varphi(x_1, y_1, \dots, x_k, y_k)$ such that, given a graph G and vertices $x_1, y_1, \dots, x_k, y_k \in V(G)$, $G \models \mathsf{dp}_k[(x_1, y_1), \dots, (x_k, y_k)]$ if and only if $G \models \varphi(x_1, y_1, \dots, x_k, y_k)$.

We set $p := 4k^3$ and $L := q(p)$, where q is the function from Theorem 2.2. The formula φ is defined as follows: We first partition $\{x_1, y_1 \dots, x_k, y_k\}$ into sets $C_1, \dots, C_r$ such that two vertices $v, u \in \{x_1, y_1 \dots, x_k, y_k\}$ are in the same C_i if there is a sequence of vertices $z_1, \dots, z_\ell$ from $\{x_1, y_1 \dots, x_k, y_k\}$ such that $v = z_1$ and $u = z_\ell$ and for every $i \in [\ell - 1]$, the distance between z_i and z_{i+1} is at most L. Since L is a fixed constant, we can express these distances with FO. With a big disjunction, we guess the number r of clusters and the number of vertices n_i contained in cluster C_i. By existentially quantifying vertices $c_1^1, \dots, c_1^{n_1}, \dots, c_r^1, \dots, c_r^{n_r}$ and requiring that each c_i^j is equal to exactly one of the vertices $\{x_1, y_1, \dots, x_k, y_k\}$, we explicitly name the elements of $\{x_1, y_1, \dots, x_k, y_k\}$ to specify to which cluster C_i they belong.

We now existentially quantify the existence of small separators $S_1, \dots, S_r$ and small sets $D_1, \dots, D_r$ with $C_i, S_i \subseteq D_i$ such that $(G[D_i], G - (D_i \setminus S_i))$ is a separation. Similar to the above, we use existential quantifiers to explicitly name the constantly many elements in these sets. The definition of the S_i includes the weight on the vertices of $\{x_1, y_1, \dots, x_k, y_k\}$, which will be explicitly hardcoded into the formulas (exactly the $2k$ elements of $\{x_1, y_1, \dots, x_k, y_k\}$ receive weight k, so that this is possible). After encoding these weights, we demand that S_i is of minimal weight.

Finally, we existentially quantify the existence of bounded (at most L) length paths in D_i that connect the vertices in C_i, say x_j, to the vertices of S_i or the pairs x_i, y_i in D_i.

Lemma 3.2. *Let $k \in \mathbb{N}$, let $p = 4k^3$, let $L = q(p)$, where q is the function of Theorem 2.2, and let $t = \max\{6k^3, 2kL + 1\}$. Also, let G be a graph that is (L, p)-unbreakable and contains K_t as a minor. Let $x_1, y_1, \dots, x_k, y_k \in V(G)$. We have $G \models \mathsf{dp}_k[(x_1, y_1), \dots, (x_k, y_k)]$ if and only if $G \models \varphi(x_1, y_1, \dots, x_k, y_k)$.*

Proof. Let $B_1, \ldots, B_t \subseteq V(G)$ be the branch sets of a minor model of K_t in G.

We first partition the set $\{x_1, y_1 \ldots, x_k, y_k\}$ into sets $C_1, \ldots, C_r$ such that two vertices v, u in $\{x_1, y_1 \ldots, x_k, y_k\}$ are in the same C_i if there is a sequence of vertices $z_1, \ldots, z_\ell$ from $\{x_1, y_1 \ldots, x_k, y_k\}$ such that $v = z_1$ and $u = z_\ell$ and for every $i \in [\ell - 1]$, the distance between z_i and z_{i+1} is at most L. We consider a weight function $w\colon V(G) \to \{1, k\}$ such that for every $v \in \{x_1, y_1 \ldots, x_k, y_k\}$, $w(v) = k$, and $w(v) = 1$ otherwise. For every $i \in \{1, \ldots, r\}$, we consider a separation (G_i^1, G_i^2) of G such that the following conditions are satisfied:

(1) $C_i \subseteq V(G_i^1)$,
(2) there is a branch set B_j, $j \in \{1, \ldots, t\}$ such that: $B_j \subseteq V(G_i^2) \setminus V(G_i^1)$, and
(3) $w(V(G_i^1) \cap V(G_i^2))$ is minimum.

Let $S_i = V(G_i^1) \cap V(G_i^2)$ and let $D_i = V(G_i^1)$. We know that $(G[C_i], G - E(G[C_i]))$ satisfies properties 1-3, hence $|S_i| \leqslant w(S_i) \leqslant w(C_i) = |C_i| \cdot k \leqslant 2k^2$ and therefore $|\bigcup_{i\in[r]} S_i| \leqslant 2k^2 \cdot 2k = p$. Since $B_j \subseteq V(G_i^2) \setminus V(G_i^1)$, we have that all branch sets $B_1, \ldots, B_t$ intersect $V(G_i^2)$ and therefore $|V(G_i^2)| > L$. For every $i \in \{1, \ldots, r\}$, by (L, p)-unbreakability, since $|S_i| \leqslant p$ we have that $|D_i| \leqslant L$. See Figure 3.

We set $C := \bigcup_{i\in\{1,\ldots,r\}} C_i$, $D := \bigcup_{i\in\{1,\ldots,r\}} D_i$, $G_1 := G[D]$, $S := \bigcup_{i\in\{1,\ldots,r\}} S_i$, and $G_2 := G - (D \setminus S)$. Note that (G_1, G_2) is a separation of G. Since $|S_i| \leqslant 2k^2$ for every $i \in \{1, \ldots, r\}$, we have that $|S| \leqslant 2k^2 \cdot 2k = p$.

Since for every $i \in \{1, \ldots, r\}$ we have that $|D_i| \leqslant L$ and $r \leqslant 2k$, it holds that $|D| \leqslant 2kL$. Also, since $B_1, \ldots, B_t$ are pairwise vertex-disjoint subsets of $V(G)$ and $t \geqslant 2kL + 1$, there is a $j \in \{1, \ldots, t\}$ such that $B_j \subseteq V(G) \setminus D$. Thus, since $|S| \leqslant p$ and $t \geqslant 6k^3 > p$ and $B_j \subseteq V(G) \setminus D$, we have that all branch sets $B_1, \ldots, B_t$ are intersecting $V(G_2)$. Therefore, $|V(G_2)| \geqslant t > L$ and thus, by (L, p)-unbreakability, we have that $|D| \leqslant L$.

Observe that the sets S_i are possibly not uniquely determined, however, such a set S_i satisfying the conditions exists, and any choice of such S_i (existentially quantified in the formula φ described before the lemma) will work. By existential quantification, some choice for S_i is being fixed, which implies a unique definition of D_i, and we can continue with the proof that the formula φ expresses exactly what we claim.

We will now show that if $G \models \mathsf{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$, then $G \models \varphi(x_1, y_1, \ldots, x_k, y_k)$. Let $P_1, \ldots, P_k$ be the internally vertex-disjoint paths of G certifying that the predicate $\mathsf{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$ is satisfied. Then, for every $i \in \{1, \ldots, k\}$, consider the paths $Q_i = P_i \cap D$. Since every S_j, $j \in \{1, \ldots, r\}$, is a separator between D_j and $G \setminus D_j$, and Q_i certifies the existence of paths of bounded length inside D, we have that $G \models \varphi(x_1, y_1, \ldots, x_k, y_k)$.

It remains to show that $G \models \varphi(x_1, y_1, \ldots, x_k, y_k)$ implies $G \models \mathsf{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$. The satisfaction of $\varphi(x_1, y_1, \ldots, x_k, y_k)$ implies that there exists a way to partition the set $\{x_1, y_1, \ldots, x_k, y_k\}$ into sets $C_1, \ldots, C_r$ and for every $i \in \{1, \ldots, r\}$, there is a minimal separator S_i such that the part that contains C_i (i.e., the part D_i) has size at most L and there are some pairwise disjoint paths connecting vertices x_i and y_i in D_i and some pairwise disjoint paths between C_i and $s_1^i, \ldots, s_{b_i}^i$, for some $b_i \leqslant p$. By (L, p)-unbreakability all branch sets $B_1, \ldots, B_t$ intersect $V(G_2)$ and so there is a branch set entirely contained in $V(G) \setminus D$.

We want to prove that G_2 rooted at S has a w-weighted k-folio that is rooted-generic. We will use Proposition 3.1 to show this. Suppose that there is a separation (F_1, F_2) of G_2 that violates the conditions of Proposition 3.1, i.e., it holds that $w(V(F_1) \cap V(F_2)) < w(S)$, $S \subseteq V(F_1)$, and there is a $j \in \{1, \ldots, t\}$ such that $B_j \cap V(F_1) = \emptyset$. Let $G_2' := F_2$ and let $G_1' := G - (V(F_2) \setminus (V(F_1) \cap V(F_2)))$. Observe that (G_1', G_2') is a separation of G such that $V(G_1') \cap V(G_2') = V(F_1) \cap V(F_2)$, $V(F_2) = V(G_2')$, and $V(F_1) \subseteq V(G_1')$. We have that if $S' = V(G_1') \cap V(G_2')$ then $|S'| \leqslant w(S') = w(V(F_1) \cap V(F_2)) < w(S) \leqslant p$. Also, it holds that $S \subseteq V(G_1')$, and for some $i \in \{1, \ldots, t\}$, $B_i \cap V(G_1') = \emptyset$. The latter implies that all branch sets $B_1, \ldots, B_t$ intersect $V(G_2')$ and therefore $|V(G_2')| \geqslant t > L$. Therefore, by (L, p)-unbreakability of G, we have that $|V(G_1')| \leqslant L$.

We set S_i' to be the set of vertices of S' that are connected with some path to some vertex of S_i in G_1'. We want to prove that $S_1', \ldots, S_r'$ are pairwise disjoint. Indeed, if there is a vertex $v \in S_i' \cap S_j'$, then there is a path P connecting vertices of C_i and C_j in G_1' that contains v. Since every two vertices $c_i \in C_i$ and $c_j \in C_j$ have a distance of at least L, then P should have length more than L. The fact that $V(P) \subseteq V(G_1')$ implies that $|V(G_1')| > L$, a contradiction. Therefore, $S_1', \ldots, S_r'$ are pairwise disjoint. This implies that if $w(S') < w(S)$, then there is some $i \in \{1, \ldots, r\}$ such that $w(S_i') < w(S_i)$. Now notice that S_i' is separating C_i from at least one branch set of the minor model of K_t and the fact that $w(S_i') < w(S_i)$ contradicts the minimality of S_i. Therefore, the k-folio of G_2 rooted at S is rooted-generic.

The fact that the k-folio of G_2 rooted at S is rooted-generic implies that for every collection of k pairs of (root) vertices in S, we can find pairwise disjoint paths connecting them. Thus, $G \models \mathsf{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$. □

4 DYNAMIC PROGRAMMING USING GAME TREES

A standard way to approach the model checking problem is via the model checking game, see e.g. [28]. This game is played by two players, Falsifier and Verifier, where the aim of Falsifier is to prove that a given formula is false on a given structure, while Verifier tries to prove the opposite. The moves of the players naturally correspond to the quantifiers of the given formula. After the players have chosen elements $v_1, \ldots, v_q$, the atomic type of this tuple determines the winner of the game. The game for FO naturally gives rise to a game tree, which however, for a structure of size n and a formula with q quantifiers is of size n^q, which is too large for the purpose of efficient model checking. We refer to [21, Chapter 4] for a nice presentation of the game-tree-based method. Our goal is to compute an equivalent pruned version of the game tree which is furthermore enriched with disjoint-paths information. The data structure for a node in the tree decomposition is this pruned game tree for the subgraph induced by all vertices in bags below the current node. Our presentation is inspired by [25].

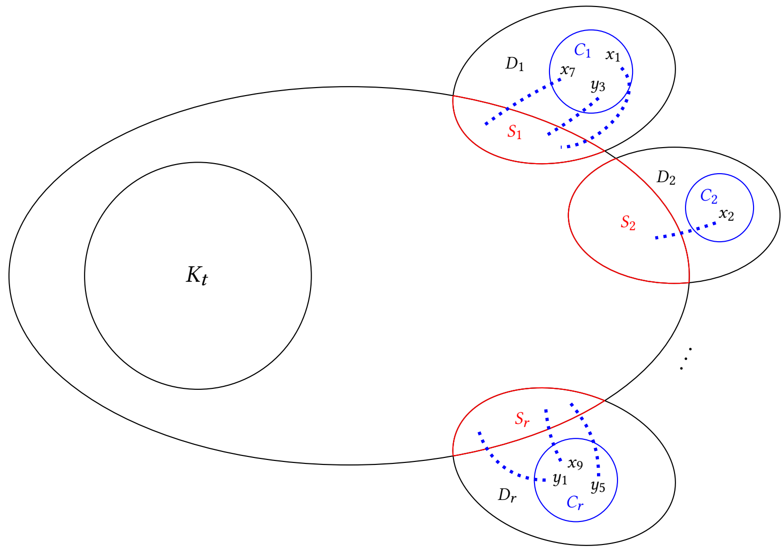

Figure 3: Disjoint-paths queries on unbreakable graphs

4.1 Annotated types

In order to prune the game tree of a graph, we have to restrict the scope of allowed moves of the players in the model checking game. To encode this, we use the notion of *annotated h-type* of an annotated graph $(G, R_1, \ldots, R_h)$, for given $h \in \mathbb{N}$, which can be seen as the "game-tree" encoding of all sentences of h variables in prenex normal form that G satisfies when the i-th quantified variable is asked to be interpreted in R_i. This corresponds to the annotated variant of *h-rank* FO/MSO-*types* given, for instance, by Shelah [44], Ebbinghaus and Flum [15, Definition 2.2.5], and Libkin [37, Section 3.4]; see also [5, 21, 25, 28].

For the sake of greater generality, we state the following definitions and results for *colored rooted* graphs, i.e., graphs equipped with some colors on their vertices (encoded as unary relations) and some roots (encoded as constant symbols).

Annotated types. We use $\Psi^{r,h,\ell}_{\mathsf{FO+DP}}$ to denote the set of all quantifier-free FO + DP-formulas with r free variables whose vocabulary is the graphs of h colors and ℓ roots. 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 $\Psi^{r,h,\ell}_{\mathsf{FO+DP}}$ is upper-bounded by some constant depending only on r, h, and ℓ. The *atomic type* of a tuple $\bar{v} \in (V(G) \cup \{\bot\})^r$ is the set of all atomic formulas that are true for $\bar{v}$ in G. Let G be a graph with h colors and ℓ roots, let $r \in \mathbb{N}$, and let $\bar{R} = (R_1, \ldots, R_r) \in (2^{V(G)})^r$. Given $\bar{v} \in (V(G) \cup \{\bot\})^r$, we define the 0-type$^r(G, \bar{R}, \bar{v})$ to be the atomic type of $\bar{v}$. Also, for each $i \in [r-1]$ and every $\bar{v} = (v_1, \ldots, v_{r-i}) \in (V(G) \cup \{\bot\})^{r-i}$, we define

$$i\text{-type}^r(G, \bar{R}, \bar{v}) = \big\{(i-1)\text{-type}^r(G, \bar{R}, \bar{v}u) \mid u \in R_{r-i+1} \cup \{\bot\}\big\}$$

Finally, we define the *r-annotated type* of (G, R) as

$$r\text{-type}^r(G, \bar{R}) = \big\{(r-1)\text{-type}^r(G, \bar{R}, v) \mid v \in R_1 \cup \{\bot\}\big\}.$$

It is easy to observe the following (see e.g. [21, Subsection 4.2] and [25, Lemma 4] for proofs of this observation using different terminology).

Observation 4.1. *Let $r \in \mathbb{N}$. Let G, G' be two colored rooted graphs and let $\bar{R} \in (2^{V(G)})^r$ and $\bar{R}' \in (2^{V(G')})^r$. If r-type$^r(G, \bar{R})$ = r-type$^r(G', \bar{R}')$, then for every sentence*

$$\varphi = Q_1 x_1 \ldots Q_r x_r\ \psi(x_1, \ldots, x_r) \wedge \Big(\bigwedge_{i \in [r]} x_i \in R_i\Big),$$

where $Q_i \in \{\forall, \exists\}$ and $\psi(x_1, \ldots, x_r)$ is a quantifier-free formula in FO + DP*, we have*

$$(G, \bar{R}) \models \varphi \iff (G', \bar{R}') \models \varphi.$$

Observation 4.1 implies that to do model checking, it suffices to compute a small size graph G' and sets $R'_1, \ldots, R'_r \subseteq V(G')$ such that $(G', R'_1, \ldots, R'_r)$ has the same annotated type as $(G, V(G), \ldots, V(G))$ (i.e., G annotated with r copies of $V(G)$).

Extended annotated types. Let G be a colored rooted graph, let $r \in \mathbb{N}$, and let $\bar{R} \subseteq (2^{V(G)})^r$. We use B to denote the set of roots of G. There are $2^{\binom{|B|}{2}}$ possible undirected graphs on B. As in Section 2, for each such graph H we write $G + H$ for the graph obtained from G by setting $G[B] = H$. Observe that every choice of H can give rise to a different annotated type for $(G + H, \bar{R})$. The *extended annotated type* of $(G, \bar{R})$ is the $2^{\binom{|B|}{2}}$-tuple of all these annotated types and is denoted by r-ext-type$^r(G, \bar{R})$.

As the set of atomic types is finite, we conclude the following.

Observation 4.2. *There is a function $f_1 : \mathbb{N}^3 \to \mathbb{N}$ such that, for all $r, \ell, h \in \mathbb{N}$, every h-colored graph G with ℓ roots and every $R_1, \ldots, R_r \subseteq V(G)$, there are minimal sets $R'_1, \ldots, R'_r \subseteq V(G)$, each of size at most $f_1(r, h, \ell)$ such that $R'_i \subseteq R_i$, for every $i \in [r]$, and r-ext-type$^r(G, R_1, \ldots, R_r) = r$-ext-type$^r(G, R'_1, \ldots, R'_r)$.*

In the next subsection, we describe how to compute these minimal representatives and how to perform dynamic programming on the decomposition given in Theorem 2.2.

4.2 Finding representatives with the same extended annotated type

In this subsection, we describe how to find small-size graphs of the same extended annotated type and same extended folio. We distinguish two cases. The first case is when the given graph contains a large clique as a minor. The additional assumption, in this case, is that the graph is unbreakable.

Large clique minors As a consequence of Lemma 3.2, we have that in the case where the graph contains a large clique minor and is unbreakable, atomic types for FO + DP are FO-definable. The definition of annotated types implies that if the atomic types for FO + DP are FO-definable, then annotated types (for FO + DP) are also FO-definable.

Corollary 4.3. *Let $k \in \mathbb{N}$, let $p = 4k^3$, let $L = q(p)$, where q is the function of Theorem 2.2, and let $t = \max\{6k^3, 2kL + 1\}$. Also, let G be a rooted (colored) graph that is (L, p)-unbreakable and contains K_t as a minor. Then, for every $r \in \mathbb{N}$ and every collection of vertices $v_1, \ldots, v_r \in V(G)$, the atomic type (for FO + DP) of $(G, v_1, \ldots, v_r)$ is FO-definable. Moreover, r-type$^r(G, \bar{R})$ is FO-definable.*

Since FO model checking is tractable in K_t-topological-minor-free classes, in a K_t-topological-minor-free (rooted) graph G, we can compute the annotated r-type of $(G, (V(G))^r)$ in FPT-time. We will also use the following result from [30].

Proposition 4.4 (Lemma 2.2 of [30]). *There is a computable function $f_2 \colon \mathbb{N}^2 \to \mathbb{N}$ such that for every $\delta, \ell \in \mathbb{N}$ and every extended δ-folio $\mathcal{F}$, the size of the minimum-size (in terms of vertices) rooted graph G on at most ℓ roots whose extended δ-folio is $\mathcal{F}$ is at most $f_2(\delta, \ell)$.*

We obtain the following result which allows us to find, given an unbreakable graph that contains large clique minors, a small-size graph of the same extended annotated type and same extended folio.

Lemma 4.5. *Let $k, r, \delta, h, \ell \in \mathbb{N}$, let $p = 4k^3$, let $L = q(p)$, where q is the function of Theorem 2.2, and let $t = \max\{6k^3, 2kL + 1\}$. There is a constant $c_{k,r,\delta,h,\ell}$ and an algorithm that given a colored rooted graph G that is (L, p)-unbreakable and contains K_t as a minor and sets $R_1, \ldots, R_r \subseteq V(G)$, outputs in quadratic time, a subgraph G' of G (rooted on the same vertices) and sets $R'_1, \ldots, R'_r \subseteq V(G')$ such that r-ext-type$^r(G, R_1, \ldots, R_r) = r$-ext-type$^r(G', R'_1, \ldots, R'_r)$, G' has the same extended δ-folio as G, and the size of G' is at most $c_{k,r,\delta,h,\ell}$, where ℓ is the number of roots of G and h is the number of colors of G.*

Proof. By Corollary 4.3, we have that r-type$^r(G, R_1, \ldots, R_r)$ is FO-definable. Therefore, by applying the model checking algorithm of Dvořák, Král, and Thomas [14] a linear number of times, we can compute, in quadratic time, a minimal set of vertices R'_1 such that r-ext-type$^r(G, R_1, \ldots, R_r) = r$-ext-type$^r(G, R'_1, R_2, \ldots, R_r)$. Indeed, by applying the model checking algorithm of Dvořák, Král, and Thomas [14], we can compute, in linear time, a minimal set of vertices $R'_1 \subseteq R_1$ such that for every $v \in R'_1$ and every undirected graph B on $R(G)$, $(r-1)$-type$^r(G + B, R_1, \ldots, R_r, v) \in r$-type$^r(G + B, R_1, \ldots, R_r)$, by computing $(r-1)$-type$^{r-1}(G + B, R_1, \ldots, R_r, w)$ for each vertex $w \in V(G)$. Note that the size of $|B|$ depends only on the constants k, r, h, ℓ and by Observation 4.2, it holds that $|R'_1| \leqslant f_1(r, h, \ell)$. Also, observe that r-ext-type$^r(G, R'_1, R_2, \ldots, R_r) = r$-ext-type$^r(G, R_1, \ldots, R_r)$. Then, one can recursively apply the algorithm of [14] to find for every $v \in R'_1$ and every undirected graph B on $R(G)$, a minimal set of vertices $X_v \subseteq R_2$ such that for $u \in X_v$, $(r-2)$-type$^r(G + B, R'_1, R_2, \ldots, R_r, vu) \in (r-1)$-type^r $(G + B, R'_1, R_2, \ldots, R_r, v)$. We set $R'_2 := \bigcup_{v \in R'_1} X_v$. Following this argument, in quadratic time, we compute the claimed minimal sets $R'_1, \ldots, R'_r \subseteq V(G)$, such that r-ext-type$^r(G, R'_1, \ldots, R'_r) = r$-ext-type^r $(G, R_1, \ldots, R_r)$.

Then, for every tuple $\bar{v} = (v_1, \ldots, v_r)$, where $v_i \in R_i \cup \{\perp\}$, we compute a minimum-size (in terms of vertices) subgraph $F_{\bar{v}}$ of G rooted at $\bar{v}$ such that 0-type$^r(G, \bar{v}) =$ 0-type$^r(F, \bar{v})$. This is done in quadratic time, using Proposition 4.4. Also, using Proposition 4.4, we compute a minimum-size subgraph F' of G (rooted on the same vertices as G) that has the same extended δ-folio. Thus the graph obtained by keeping only the vertices in $\bigcup_{i \in [r]} R'_i$, in $\bigcup_{\bar{v} \in R'_1 \times \cdots \times R'_r} V(F_{\bar{v}})$ and in $V(F')$ and the edges with both endpoints in $\bigcup_{i \in [r]} R'_i$, as well as the edges in $\bigcup_{\bar{v} \in R'_1 \times \cdots \times R'_r} E(F_{\bar{v}})$ and in $E(F')$, is the desired graph. □

Excluding large clique minors. We now deal with the case where the given graph does *not* contain large clique minors. We need the following result from [25].

Proposition 4.6. *There are two functions $f_3, f_4 \colon \mathbb{N}^3 \to \mathbb{N}$ and an algorithm that given $h, \ell, r \in \mathbb{N}$, an h-colored ℓ-rooted n-vertex graph G and sets $R_1, \ldots, R_r \subseteq V(G)$, outputs, in time $O_{h,\ell,r}(n^2)$, either a report that $K_{f_3(r,\ell,h)}$ is a minor of G, or a set $V \subseteq V(G)$ and sets $R'_1, \ldots, R'_r \subseteq V$, such that $\mathrm{tw}(G[V]) \leqslant f_4(r, \ell, h)$ and r-ext-type$^r(G, R_1, \ldots, R_r) = r$-ext-type$^r(G[V], R'_1, \ldots, R'_r)$.*

We will also use the following result for the evaluation of MSO-formulas [1] (see also [8]).

Proposition 4.7. *There is an algorithm that, given a graph G and an MSO formula $\varphi(\bar{X})$, either finds a collection $\bar{S}$ of $|\bar{X}|$-many subsets of $V(G)$ such that $G \models \varphi(\bar{S})$, or correctly reports that no such sets exist, in time $f(\varphi, \mathrm{tw}) \cdot |V(G)|$, where f is a computable function and tw is the treewidth of G.*

Using Propositions 4.4, 4.6 and 4.7, we obtain the following result.

Lemma 4.8. *Let $r, \delta, h, \ell \in \mathbb{N}$. Also, let $t' := f_3(r, \ell, h)$, where f_3 is the first function of Proposition 4.6. There is a constant $c'_{r,\delta,h,\ell}$ and an algorithm that given $r, \delta \in \mathbb{N}$, an h-colored ℓ-rooted graph G and sets $R_1, \ldots, R_r \subseteq V(G)$, outputs in quadratic time,*

- *either a report that $K_{t'}$ is a minor of G, or*
- *a subgraph G' of G (rooted on the same vertices) and sets $R'_1, \ldots, R'_r \subseteq V(G')$ such that the following hold: G' has the same extended δ-folio as G, r-ext-type$^r(G, R_1, \ldots, R_r) = r$-ext-type$^r(G', R'_1, \ldots, R'_r)$, and $|G'| \leqslant c'_{r,\delta,h,\ell}$.*

Proof of Lemma 4.8. We first apply Proposition 4.6. By this, we obtain a set $V \subseteq V(G)$ and sets $R'_1, \ldots, R'_r \subseteq V$, such that r-ext-type$^r(G, R_1, \ldots, R_r) = r$-ext-type$^r(G[V], R'_1, \ldots, R'_r)$ and also

$\mathrm{tw}(G[V]) \leqslant f_4(r, \ell, h)$. Then, we enhance V by adding all vertices that are needed so that G and $G[V]$ have the same extended δ-folio. This is done using Proposition 4.4. We know that the resulting graph $G[V]$ has bounded treewidth. Using Proposition 4.7, we can compute a subgraph G' of $G[V]$ and sets $R_1'', \ldots, R_r'' \subseteq V(G')$ such that $r\text{-ext-type}^r(G[V], R_1', \ldots, R_r') = r\text{-ext-type}^r(G', R_1'', \ldots, R_r'')$, G' has the same extended δ-folio as $G[V]$ (and therefore as G), and the size of G' is at most $c'_{r,\delta,h,\ell}$. □

Combining game trees. We now prove the following result that intuitively says for every separation (X, Y) of a graph G, one can safely replace $G[X]$ by a subgraph of it (that also contains $X \cap Y$) of the same extended annotated type without affecting the extended annotated type of the whole graph. This is one of the key arguments for the correctness of the dynamic programming algorithm described in Section 4.3.

Lemma 4.9. *Let G be a graph rooted at some set B, let $r \in \mathbb{N}$, and let $R_1, \ldots, R_r \subseteq V(G)$. For every separation (X, Y) of G where $B \subseteq Y$, every subgraph G' of $G[X]$ rooted at $X \cap Y$, and every $R_1', \ldots, R_r' \subseteq V(G')$ such that*

$$r\text{-ext-type}^r(G[X], R_1, \ldots, R_r) = r\text{-ext-type}^r(G', R_1', \ldots, R_r'),$$

it holds that

$$r\text{-ext-type}^r(G, R_1, \ldots, R_r) = r\text{-ext-type}^r(\tilde{G}, \tilde{R}_1, \ldots, \tilde{R}_r),$$

where $\tilde{G} = G[Y \cup V(G')]$ and $\tilde{R}_i = R_i' \cup (R_i \cap Y)$, $i \in [r]$.

Proof. We say that two tuples $\bar{v}, \bar{v}' \in (X \cup \{\bot\})^r$ are *compatible* if for every $i \in [r]$, $v_i = \bot \iff v_i' = \bot$ and if $v_i \in X \cap Y$ then $v_i = v_i'$. Let $\bar{v} \in (X \cup \{\bot\})^r$. An *extension of $\bar{v}$ with a tuple of elements of* $(Y \setminus X) \cup \{\bot\}$ is a tuple of $V(G) \cup \{\bot\}$ obtained after replacing the $\bot$-elements of $\bar{v}$ with a tuple of elements from $(Y \setminus X) \cup \{\bot\}$. We use $\text{ext-atp}(G, \bar{v})$ to denote the tuple of all different atomic types, for each $(G^H, \bar{v})$, where $G^H = (V(G), E(G) \cup E(H))$ for some undirected graph H on the roots of G.

Claim 1. *Let $\bar{v}, \bar{v}' \in (X \cup \{\bot\})^r$ be compatible tuples such that $\text{ext-atp}(G[X], \bar{v}) = \text{ext-atp}(G[X], \bar{v}')$. Then for every extension $\bar{u}, \bar{u}'$ of $\bar{v}, \bar{v}'$ with the same tuple of elements from $(Y \setminus X) \cup \{\bot\}$, it holds that $\text{ext-atp}(G, \bar{u}) = \text{ext-atp}(G, \bar{u}')$.*

Proof of Claim 1. Let $\bar{v}, \bar{v}' \in (X \cup \{\bot\})^r$ be two compatible tuples such that $\text{ext-atp}(G[X], \bar{v}) = \text{ext-atp}(G[X], \bar{v}')$. We fix some extension $\bar{u}, \bar{u}'$ of $\bar{v}, \bar{v}'$ with a collection of vertices from $(Y \setminus X) \cup \{\bot\}$. We also fix some undirected graph H on the roots of G and we set $G^H = (V(G), E(G) \cup E(H))$. We know that there is some $\psi \in \mathsf{FO+DP}$ such that $\psi(\bar{u}) = 0\text{-type}^r(G^H, \bar{u})$. By definition, we have that $G^H \models \psi(\bar{u})$.

For every $\mathrm{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$ that appears in ψ, we will prove that if we set, for every $i \in [k]$, $x_i' = u_i'$ (resp. $y_i' = u_i'$), if $x_i = u_i$ (resp. $y_i' = u_i$) and $x_i' = x_i$ (resp. $y_i' = y_i$), if $x_i \in X \cap Y$ (resp. $y_i \in X \cap Y$), then $G \models \mathrm{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$ if and only if $G \models \mathrm{dp}_k[(x_1', y_1'), \ldots, (x_k', y_k')]$.

Suppose that $G \models \mathrm{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$. This implies the existence of pairwise internally vertex-disjoint paths $P_1, \ldots, P_k$ in G, where for every $i \in [k]$, P_i is an (x_i, y_i)-path. The following arguments are inspired by the proof of [30, Lemma 2.4]. We define a graph $F^\star$ on $(X \cap Y) \cap \bigcup_{i \in [k]} V(P_i)$ such that two vertices a, b of $F^\star$ are adjacent if there is some $i \in [k]$ and a subpath of P_i with endpoints a and b and every internal vertex in $Y \setminus X$. For every $ab \in E(F^\star)$, we denote by P_{ab} this subpath. For every path P in G that has endpoints in X, we use $P^{\langle X \rangle}$ to denote the path of $G[X]^{F^\star}$ obtained by replacing subpaths of P whose internal vertices are in $Y \setminus X$ by the appropriate edges of $F^\star$. Similarly, if Q is a path of $G[X]^{F^\star}$, we denote by $Q^{(Y)}$ the path of G obtained by replacing each edge ab of $F^\star$ by the corresponding path P_{ab}. For every $i \in [k]$, we define a path $\hat{P}_i$ as follows:

- if $x_i, y_i \in X$, then $\hat{P}_i = P_i^{\langle X \rangle}$,
- if $x_i \in X$ and $y_i \notin X$ then $\hat{P}_i$ is the (x_i, z_i)-subpath of P_i, where z_i is the last vertex of P_i (traversing from x_i to y_i) that belongs to X (we define $\hat{P}_i$ analogously when $x_i \notin X$ and $y_i \in X$),
- if $x_i, y_i \notin X$, and P_i does not intersect X, then $\hat{P}_i$ is defined as the empty graph,
- if $x_i, y_i \notin X$ and P_i intersects X at a single vertex z, then $\hat{P}_i$ is the trivial (z, z)-path,
- if $x_i, y_i \notin X$ and P_i intersects X at least twice, then if z and w are the first and the last vertex of P_i in X, then $\hat{P}_i$ is the (z, w)-subpath of $P_i^{\langle X \rangle}$.

Observe that the paths $\hat{P}_1, \ldots, \hat{P}_k$ are pairwise internally vertex-disjoint paths in $G[X]^{F^\star}$. Then, the fact that $\text{ext-atp}(G[X], \bar{v}) = \text{ext-atp}(G[X], \bar{v}')$ implies the existence of a collection of paths $\hat{Q}_1, \ldots, \hat{Q}_k$, where if $\hat{P}_i$ has endpoints z, w then $\hat{Q}_i$ has endpoints z', w', where $z' = v_j'$ if $z = v_j$ (resp. $w' = v_j'$ if $w = v_j$) for some $j \in [r]$, and $z' = z$ (resp. $w' = w$) otherwise, and $\hat{Q}_1, \ldots, \hat{Q}_k$ are pairwise internally vertex-disjoint paths in $G[X]^{F^\star}$. Note that, if we set $Q_i = \hat{Q}_i^{(Y)}$ for each $i \in [k]$, then each Q_i is an (x_i', y_i')-path and $Q_1, \ldots, Q_k$ are pairwise internally vertex-disjoint paths of G. Therefore $G \models \mathrm{dp}_k[(x_1', y_1'), \ldots, (x_k', y_k')]$.

The reverse implication, i.e., that $G \models \mathrm{dp}_k[(x_1', y_1'), \ldots, (x_k', y_k')]$ implies $G \models \mathrm{dp}_k[(x_1, y_1), \ldots, (x_k, y_k)]$, is proven by symmetric arguments. Thus, $\text{ext-atp}(G, \bar{u}) = \text{ext-atp}(G, \bar{u})$. This proves the claim.

The above claim implies that for every subgraph G' of $G[X]$ rooted at $X \cap Y$ and every $R_1', \ldots, R_r' \subseteq V(G')$ for which it holds that $r\text{-ext-type}^r(G[X], R_1, \ldots, R_r) = r\text{-ext-type}^r(G', R_1', \ldots, R_r')$, we have

$$r\text{-ext-type}^r(G, R_1, \ldots, R_r) = r\text{-ext-type}^r(\tilde{G}, \tilde{R}_1, \ldots, \tilde{R}_r).$$

□

4.3 Proof of Theorem 1.1

Now we can finally prove Theorem 1.1 which we repeat for convenience.

Theorem 1.1. *Let $\mathscr{C}_H$ be the class of graphs excluding a fixed graph H as a topological minor. Then, there is an algorithm that, given $G \in \mathscr{C}_H$ and an $\mathsf{FO+DP}$ formula $\varphi(\bar{x})$ and $\bar{v} \in V(G)^{|\bar{x}|}$, decides whether $G \models \varphi(\bar{v})$ in time $f(\varphi) \cdot |V(G)|^3$, where f is a computable function depending on H.*

In fact, we will show how a more general statement, namely model checking on h-colored graphs, for some $h \in \mathbb{N}$. Let G be an H-topological-minor-free graph with h colors, let $\varphi(\bar{x})$ be an

FO+DP formula, and let $\bar{v} \in V(G)^{|\bar{x}|}$. We set $\ell := |\bar{x}|$ and $t := |V(H)|$. Note that if G excludes H as a topological minor, then it also excludes K_t. Let r be the quantifier rank of φ, let k be the maximum k' such that the predicate $\mathsf{dp}_{k'}([(x_1, y_1), \ldots, (x_{k'}, y_{k'})])$ appears as an atomic formula in φ, and observe that there is a function $g : \mathbb{N}^3 \to \mathbb{N}$ such that $k \leqslant g(r, h, \ell)$. Finally, we set $p := 4k^3$ (as in Lemma 3.2), $\delta := q(p)$, where q is the function of Theorem 2.2, and $c := \max\{c_{k,r,\delta,h,\ell}, c'_{r,\delta,h,\ell}\}$. These last two constants are taken from Lemma 4.5 and Lemma 4.8, respectively. We also set $R_1 = \ldots = R_r = V(G)$.

We call the algorithm of Theorem 2.2 and we compute a strongly $(q(p), p)$-unbreakable tree decomposition $\mathcal{T} = (T, \mathsf{bag})$ of G of adhesion at most $q(p)$. Then, in a bottom-up way, for every node $x \in V(T)$, we compute a graph G'_x rooted at $\mathsf{adh}(x)$ and sets $R'_{1,x}, \ldots, R'_{r,x} \subseteq V(G'_x)$ such that $r\text{-ext-type}^r(G'_x, R'_{1,x}, \ldots, R'_{r,x}) = r\text{-ext-type}^r(G[\mathsf{cone}(x)], R_1 \cap \mathsf{cone}(x), \ldots, R_r \cap \mathsf{cone}(x))$, G'_x and $G[\mathsf{cone}(x)]$ have the same extended $q(p)$-folio, and G'_x has at most c vertices.

This is done as follows: Fix a node $x \in V(T)$ and suppose that for every $z \in \mathsf{children}(x)$, we have already computed a graph G'_z rooted at $\mathsf{adh}(z)$ and sets $R'_{1,z}, \ldots, R'_{r,z} \subseteq V(G'_z)$ where

- $r\text{-ext-type}^r(G'_z, R'_{1,z}, \ldots, R'_{r,z})$ is equal to $r\text{-ext-type}^r(G[\mathsf{cone}(z)], R_1 \cap \mathsf{cone}(z), \ldots, R_r \cap \mathsf{cone}(z))$,
- G'_z and $G[\mathsf{cone}(z)]$ have the same extended $q(p)$-folio, and
- G'_z has at most c vertices.

Let A_x be the set of all $z \in \mathsf{children}(x)$ so that there is no $z' \neq z$ in A_x with $\mathsf{adh}(z) \subseteq \mathsf{adh}(z')$. Note that $\bigcup_{z \in \mathsf{children}(x)} \mathsf{adh}(z) \subseteq \bigcup_{z \in A_x} \mathsf{adh}(z)$. For each $z \in A_x$, let $G^{\langle z \rangle}$ be the (rooted) graph obtained from the union of all (rooted) graphs G'_w, where $w \in \mathsf{children}(x)$ and $\mathsf{adh}(w) \subseteq \mathsf{adh}(z)$, after identifying same-indexed roots in all such G'_w. Observe that $G^{\langle z \rangle}$ has treewidth at most $c + q(p)$. Therefore, using Courcelle's theorem, we are able to find a graph $\tilde{G}_z$, also rooted at $\mathsf{adh}(z)$, and sets $\tilde{R}_{1,z}, \ldots, \tilde{R}_{r,z} \subseteq V(\tilde{G}_z)$ with the following properties: $r\text{-ext-type}^r(\tilde{G}_z, \tilde{R}_{1,z}, \ldots, \tilde{R}_{r,z})$ is equal to $r\text{-ext-type}^r(G^{\langle z \rangle}, R_1 \cap V(G^{\langle z \rangle}), \ldots, R_r \cap V(G^{\langle z \rangle}))$, the (rooted) graphs $\tilde{G}_z$ and $G^{\langle z \rangle}$ have the same extended $q(p)$-folio, and $\tilde{G}_z$ has at most c vertices. Note that the graph $\hat{G}$ obtained from $G[\mathsf{bag}(x)]$ after gluing each $\tilde{G}_z$ to $\mathsf{adh}(z)$ is $K_{t'}$-topological-minor-free by Lemma 2.3, where $t' = \max\{t, 2p + 2, c\}$. Also, by Lemma 2.4, $\hat{G}$ is $(f(c, p), p)$-unbreakable. We now aim to find a graph G'_x rooted at $\mathsf{adh}(x)$ and sets $R'_{1,x}, \ldots, R'_{r,x} \subseteq V(G'_x)$ with the following properties: $r\text{-ext-type}^r(G'_x, R'_{1,x}, \ldots, R'_{r,x})$ is equal to $r\text{-ext-sig}^r(G[\mathsf{cone}(x)], R_1 \cap \mathsf{cone}(x), \ldots, R_r \cap \mathsf{cone}(x))$, G'_x and $G[\mathsf{cone}(x)]$ have the same extended $q(p)$-folio, and G'_x has at most c vertices. For this, we first apply the algorithm of Lemma 4.8 that either outputs a report that $\hat{G}$ contains $K_{f_3(r,h,\ell)}$ as a minor, or an annotated graph $(G'_x, R'_1, \ldots, R'_r)$ with the claimed properties. In the former case, i.e., if $\hat{G}$ contains $K_{f_3(r,h,\ell)}$ as a minor, we set $(G'_x, R'_1, \ldots, R'_r)$ to be the annotated graph given by Lemma 4.5. By Lemma 4.9, we have that $r\text{-ext-type}^r(\hat{G}, R_1, \ldots, R_r)$ is equal to $r\text{-ext-type}^r(G'_x, R'_1, \ldots, R'_r)$.

5 CONCLUSION

In this work, we have fully classified the subgraph-closed classes admitting efficient encoding of topological minors on which model checking for FO+DP is fixed-parameter tractable. A natural next question is to study the model checking problem also for dense graph classes that are not necessarily closed under taking subgraphs.

Another interesting question is the following. In [40] the authors considered a framework where after a polynomial time preprocessing queries of separator logic can be answered in constant time. Is the same true for disjoint-paths logic? The most basic question is whether we can answer disjoint-paths queries in constant time (or even linear time) after preprocessing. Even though we failed to implement the framework of [40] for model checking, it may be the case that we can extend this framework for query answering after preprocessing, since we can incorporate the data structure computed by dynamic programming. Nevertheless, at this point, there are more difficulties because we do not know how to answer disjoint-paths queries in constant time in minor-closed classes. It seems plausible that we can lift results for graphs of bounded genus to almost embeddable graphs and use the structure theorem to improve the running time for classes with excluded minors to linear. Then using our methods we would be able to improve it for all graphs to linear after preprocessing. Based on the nature of the irrelevant vertex technique it seems unlikely that we can improve the query time to constant.

ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers of previous versions of this paper. Their remarks and suggestions were valuable.

This work is supported by the French-German Collaboration ANR/DFG Project UTMA (ANR-20-CE92-0027, DFG 446200270). The work of the fourth author was also supported by the MEAE and the MESR, via the Franco-Norwegian project PHC Aurora projet n. 51260WL (2024)

The work of Giannos Stamoulis was also supported by the project BOBR that has received funding from the European Research Council (ERC), grant agreement No 948057.

REFERENCES

[1] Stefan Arnborg, Jens Lagergren, and Detlef Seese. 1991. Easy problems for tree-decomposable graphs. *Journal of Algorithms* 12, 2 (1991), 308–340.

[2] Christoph Berkholz, Jens Keppeler, and Nicole Schweikardt. 2018. Answering FO+MOD queries under updates on bounded degree databases. *ACM Transactions on Database Systems (TODS)* 43, 2 (2018), 1–32.

[3] Mikołaj Bojańczyk. 2021. Separator logic and star-free expressions for graphs. *arXiv preprint arXiv:2107.13953* (2021).

[4] Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Toruńczyk. 2022. Twin-width IV: ordered graphs and matrices. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. 924–937.

[5] Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. 2021. Twin-width I: tractable FO model checking. *Journal of the ACM (JACM)* 69, 1 (2021), 1–46.

[6] Bruno Courcelle. 1990. The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs. *Information and Computation* 85, 1 (1990), 12–75.

[7] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. 2000. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems* 33, 2 (2000), 125–150.

[8] Bruno Courcelle and Mohamed Mosbah. 1993. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science* 109, 1 (1993), 49–82.

[9] Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. 2019. Minimum bisection is fixed-parameter tractable. *SIAM J. Comput.* 48, 2 (2019), 417–450.

[10] Anuj Dawar, Martin Grohe, and Stephan Kreutzer. 2007. Locally excluding a minor. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS)*. IEEE, 270–279.

[11] Jan Dreier, Ioannis Eleftheriadis, Nikolas Mählmann, Rose McCarty, Michał Pilipczuk, and Szymon Toruńczyk. 2023. First-Order Model Checking on Monadically Stable Graph Classes. *arXiv preprint arXiv:2311.18740* (2023).

[12] Jan Dreier, Nikolas Mählmann, and Sebastian Siebertz. 2023. First-Order Model Checking on Structurally Sparse Graph Classes. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, Barna Saha and Rocco A. Servedio (Eds.). ACM, 567–580.

[13] Jan Dreier and Peter Rossmanith. 2021. Approximate evaluation of first-order counting queries. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1720–1739.

[14] Zdeněk Dvořák, Daniel Král, and Robin Thomas. 2010. Deciding first-order properties for sparse graphs. In *Proceedings of the IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 133–142.

[15] Heinz-Dieter Ebbinghaus and Jörg Flum. 1999. *Finite model theory*. Springer Science & Business Media.

[16] Kord Eickmeyer and Ken-ichi Kawarabayashi. 2017. FO model checking on map graphs. In *International Symposium on Fundamentals of Computation Theory*. Springer, 204–216.

[17] Jörg Flum and Martin Grohe. 2001. Fixed-parameter tractability, definability, and model-checking. *SIAM J. Comput.* 31, 1 (2001), 113–145.

[18] Fedor V. Fomin, Petr A. Golovach, Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. 2023. Compound Logics for Modification Problems. In *Proceedings of the 50th International Colloquium on Automata, Languages, and Programming (ICALP) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 261)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 61:1–61:21.

[19] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. 2020. Hitting topological minors is FPT. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. ACM, 1317–1326.

[20] Markus Frick and Martin Grohe. 2001. Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM (JACM)* 48, 6 (2001), 1184–1206.

[21] Jakub Gajarský, Maximilian Gorsky, and Stephan Kreutzer. 2022. Differential Games, Locality, and Model Checking for FO Logic of Graphs. In *Proceedings of the 30th EACSL Annual Conference on Computer Science Logic (CSL) (LIPIcs, Vol. 216)*. 22:1–22:18.

[22] Jakub Gajarský, Stephan Kreutzer, Jaroslav Nešetřil, Patrice Ossona De Mendez, Michał Pilipczuk, Sebastian Siebertz, and Szymon Toruńczyk. 2020. First-order interpretations of bounded expansion classes. *ACM Transactions on Computational Logic (TOCL)* 21, 4 (2020), 1–41.

[23] Robert Ganian, Petr Hliněný, Daniel Král, Jan Obdržálek, Jarett Schwartz, and Jakub Teska. 2013. FO model checking of interval graphs. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*. Springer, 250–262.

[24] Robert Ganian, Petr Hliněný, Alexander Langer, Jan Obdržálek, Peter Rossmanith, and Somnath Sikdar. 2014. Lower bounds on the complexity of MSO_1 model-checking. *J. Comput. System Sci.* 80, 1 (2014), 180–194.

[25] Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. 2022. Model-Checking for First-Order Logic with Disjoint Paths Predicates in Proper Minor-Closed Graph Classes. *arXiv preprint arXiv:2211.01723* (2022).

[26] Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. 2023. Model-Checking for First-Order Logic with Disjoint Paths Predicates in Proper Minor-Closed Graph Classes. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 3684–3699.

[27] Erich Grädel, Phokion G. Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Y Vardi, Yde Venema, and Scott Weinstein. 2007. *Finite Model Theory and its applications*. Springer.

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

[29] Martin Grohe. 2008. Logic, graphs, and algorithms. *Logic and automata* 2 (2008), 357–422.

[30] Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. 2011. Finding topological subgraphs is fixed-parameter tractable. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC)*. ACM, 479–488.

[31] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. 2017. Deciding First-Order Properties of Nowhere Dense Graphs. *Journal of the ACM* 64, 3 (2017), 17:1–17:32.

[32] Martin Grohe and Nicole Schweikardt. 2018. First-order query evaluation with cardinality conditions. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*. 253–266.

[33] Petr Hlinený, Filip Pokrývka, and Bodhayan Roy. 2017. FO Model Checking of Geometric Graphs. In *Proceedings of the 12th International Symposium on Parameterized and Exact Computation (IPEC) (LIPIcs, Vol. 89)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 19:1–19:12.

[34] Stephan Kreutzer and Siamak Tazari. 2010. Lower bounds for the complexity of monadic second-order logic. In *Proceedings of the 2010 25th Annual IEEE Symposium on Logic in Computer Science (LICS)*. IEEE, 189–198.

[35] Dietrich Kuske and Nicole Schweikardt. 2017. First-order logic with counting. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. IEEE, 1–12.

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

[37] Leonid Libkin. 2004. *Elements of finite model theory*. Vol. 41. Springer.

[38] Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. 2018. Reducing CMSO Model Checking to Highly Connected Graphs. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP) (LIPIcs, Vol. 107)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 135:1–135:14.

[39] Jaroslav Nešetřil, Patrice Ossona de Mendez, and Sebastian Siebertz. 2023. Modulo-counting first-order logic on bounded expansion classes. *Discrete Mathematics* (2023), 113700.

[40] Michał Pilipczuk, Nicole Schirrmacher, Sebastian Siebertz, Szymon Toruńczyk, and Alexandre Vigny. 2022. Algorithms and Data Structures for First-Order Logic with Connectivity Under Vertex Failures. In *Proceedings of the 49th International Colloquium on Automata, Languages, and Programming, (ICALP) (LIPIcs, Vol. 229)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 102:1–102:18.

[41] Neil Robertson and Paul D. Seymour. 1995. Graph minors. XIII. The disjoint paths problem. *Journal of combinatorial theory, Series B* 63, 1 (1995), 65–110.

[42] Nicole Schirrmacher, Sebastian Siebertz, and Alexandre Vigny. 2023. First-order Logic with Connectivity Operators. *ACM Trans. Comput. Log.* 24, 4, 30:1–30:23.

[43] Detlef Seese. 1996. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science* 6, 6 (1996), 505–526.

[44] Saharon Shelah. 1975. The Monadic Theory of Order. *Annals of Mathematics* 102, 3 (1975), 379–419.

[45] Szymon Toruńczyk. 2020. Aggregate queries on sparse databases. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 427–443.