Locally Excluding a Minor

Anuj Dawar
University of Cambridge Computer Laboratory
anuj.dawar@cl.cam.ac.uk

Martin Grohe
Institut für Informatik,
Humboldt Universität zu Berlin
grohe@informatik.hu-berlin.de

Stephan Kreutzer
Institut für Informatik,
Humboldt Universität zu Berlin
kreutzer@informatik.hu-berlin.de

Abstract

We introduce the concept of locally excluded minors. Graph classes locally excluding a minor are a common generalisation of the concept of excluded minor classes and of graph classes with bounded local tree-width.

We show that first-order model-checking is fixed-parameter tractable on any class of graphs locally excluding a minor. This strictly generalises analogous results by Flum and Grohe on excluded minor classes and Frick and Grohe on classes with bounded local tree-width.

As an important consequence of the proof we obtain fixed-parameter algorithms for problems such as dominating or independent set on graph classes excluding a minor, where now the parameter is the size of the dominating set and the excluded minor.

We also study graph classes with excluded minors, where the minor may grow slowly with the size of the graphs and show that again, first-order model-checking is fixed-parameter tractable on any such class of graphs.

1 Introduction

An important task in the theory of algorithms is to find feasible instances of otherwise intractable algorithmic problems. For this purpose, the notion of bounded tree-width has proved to be extremely useful. Many NP-complete problems become tractable on graphs whose tree-width is bounded by a fixed constant. These include k-colourability, Hamiltonicity and the k-dominating and k-vertex cover problems. Courcelle [2] proved a meta-theorem stating that any decision problem definable in monadic second-order logic can be decided in linear time on any class of graphs of bounded tree-width.

Although among the most prominent, tree-width is not the only structural property of graphs that allows for efficient solutions of otherwise intractable problems. Other important restrictions are planarity or bounded degree. Unfortunately, Courcelle's theorem fails for any of these restrictions. Seese [16] was the first to give a logical meta-theorem for another general class of graphs. He showed that any first-order definable decision problem can be decided in linear time on any class of graphs of bounded degree. A property of planar graphs and graphs of bounded degree that is often used to obtain tractable algorithms is the fact that every neighbourhood of a vertex has low tree-width, i.e. the tree-width of any neighbourhood of a vertex only depends on its diameter. This observation led Eppstein [6] to introduce the notion of bounded local tree-width (which he calls diameter tree-width property). The concept of local tree-width strictly generalises tree-width, planarity and bounded degree.

In [10], Frick and Grohe generalise Seese's result to classes of graphs of bounded local tree-width. They proved that first-order definable decision problems can be decided in linear time on what they call locally tree-decomposable classes of graphs. In the same paper, they show that any first-order definable decision problem can be decided in quadratic time on any class of graphs of bounded local tree-width.

Another important concept of graph structure that has been used to obtain tractable algorithms is the concept of excluded minors. In a series now running to 23 papers, Robertson and Seymour developed their groundbreaking theory on graph minors that culminated in the proof of Wagner's conjecture stating that in every infinite class of finite graphs one graph is a minor of another. In other words, every minor closed class of graphs that is not the class of all graphs can be characterised by a finite set of excluded minors. They also proved that testing whether a fixed graph is a minor of a graph G can be done in cubic time. It follows, that any minor closed class of graphs can be decided in cubic time.

Many parts of the rich and deep theory developed by Robertson and Seymour have found algorithmic applications. In [7], Flum and Grohe proved a meta-theorem similar to the results mentioned above. They showed that any first-order definable decision problem can be decided in polynomial time on any class of graphs excluding a fixed minor. The concept of excluded minors is incomparable to the concept of local tree-width, in fact even to bounded degree. It is therefore a natural question whether there exists a common generalisation of excluded minor classes and bounded local-tree width classes on which we can still efficiently decide first-order definable decision problems. The main contribution of this paper is to introduce such a generalisation. It is based on the following simple concept. Let $\mathcal{C}$ be a class of graphs. Instead of requiring that any graph in $\mathcal{C}$ excludes a fixed minor, we only require that every neighbourhood excludes a minor, depending on its radius. Formally, we require that for every radius r there is a graph H_r so that every r-neighbourhood of a vertex of any member of $\mathcal{C}$ excludes H_r. We call classes of graphs with this property *graph classes locally excluding a minor*. It is easily seen that if $\mathcal{C}$ is a class of graphs of bounded local tree-width or $\mathcal{C}$ is a class of graphs excluding a fixed minor, then $\mathcal{C}$ also locally excludes a minor. The following is our main result on locally excluded minor classes.

Theorem 1.1 *Let $\mathcal{C}$ be a class of graphs locally excluding a minor. Then deciding first-order properties is fixed-parameter tractable on $\mathcal{C}$.*

Here, the exponent of the polynomial is fixed and neither depends on the formula used to define the problem nor on the locally excluded minors.

It is instructive to compare graph classes locally excluding a minor with other generalisation of graphs of bounded tree-width. As stated above, they generalise classes of bounded local tree-width and classes with excluded minors. However, they are incomparable with classes of bounded clique-widthh [3], for which first-order model checking is known to be fixed-parameter tractable and also classes of graphs of bounded expansion [17], for which the question of fixed-parameter tractability of first-order logic remains open.

We also consider classes $\mathcal{C}$ of graphs such that there is a (slowly growing) function $f : \mathbb{N} \to \mathbb{N}$ with the property that any $G \in \mathcal{C}$ exclude a minor of cardinality at most $f(|G|)$ and show that also on such classes of graphs, first-order model checking is still fixed-parameter tractable.

Theorem 1.2 *There is an unbounded function $f : \mathbb{N} \to \mathbb{N}$ such that deciding first-order properties is fixed-parameter tractable on the class $\mathcal{C}_f$ of graphs G excluding a clique of order at most $f(|G|)$.*

The method we use to show the theorem has further important consequences. It is well known that various intractable problems such as k-dominating set and others are fixed-parameter tractable on classes of graphs excluding a fixed minor, where k is the parameter. However, it was an open problem whether the exponent of the polynomials can be made independent of the excluded minor, i.e. whether these problems can be solved by a fixed-parameter algorithm where the parameter is both k and the excluded minor. The second main contribution of this paper is to give a positive answer to this question.

At the core of many algorithms on excluded minor classes is a deep decomposition theorem by Robertson and Seymour which states that any graph excluding a fixed minor can be decomposed into a tree whose bags are almost embeddable into a surface that almost has bounded genus. In [4], Demaine et al. give a polynomial time algorithm for computing such decompositions. Grohe [12] derived from Robertson and Seymour's work a different decomposition that is often easier to use in the design of algorithms. He showed that any graph excluding a fixed minor can be decomposed into a tree whose bags have bounded local tree-width after removal of a constant number of elements. The main technical contribution of this paper is to show that we can compute such a decomposition in time $f(|H|) \cdot n^{\mathcal{O}(1)}$, where H is the excluded minor. We use this to show the following theorem.

Theorem 1.3 *Let $\mathcal{C}$ be a class of graphs excluding the fixed minor H. Then any first-order definable decision problem can be solved in time $f(|H|, |\varphi|) \cdot n^{\mathcal{O}(1)}$, where f is a computable function, φ is the sentence defining the decision problem, and H is the excluded minor.*

This result immediately implies fixed-parameter algorithms for problems such as the k-dominating set problem, where now the parameter is k and the excluded minor. On the other hand, the new decomposition can also be used directly to obtain faster algorithms for a variety of problems on excluded minor classes.

2 Preliminaries

We denote the set of real numbers by $\mathbb{R}$, the set of integers by $\mathbb{Z}$, and the set of positive integers (natural numbers) by $\mathbb{N}$. For all $n \in \mathbb{N}$, by $[n]$ we denote the set $\{1, \dots, n\}$, and for all $m, n \in \mathbb{Z}$, by $[m, n]$ we denote the set $\{m, \dots, n\}$ (the empty set if $m > n$). Occasionally, we use $[0, 1]$ to denote the unit interval $\{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$ instead of the set $\{0, 1\}$; it will always be clear from the context what is meant.

Graphs are undirected and simple. If G is a graph, we write $V(G)$ for its vertex set and $E(G)$ for its set of edges. For any set $X \subseteq V(G)$ we write $G[X]$ for the subgraph of G induced by X, and we let $G \setminus X = G[V(G) \setminus X]$. For a set $F \subseteq E$, we write $G - F$ for the graph $(V, E \setminus F)$. For every set S, by $K[S]$ we denote the complete graph with vertex set S. Furthermore, for every $k \in \mathbb{N}$ we let $K_k = K\big[[k]\big]$.

The *distance* $d^G(v, w)$ between two vertices v, w of a graph G is defined to be the length of the shortest path from v to w. For nonempty sets $X, Y \subseteq V(G)$, we let $d^G(X, Y) = \min\{d^G(x, y) \mid x \in X, y \in Y\}$, and similarly we define the distance $d^G(v, X)$ of a vertex $v \in V(G)$ from a nonempty set $X \subseteq V(G)$. We write $\rho(G) := \min_{v \in V(G)} \max_{v' \in V(G)} d^G(v, v')$ for the radius of G.

For every nonempty set $X \subseteq V(G)$ and every $r \geq 1$ we define the *r-neighborhood of X* and the *r-sphere around X* by

$$N_r^G(X) = \{v \in V(G) \ : \ d^G(v, X) \leq r\},$$
$$S_r^G(X) = \{v \in V(G) \ : \ d^G(v, X) = r\}.$$

For a vertex $v \in V(G)$, we let $N_r^G(v) = N_r^G(\{v\})$ and $S_r^G(v) = S_r^G(\{v\})$. For a nonempty subgraph $H \subseteq G$ we let $N_r^G(H) = N_r^G(V(H))$ and $S_r^G(H) = S_r^G(V(H))$. For 1-neighborhoods and 1-spheres, we write N^G and S^G instead of N_1^G and S_1^G, respectively.

A *separation* of a graph G is a pair (X, Y) of subsets of $V(G)$ such that $G = G[X] \cup G[Y]$. The set $X \cap Y$ is called the *separator* of the separation (X, Y), and its cardinality is the *order* of the separation.

We write $G \preceq H$ to denote that G is a minor of H. A class $\mathcal{C}$ of graphs is *minor-closed* if it is downward closed under $\preceq$. $\mathcal{C}$ is an *excluded minor class* if its minor-closure is not the class of all graphs. Note that every class $\mathcal{C}$ of graphs that excludes a fixed minor excludes a fixed clique as a minor. For any such class $\mathcal{C}$, the *excluded clique number* of $\mathcal{C}$ is the least $k \in \mathbb{N}$ such that $K_k \not\preceq G$ for any $G \in \mathcal{C}$.

Theorem 2.1 (Robertson and Seymour [14]) *The following problem is fixed-parameter tractable:*

p-MINOR	
Input:	Graphs G, H.
Parameter:	$\lvert V(H)\rvert$.
Problem:	Decide if $H \preceq G$.

More precisely, there is a computable function f and an algorithm that solves the problem in time $f(k) \cdot n^3$, where $k = |V(H)|$ and $n = |V(G)|$.

A *curve* in the plane $\mathbb{R}^2$ is the image of a continuous function $f : [0, 1] \to \mathbb{R}^2$ (here $[0, 1]$ denotes the unit interval of real numbers). The *endpoints* of the curve are $f(0)$ and $f(1)$, and its *interior* is the set of all other points. The curve is *simple* if f is one-to-one. A *plane graph* is a graph Γ whose vertices are distinct elements of the plane and whose edges are simple curves such that the endpoints of each edge are the two vertices incident with it, and the interior of each edge is disjoint from the vertex set and from all other edges. Abusing notation, we also write Γ to denote the point set $V(\Gamma) \cup \bigcup E(\Gamma) \subseteq \mathbb{R}^2$. We say that Γ is *embedded* in a subset $X \subseteq \mathbb{R}^2$ if $\Gamma \subseteq X$.

3 Tree-Decompositions

A *tree decomposition* of a graph G is a pair $D = (T, B)$, where T is a tree and B is a mapping that associates with every node $t \in V(T)$ a set $B_t \subseteq V(G)$ such that $G = \bigcup_{t \in V(T)} G[B_t]$, and for every $v \in V(G)$ the set $B^{-1}(v) = \{t \in V(T) \mid v \in B(t)\}$ is connected in T. The sets B_t, for $t \in V(T)$, are called the *bags* of the decomposition D. For a subset $U \subseteq V(T)$ we let $B(U) = \bigcup_{t \in U} B_t$, and for a subtree or forest $T' \subseteq T$ we let $B(T') = \bigcup_{t \in V(T')} B_t$.

With each bag B_t of a tree decomposition (T, B) of a graph G we associate two graphs:

- The *closure* of B_t is the graph $\overline{B}_t = G[B_t] \cup \bigcup_{u \in S^T(t)} K[B_t \cap B_u]$.
- The *companion* of B_t is the graph $\widehat{B}_t$ obtained from $G[B_t]$ by adding new vertices x_u for all $u \in S^T(t)$ and edges from x_u to all $v \in B_t \cap B_u$.

A tree decomposition is *strongly over* a class $\mathcal{C}$ of graphs if the closures of all its bags belong to $\mathcal{C}$, and it is *weakly over* $\mathcal{C}$ if the companions of all its bags belong to $\mathcal{C}$. Strong and weak decompositions over a class of graphs differ in how much the interaction between a bag and the rest of the graph is taken into account within the bag. Strong decompositions are better behaved, because the decompositions are minor monotone, that is, if C is a class of graphs closed under taking minors and D is the class of all graphs having a tree decomposition strongly over C, then D is also closed under taking minors. However, we only get an algorithm that computes weak decompositions, but fortunately that is good enough for our purposes.

The proof of the following lemma is straightforward:

Lemma 3.1 *Let $(T, (B_t)_{t \in V(T)})$ be a decomposition of a graph G, and let $t \in V(T)$. Then for every connected subgraph C of G the graph $\overline{B}_t[V(C) \cap B_t]$ is either empty or connected.*

For a tree T and an edge $e = \{t, u\} \in E(T)$, by T_{tu} and T_{ut} we denote the two connected components of $T - \{e\}$ such that $u \in V(T_{tu})$ and $t \in V(T_{ut})$. (Thus the indices determine which way we are looking from e.) The following lemma is well-known and easy to prove.

Lemma 3.2 *Let (T, B) be a tree decomposition of a graph G and $\{t, u\} \in E(T)$. Then $\big(B(T_{tu}), B(T_{ut})\big)$ is a separation of G with separator $B_t \cap B_u$.*

Let $D = (T, (B_t)_{t \in V(T)})$ be a tree decomposition of a graph G. For every edge $e = \{t, u\} \in E(T)$, we call $\big(B(T_{tu}), B(T_{ut})\big)$ the separation *at* e. We call the collection of all such separations the separations *of* D. Similarly, we call $B_t \cap B_u$ the separator *at* e and the collection of all these separators the separators *of* D. The *order* of the decomposition D is defined to be the maximum of the cardinalities of its separators.

The *width* of the decomposition is the number $\mathrm{width}(D) = \max\{|B_t| \mid t \in V(T)\} - 1$. The *tree width* of a graph G is the number

$$\mathrm{tw}(G) = \min\{\mathrm{width}(D) \mid D \text{ tree decomposition of } G\}.$$

The *local tree width* of G is the function $\mathrm{ltw}(G, \cdot)$ defined by $\mathrm{ltw}(G,r) = \max\Big\{ \mathrm{tw}\left(G[N_r^G(v)]\right) \ : \ v \in V(G)\Big\}$. A class $\mathcal{C}$ of graphs has *bounded local tree-width* if there is a function $f : \mathbb{N} \to \mathbb{N}$ that dominates the local tree-width of every $G \in \mathcal{C}$. For all nonnegative integers λ, μ we let

$$\mathcal{L}(\lambda) = \big\{G \ : \ \forall H \preceq G \, \forall r \geq 0 : \ \mathrm{ltw}(H,r) \leq \lambda \cdot r\big\},$$

$$\mathcal{L}(\lambda,\mu) = \big\{G \ : \ \exists X \subseteq V(G) \text{ s. th. } \begin{array}{c}|X| \leq \mu \text{ and} \\ G \setminus X \in \mathcal{L}(\lambda)\end{array} \big\}.$$

The following result is a consequence of Robertson and Seymour's deep structure theorem for graphs with excluded minors proved in [15].

Theorem 3.3 (Grohe [12]) *There are computable functions $\lambda, \mu : \mathbb{N} \to \mathbb{N}$ such that for every $k \in \mathbb{N}$, every graph G with $K_k \not\preceq G$ has a tree decomposition that is strongly over $\mathcal{L}(\lambda(k), \mu(k))$.*

The previous theorem can be used to obtain algorithms for various problems on excluded minor classes. These algorithms usually work along the following lines. Given a graph with a fixed forbidden minor, the first step is to compute the tree decomposition over a class $\mathcal{L}(\lambda, \mu)$. Then, for each block of the decomposition, the μ elements that need to be removed to obtain a graph with small local tree-width are computed. The problem, for instance the dominating set problem, is then solved on each of these blocks using methods on minor closed classes with bounded local tree-width. The results are then extended to the blocks with the μ removed elements put back in and then to the whole tree. For this approach to work we need to be able to a) compute the tree decomposition over the class $\mathcal{L}(\lambda, \mu)$ and b) compute, for each block, the μ-elements that need to be removed to obtain a graph in $\mathcal{L}(\lambda)$.

Polynomial-time algorithms for both steps are known, but the exponent of the polynomials depend on the excluded minor, for instance on the number μ of elements that need to be removed. In the following two sections, we show that both steps are fixed parameter tractable if the parameter is the excluded minor.

In the next section we give an algorithm to compute the decomposition within the desired running time. In the section thereafter we show that the second step can be computed within suitable time bounds.

4 Computing tree-decompositions over $\mathcal{L}(\lambda, \mu)$

The goal of this section is to prove the following theorem:

Theorem 4.1 *There are computable functions $f, \lambda, \mu : \mathbb{N} \to \mathbb{N}$ and an algorithm that, given a graph G with $K_k \not\preceq G$, computes a tree decomposition of G that is weakly over $\mathcal{L}(\lambda(k), \mu(k))$ in time $f(k) \cdot n^{O(1)}$.*

Let us first give a high level description of the proof. The algorithm to compute the tree decompositions is actually quite simple. Essentially we repeatedly compute small separators in the input graph G to obtain a clique-sum decomposition of G of order k. It turns out that we can take any separator of order k in G, as long as the graph is separated into two parts which are both not too small, where we can take "not too small" to mean at least of order $k!$.

We then associate with any such clique-sum decomposition a certain tree decomposition and show that it can easily be computed given the clique-sum decomposition. We call these k-CMS tree decompositions.

The main and difficult part of the proof consists in showing that these k-CMS tree decompositions actually are tree decompositions that are weakly over $\mathcal{L}(\lambda, \mu)$ for some $\lambda := \lambda(k)$ and $\mu := \mu(k)$.

A separation (X, Y) of order $k \geq 1$ of some graph is *substantial* if $|X|, |Y| > k!$. A separation (X, Y) of order 0 is substantial if $|X|, |Y| > 0$. (X, Y) is a *minimum substantial separation* if it is substantial, and if there is no substantial separation of smaller order.

Lemma 4.2 *There is an fpt-algorithm with parameter k that decides whether a given graph G has a substantial separation of order at most k and computes a minimum substantial separation if there is one of order at most k.*

Proof. We shall design a recursive fpt-algorithm for the following more general problem:

Input:	Graph G, sets $P, Q \subseteq V(G)$, and nonnegative integers k, k', ℓ, m with $k' \leq k$ and $\ell, m \leq k!$.
Parameter:	k.
Problem:	Decide whether G has a separation (X, Y) of order k' such that $P \subseteq X$ and $Q \subseteq Y$ and $|X \setminus Y| \geq \ell$ and $|Y \setminus X| \geq m$ and compute such a separation if there is one.

Let G, P, Q, k, k', ℓ, m be an instance of the problem. Without loss of generality we may assume that $k \geq 2$. Let $n = |V(G)|$. If $n < k' + \ell + m$ no substantial separator with the desired properties exists, and the algorithm reports failure. So suppose that $n \geq k' + \ell + m$. If $k' = 0$, it is straightforward to check whether the connected components of G can be arranged into a separation with the desired properties, so we further assume that $k' > 0$.

Using standard techniques, the algorithm computes some separation (X, Y) of G of order k' with $P \subseteq X$ and $Q \subseteq Y$. If no such separation exists, then the algorithm reports failure. If $|X \setminus Y| \geq \ell$ and $|Y \setminus X| \geq m$, the algorithm returns the separation (X, Y). In the following, we assume without loss of generality that $|X \setminus Y| < \ell$. Let $S = X \cap Y$.

Now for all separations (X_1, X_2) of $G[X]$ of order at most k' with $P \cap X \subseteq X_1$ and $Q \cap X \subseteq X_2$, we recursively call our algorithm on the instance $G', P', Q', k, k'', \ell', m'$ defined as follows: Let $T = X_1 \cap X_2$ and $S_1 = (S \cap X_1) \setminus T$, $S_2 = (S \cap X_2) \setminus T$.Let

- $G' = G[Y] \setminus T$,
- $k'' = k' - |T|$,
- $P' = (P \cup S_1) \cap V(G')$,
- $Q' = (Q \cup S_2) \cap V(G')$,
- $\ell' = \max\{0, \ell - |X_1 \setminus (X_2 \cup S)|\}$ and
- $m' = \max\{0, m - |X_2 \setminus (X_1 \cup S)|\}$.

If the recursive call returns a separation (Y_1, Y_2), then our algorithm returns the separation $(X_1 \cup Y_1, X_2 \cup Y_2)$, which can easily be seen to have the desired properties. If the recursive calls fail for all (X_1, X_2), then the algorithm reports failure.

Let $t_k(n, k, \ell, m)$ denote the worst case running time of the algorithm in terms of the parameters n, ℓ, m, k, and let $T(n, k, s) = \max\{t(n, k, \ell, m) \mid 0 \leq \ell, m \leq s, \ell + m = s\}$. (The parameter k' is irrelevant for our running time analysis.) We get the following recurrence in terms of s:

$$\begin{aligned} T(n, k, 0) &= n^{O(1)} && \text{for } n \geq 0 \\ T(n, k, s) &= O(1) && \text{for } n < s \\ T(n, k, s) &= O\big(2^{k!} \cdot T_k(n, s-1)\big) && \text{for } n \geq s \geq 1. \end{aligned}$$

To obtain this recurrence, we put a crude upper bound of $2^{k!}$ on the number of separations (X_1, X_2) of $G[X]$ and ignore the fact that not only s, but n also decreases in the recursive calls. Since $s \leq 2k!$, this yields an fpt-bound on the running time. □

Let G be a graph. The graph G is a *clique sum* of graphs A, B (we write $G = A \oplus B$) if there is a separation (X, Y) of G such that $A = G[X] \cup K[X \cap Y]$ and $B = G[Y] \cup K[X \cap Y]$. We write $G = A \oplus_k B$ to indicate that $G = A \oplus B$ and the order of the corresponding separation is at most k, and we similarly write $G = A \oplus_S B$ to indicate that the separator is S. A *clique-sum decomposition* of G is a pair (T, A) where T is a rooted binary tree and H is a mapping that associates a graph A_t with every node $t \in V(T)$ such that:

1. $A_r = G$ for the root r of T;
2. $A_t = A_{t_1} \oplus A_{t_2}$ for every node $t \in V(T)$ with children t_1, t_2.

The separation corresponding to the clique sum $A_t = A_{t_1} \oplus A_{t_2}$ is called the separation *at* node t, and its separator $V(A_{t_1}) \cap V(A_{t_2})$ is called the separator *at* node t. The graphs A_t for the leaves t of T are called the *pieces* of the decomposition. The *order* of a clique-sum decomposition (T, A) is the least k such that $A_t = A_{t_1} \oplus_k A_{t_2}$ for every node $t \in V(T)$ with children t_1, t_2. The *height* of the decomposition is the height of the tree T, that is, the maximum length of a path from the root of T to a leaf.

A clique-sum decomposition (T, A) is an *MS decomposition* if all its separations are minimum substantial. It is a *complete MS decomposition of order k* (for short: *k-CMS decomposition*) if the order of the decomposition is at most k, and if for all leaves t of T the graph A_t has no substantial separation of order at most k. The proof of the next lemma follows directly from Lemma 4.2.

Lemma 4.3 *There is an fpt-algorithm with parameter k that computes a k-CMS-decomposition for a given graph G.*

Definition 4.4 *Let $D = (T, A)$ be a clique-sum decomposition of a graph G.*

A tree decomposition associated *with the clique decomposition D is a tree decomposition $D' = (T', A')$ such that*

- *the closures of the bags of D' are the pieces of D (and hence the bags of D' are the vertex sets of the pieces of D).*
- *for every node $t \in V(T)$ there is an edge $e \in E(T')$ such that for the separation (X, Y) at t and the separation (X', Y') at e it holds that $X' \cap V(A_t) = X$ and $Y' \cap V(A_t) = Y$.*
- *for every edge $e \in E(T')$ there is a node $t \in V(T)$ such that for the separation (X, Y) at t and the separation (X', Y') at e it holds that $X' \cap V(A_t) = X$ and $Y' \cap V(A_t) = Y$.*
- *D and D' have the same separators and hence the same order.*

Lemma 4.5 *For every clique-sum decomposition $D = (T, A)$ of a graph G there is an associated tree decomposition $D' = (T', A')$.*

Proof. By induction on the height of $t \in V(T)$, we define a tree T'_t whose vertex set is the set of all leaves of T that are in the subtree with root t. Let $t \in V(T)$. If $t \in V(T)$ is a leaf of T, we let T'_t be the tree $(\{t\}, \varnothing)$. So suppose that t has children t_1, t_2. Let (X_1, X_2) be the separation at t such that $V(A_{t_1}) \subseteq X_1$ and $V(A_{t_2}) \subseteq X_2$. Let $S = X \cap Y$. As S induces a clique in both A_{t_1} and A_{t_2}, there must be leaves u_1, u_2 of T such that for $i = 1, 2$ the leaf u_i is in the subtree of T with root t_i, and $S \subseteq V(A_{u_i})$.

Applying the induction hypothesis, let T'_{t_i} be the trees constructed for T_{t_i}. Thus $u_i \in V(T'_{t_i})$. We let T'_t be the tree obtained from the disjoint union of T'_{t_1} and T'_{t_2} by adding an edge from u_1 to u_2.

Now let $T' = T'_r$ for the root r of T. For every $t \in V(T)$ we let $A'_t = V(A_t)$. It is easy to see that (T', A') is a tree decomposition of G with the desired properties. □

A *k-CMS tree decomposition* of a graph G is a tree decomposition associated with a k-CMS decomposition.

Corollary 4.6 *There is an fpt-algorithm with parameter k that computes a k-CMS tree decomposition for a given graph G.*

The next lemma shows that any κ-CMS tree-decomposition actually is a tree-decomposition weakly over some $\mathcal{L}(\lambda, \mu)$. Due to space limitations, we refer the reader to the full version of the paper for a proof of the lemma.

Lemma 4.7 *Let $\lambda, \mu \in \mathbb{N}$ and $\kappa = \lambda + \mu + 1$. Let G be a graph that has a tree decomposition over $\mathcal{L}(\lambda, \mu)$, and let $D = (T, A)$ be a κ-CMS tree decomposition of G. Then the companions of all bags of D are in $\mathcal{L}(2\kappa!, \mu)$.*

As a consequence we obtain Theorem 4.1, the main theorem of this section.

5 Graphs of almost bounded local tree-width

The goal of this section is to prove that there is an fpt-algorithm that, given a graph in $\mathcal{L}(\lambda, \mu)$, computes a set of at most μ vertices such that the graph obtained by deleting these vertices is in $\mathcal{L}(\lambda')$ for a suitable λ' (Corollary 5.7). The proof is based on a deep structure theorem (Theorem 5.1) by Robertson and Seymour that essentially says that any graph either has small tree-width, or contains a large clique minor, or consists of a large planar wall to which subgraphs of small tree-width are attached in a not too complicated way together with a bounded number of elements which may have arbitrary connections to the rest. As the problem can easily be solved if G has low tree-width and no graph in $\mathcal{L}(\lambda, \mu)$ contains a large clique minor, we only have to deal with the third case. This, however, requires some work. Due to space limitations we only sketch the result here. A full proof can be found in the full version of the paper.

5.1 Walls and Layouts

An *elementary wall of height $h \geq 1$* is a graph as illustrated in Figure 5.1. A *wall of height h* is a subdivision of an elementary wall of height h. The *perimeter* of a wall is the boundary cycle (cf. Figure 5.2). A wall *in* a graph G is a wall H that is a subgraph of G. Note that, up to homeomorphisms, walls have unique embeddings in the sphere. For walls of height 1, this is obvious, and for walls of height $h \geq 2$ this follows from a well known theorem due to Tutte stating that 3-connected graphs have unique embeddings, because walls of height ≥ 2 are subdivisons of 3-connected graphs.

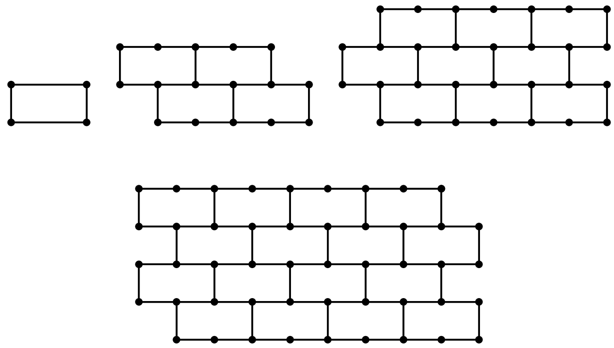

Figure 5.1. Elementary walls of height 1–4

For a subgraph D of a graph G, we let $\partial^G D$ be the set of all vertices of D that are incident with an edge in $E(G) \setminus E(D)$.

In the following, let H be a wall of height at least 2 in a graph G, and let P be the perimeter of H. Let K' be the unique connected component of $G \setminus P$ that contains $H \setminus P$. The graph $K = K' \cup P$ is called the *compass* of H in G. A *layout of K (with respect to the wall H in G)* is a family $(C, D_1, \ldots, D_m)$ of connected subgraphs of K such that:

1. $K = C \cup D_1 \cup \ldots \cup D_\ell$;
2. $H \subseteq C$, and there is no separation (X, Y) of C of order ≤ 3 with $V(H) \subseteq X$ and $Y \setminus X \neq \varnothing$;
3. $\partial^G D_i \subseteq V(C)$ for all $i \in [m]$;
4. $|\partial^G D_i| \leq 3$ for all $i \in [m]$;
5. $\partial^G D_i \neq \partial^G D_j$ for all $i \neq j \in [m]$.

We let $\overline{C}$ be the graph obtained from C by adding new vertices $d_1, \ldots, d_m$ and, for $1 \leq i \leq m$, edges between d_i and the vertices in $\partial^G D_i$ and edges between all vertices in $\partial^G D_i$. Hence, for each $i \in [m]$ the vertex d_i together with the (at most 3) vertices in $\partial^G D_i$ form a clique. We call $\overline{C}$ the *core* of the layout and $D_1, \ldots, D_m$ its *extensions*. The layout $(C, D_1, \ldots, D_m)$ is *flat* if its core $\overline{C}$ is planar. Note that this implies that the core has an embedding in the plane that extends the "standard planar embedding" of the wall H (as shown in Figures 5.1 and 5.2), because the wall H has a unique embedding into the sphere. We call the wall H *flat* (in G) if the compass of H has a flat layout.

Theorem 5.1 (Robertson and Seymour [14]) *There are computable functions $f, g : \mathbb{N}^2 \to \mathbb{N}$ and an algorithm A that, given a graph G and nonnegative integers k, h, computes either*

1. *a tree decomposition of G of width $f(k,h)$, or*

2. *a K_k-minor of G, or*

3. *a subset $X \subseteq V(G)$ with $|X| < \binom{k}{2}$, a wall H of height h in $G \setminus X$, and a flat layout $(C, D_1, \dots, D_m)$ of the compass of H in $G \setminus X$ such that the tree width of each of the extensions $D_1, \dots, D_m$ is at most $f(k,h)$.*

Furthermore, the running time of the algorithm is bounded by $g(k,h) \cdot n^2$, where n is the number of vertices of the input graph G.

Proof. This is (essentially) Lemma (9.8) of [14]. Concerning the uniformity, see the remarks at the end of [14] (on page 109). $\square$

5.2 Algorithmic Aspects

We now prove the main algorithmic result of this section. It is based on Theorem 5.1 and the following two lemmas, whose proofs can be found in the full version of the paper.

For every $\ell \geq 1$, the $(\ell \times \ell)$-*pyramid* is the graph obtained from the $(\ell \times \ell)$-grid by adding a new vertex a (the *apex*) and edges from a to all vertices of the grid. For every $\lambda \geq 1$, we let

$$\mathcal{K}(\lambda) = \{G \mid \begin{array}{l}\text{the } (\lambda+1) \times (\lambda+1)\text{-pyramid} \\ \text{is not a minor of } G\end{array}\},$$

$$\mathcal{K}(\lambda, \mu) = \{G \mid \begin{array}{l}\exists X \subseteq V(G) : \ |X| \leq \mu \\ \text{and } G \setminus X \in \mathcal{K}(\lambda)\end{array}\}.$$

The *bricks* of an elementary wall are the cycles of length 6, and the *bricks* of a wall are the subdivisions of the bricks of the corresponding elementary wall. Two bricks are *adjacent* if they are distinct and have a nonempty intersection. We can assign coordinates to the bricks of a wall as shown in Figure 5.2. In the folllowing, we always assume that we have fixed the coordinates in our walls. Let H be a wall. For $1 \leq i, j \leq h$, let B_{ij} denote the brick of H with coordinates (i,j). A subgraph $H' \subseteq H$ is a *subwall* if there exist $h' \leq h$ and $i, j \leq h - h'$ such that

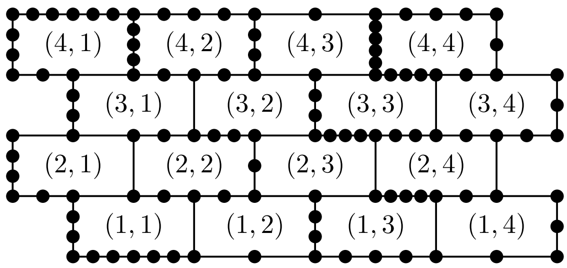

Figure 5.2. The coordinates of the bricks of a wall

$$H' = \bigcup_{\substack{i+1 \leq i' \leq i+h' \\ j+1 \leq j' \leq j+h'}} B_{i'j'}.$$

Hence a subwall consists of consecutive bricks both in horizontal and vertical direction.

Lemma 5.2 *There is a computable function $f : \mathbb{N}^4 \to \mathbb{N}$ such that the following holds for all $h', \kappa, \lambda, \mu \in \mathbb{N}$: Let $G \in \mathcal{K}(\lambda, \mu)$, $X \subseteq V(G)$ with $|X| \leq \kappa$, and let H be a flat wall of height at least $f(h', \kappa, \lambda, \mu)$ in $G \setminus X$. Then*

1. *either there is a vertex $x \in X$ such that $G \setminus \{x\} \in \mathcal{K}(\lambda, \mu - 1)$,*

2. *or H has a subwall H' of height h' such that the compass of H' in G has an empty intersection with X.*

Furthermore, there is an algorithm with running time $f(h', \kappa, \lambda, \mu) \cdot n^{\mathcal{O}(1)}$ that tests which of the two cases applies and either computes the subwall H' or the element $x \in X$ so that $G \setminus \{x\} \in \mathcal{K}(\lambda, \mu - 1)$.

Lemma 5.3 *There is a computable function $f : \mathbb{N}^2 \to \mathbb{N}$ such that for all $\lambda, \mu \geq 1$ the following holds: Let G be a graph, H a wall of height $f(\lambda, \mu)$ in G, and $(C, D_1, \dots, D_m)$ be a flat layout of the compass K of H such that every extension D_i has tree width at most $\lambda - 4$. Furthermore, let z be a vertex of the central brick of H. Then*

$$G \in \mathcal{K}(\lambda, \mu) \leftrightarrow G \setminus \{z\} \in \mathcal{K}(\lambda, \mu)$$

and if $X \subseteq V(G) \setminus \{z\}$ is such that $|X| \leq \mu$ and $G \setminus (X \cup \{z\}) \in K(\lambda)$ then $G \setminus X \in K(\lambda)$.

We are now ready to prove the main result of this section.

Theorem 5.4 *There are computable functions $f, g : \mathbb{N}^2 \to \mathbb{N}$ and an algorithm that, given a graph $G \in \mathcal{K}(\lambda, \mu)$, computes a set $X \subseteq V(G)$ with $|X| \leq \mu$ such that $G \setminus X \in \mathcal{K}(g(\lambda, \mu))$ in time $f(\lambda, \mu) \cdot n^{O(1)}$.*

Proof. Let $k := (\lambda+1)^2 + \mu + 2$ and let h be "big enough", so that the recursive calls to the algorithms work. (It is easy but tedious to compute the correct value for h.) By Theorem 5.1, G either has tree-width at most $f(k,h)$, or contains a K_k-minor, or there is a subset $X \subseteq V(G)$ with $|X| < \binom{k}{2}$, a wall H of height h in $G \setminus X$, and a flat layout $(C, D_1, \dots, D_m)$ of the compass of H in $G \setminus X$ such that the tree width of each of the extensions $D_1, \dots, D_m$ is at most $f(k,h)$. As no graph in $\mathcal{K}(\lambda, \mu)$ can contain $K_{(\lambda+1)^2+\mu+2}$ as a minor and the problem can easily be solved for graphs of bounded tree-width, we only have to deal with Case 3). In this case, the algorithm in Theorem 5.1 actually returns the wall and its layout. We can now apply the algorithm from Lemma 5.2. It either returns an element $x \in X$ so that $G \setminus \{x\} \in \mathcal{K}(\lambda, \mu - 1)$ or a subwall H' whose compass has an empty intersection with X. In the first case we have found one of the μ elements and repeat the process on the graph $G \setminus \{x\}$. In the latter case, we call the algorithm recursively on the smaller graph $G \setminus \{z\}$ for some vertex z

of the central brick of H'. By Lemma 5.3, $G \in \mathcal{K}(\lambda, \mu)$ if, and only if, $G \setminus \{z\} \in \mathcal{K}(\lambda, \mu)$ and, in addition, if the recursive call to the algorithm on $G \setminus \{z\}$ returns a set X then, by Lemma 5.3 again, $G \setminus X \in \mathcal{K}(\lambda)$. □

The previous theorem gives an fpt algorithm to compute, given a graph $G \in \mathcal{K}(\lambda, \mu)$, a set X of vertices with $|X| \leq \mu$ such that $G \setminus X \in \mathcal{K}(\lambda)$. However, we aim at computing for a given graph $G \in \mathcal{L}(\lambda, \mu)$ a set X so that $G \setminus X \in \mathcal{L}(\lambda)$. The following lemma by Eppstein [6] and Demaine and Hajiaghayi [5] solves this problem.

Lemma 5.5 *There is a computable function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that for all $\lambda \in \mathbb{N}$: $\mathcal{L}(\lambda) \subseteq \mathcal{K}(\lambda) \subseteq \mathcal{L}(f(\lambda))$.*

Corollary 5.6 *There is a computable function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that for all $\lambda, \mu \in \mathbb{N}$: $\mathcal{L}(\lambda, \mu) \subseteq \mathcal{K}(\lambda, \mu) \subseteq \mathcal{L}(f(\lambda), \mu)$.*

Corollary 5.7 *There are computable functions $f, g : \mathbb{N}^2 \rightarrow \mathbb{N}$ and an algorithm that, given a graph $G \in \mathcal{L}(\lambda, \mu)$, computes a set $X \subseteq V(G)$ with $|X| \leq \mu$ such that $G \setminus X \in \mathcal{L}(g(\lambda, \mu))$ in time $f(\lambda, \mu) \cdot n^{O(1)}$.*

6 FO-model checking on graphs with excluded minors

In [7], Flum and Grohe show that the model-checking problem for first-order logic is fixed-parameter tractable with parameter φ on any class of graphs with an excluded minor. In their proof, Flum and Grohe use the tree-decomposition of graphs excluding a fixed minor guaranteed by Theorem 3.3. The proof can easily be modified to work with tree-decompositions that are weakly over $\mathcal{L}(\lambda, \mu)$. As a consequence, we immediately get the following theorem.

Theorem 6.1 *The following problem*

FO-MODEL-CHECKING	
Input:	Graphs G, H such that $H \not\preceq G$ and $\varphi \in$ FO.
Parameter:	$\lvert\varphi\rvert + \lvert H\rvert$.
Problem:	Decide whether $G \models \varphi$.

is fixed-parameter tractable.

The theorem implies that problems such as the dominating or independent set problem become fixed-parameter tractable when the parameter is the excluded minor and the size of the solution, e.g. the size of the independent set. This improves over previously known results, where the minor was not part of the parameter and determined the exponent of the polynomials.

Another consequence of the methods developed in the previous sections is the following result. For any function $f : \mathbb{N} \rightarrow \mathbb{N}$, let $\mathcal{C}_f$ be the class of graphs G such that the excluded clique number of G is at most $f(|G|)$.

Theorem 6.2 *There is an unbounded function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that first-order model checking is fixed-parameter tractable on $\mathcal{C}_f$.*

The algorithms presented in the previous sections depend in various ways on the excluded minor H. For instance H determines the numbers λ and μ used throughout the sections. We therefore refrain from giving expicit bounds on the function f whose existence is proved in Theorem 6.2.

7 Locally Excluding a Minor

In [7] Flum and Grohe prove that first-order model checking is fixed-parameter tractable on any class of graphs with an excluded minor. In the same year, Frick and Grohe [10] established the analogous result for graph classes with bounded local tree-width. As the two structural properties are incomparable, i.e. there are classes of graphs excluding a minor but with unbounded local tree-width and vice-versa, it is a natural question, whether there is a common generalisation of excluded minors and bounded local tree-width on which first-order model checking is still fixed-parameter tractable. In this section we present such a generalisation.

7.1 Definition

Definition 7.1 *A class $\mathcal{C}$ of graphs* locally excludes a minor *if for every $r \in \mathbb{N}$ there is a graph H_r so that if $G \in \mathcal{C}$ and $v \in V(G)$ then $H_r \not\preceq N_r^G(v)$, i.e. H_r is not a minor of the r-neighbourhood of v in G.*

It is easily seen that any class of graphs with bounded local tree-width locally excludes a minor as does any class of graphs excluding a fixed minor.

Proposition 7.2 *If $\mathcal{C}$ is a class of graphs with*

- *bounded local tree-width or*
- *excluding a fixed minor,*

then $\mathcal{C}$ locally excludes a minor. The converse is not true, i.e. there are classes of graphs locally excluding a minor with local tree-width or whose minor closure is the class of all graphs.

Note, however, that the concept of excluding slowly growing minors as we considered in Section 6 is incomparable to locally excluding a minor. In [17], Nešetřil and de Mendez introduce the concept of graph classes with bounded expansion. This also generalises the concept of excluded minors. However, the concepts of locally excluded minors and bounded expansion are mutually incomparable, as there are examples separating the two concepts in either direction.

7.2 First-Order Model-Checking

We show next that first-order model checking is fixed-parameter tractable on any class of graphs locally excluding a minor.

Gaifman [11] showed that any first-order sentence is equivalent to a Boolean combination of basic-local sentences. We recall the definition. For every $r \geq 0$ we will use formulas $d(x, y) \leq r$ and $d(x, y) > r$ to say that the distance between x and y is at most r and greater than r, respectively. Clearly, these are easily first-order definable. If $\varphi(x)$ is a first-order formula, then $\varphi^{N_r(x)}(x)$ is the formula obtained from φ by relativising the quantifiers in φ to the r-neighbourhood of x, i.e. replacing $\forall y\psi$ by $\forall y(d(x, y) \leq r \rightarrow \psi)$ and $\exists y\psi$ by $\exists y(d(x, y) \leq r \wedge \psi)$. A formula $\psi(x)$ of the form $\varphi^{N_r(x)}(x)$ is called *r-local*. The essential property of an r-local formula is that its truth value at a vertex x in G only depends on the r-neighbourhood of x in G.

Theorem 7.3 (Gaifman [11]) *Every first-order sentence is equivalent to a Boolean combination of basic-local sentences, i.e. a Boolean combination of sentences of the form*

$$\exists x_1 \ldots \exists x_k \big(\bigwedge_{1 \leq i < j \leq k} d(x_i, x_j) > 2r \wedge \bigwedge_{1 \leq i \leq k} \vartheta(x_i) \big),$$

for suitable $r, k > 0$ and an r-local formula $\vartheta(x)$.

We are now ready to prove the main result of this section.

Theorem 7.4 *Let $\mathcal{C}$ be a class of graphs locally excluding a minor. Then the following problem*

FO-ModelChecking on $\mathcal{C}$	
Input:	$G \in \mathcal{C}$, $\varphi \in$ FO.
Parameter:	$\lvert\varphi\rvert$.
Problem:	Decide $G \models \varphi$.

is fixed-parameter tractable.

Proof. Let $\varphi \in$ FO be a sentence. By Gaifman's theorem 7.3, φ is equivalent to a Boolean combination of basic local sentences. Hence, to prove the theorem, it suffices to only consider the case where φ is a sentence of the form

$$\exists x_1 \ldots \exists x_k \bigwedge_{1 \leq i < j \leq k} d(x_i, x_j) > 2 \cdot r \wedge \bigwedge_{1 \leq i \leq k} \vartheta(x_i)$$

for $r, k \geq 1$ and an r-local formula ϑ.

As $\mathcal{C}$ locally excludes a minor, there is for every $s \in \mathbb{N}$ a graph H_s such that H_s is excluded in every s-neighbourhood of vertices in any member of $\mathcal{C}$. Let $G \in \mathcal{C}$.

The first step of the evaluation algorithm is to compute the set $P \subseteq V(G)$ of vertices v such that $G[N_r(v)] \models \vartheta(v)$. As $H_r \not\preceq G[N_r(v)]$, Theorem 6.1 implies that there is a computable function $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ such that checking whether $G[N_r(v)] \models \vartheta(v)$ can be done in time $f(|H_r|, |\vartheta|) \cdot |N_r(v)|^{\mathcal{O}(1)}$.

It remains to find a set of k elements of P whose distance is pairwise $> 2r$. For this, we proceed as follows. Set $Q := P$ and set $l := 0$. While $Q \neq \varnothing$ and $l < k$, choose an arbitrary element $a_l \in Q$, increase l to $l + 1$ and remove $N_r^G(a_l)$ from Q. If this process stops with $l = k$, we can accept, as then $\{a_1, \ldots, a_k\}$ is the required set. If $l = 0$, then $P := \varnothing$ and therefore ϑ is false in every r-neighbourhood of a vertex in G and hence $G \not\models \varphi$. Finally, if $0 < l < k$, we know that every $v \in P$ is contained in the $2r$-neighbourhood of some a_i, $1 \leq i \leq l$. Let $N := G\big[N_{2r}^G\big(\{a_1, \ldots, a_l\}\big)\big]$. By construction, the radius of N is at most $2r \cdot l$ and hence $H := H_{2rl} \not\preceq N$. By Theorem 6.1, there is a function $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ such that we can test whether $(N, P) \models \psi$ in time $g(|H|, |\psi|) \cdot |N|^{\mathcal{O}(1)}$, where $\psi := \exists x_1 \ldots \exists x_k \big(\bigwedge_{i=1}^{k} Px_i \wedge \bigwedge_{1 \leq i < j \leq k} d(x_i, x_j) > 2r \big)$. If $(N, P) \models \psi$, then there is a set of k vertices in P pairwise far apart and we can accept. Otherwise we reject.

Note that k and hence $|\psi|$, $|H_r|$, and $|H|$ only depend on φ and hence are bounded by the parameter $|\varphi|$. The algorithm correctly determines whether $G \models \varphi$ and has a total running time of $h(|\varphi|) \cdot |G|^{\mathcal{O}(1)}$, where $h : \mathbb{N} \rightarrow \mathbb{N}$ is a function that dominates the functions f and g above. ☐

As an immediate consequence of this we obtain that on classes $\mathcal{C}$ locally excluding a minor the following problems are fixed parameter tractable with parameter H: For every fixed graph H decide whether for a graph $G \in \mathcal{C}$: H has a homomorphism to G; H is a subgraph of G; H is an induced subgraph of G.

Furthermore, problems such as independent or dominating set and many others are fixed-parameter tractable on any class of graphs locally excluding a minor.

8 Conclusions

We introduce the notion of graph classes locally excluding a minor and prove that deciding first-order properties of such classes is fixed-parameter tractable. The result is particularly interesting because it unifies incomparable previous results for classes of bounded local tree width [10] and for classes with excluded minors [7] in a natural way. But the result is considerably stronger than just a combination of those two.

To prove the result, we need to strengthen the fixed-parameter tractability result for classes with excluded minors [7] in such a way that the size of the excluded minor can now be taken as a parameter in the running time analysis. This implies fixed-parameter tractability results for problems such as dominating set and independent set, now parameterized by the size of the desired solution and the size of the excluded minor. Even though both problems have been intensely studied on restricted graph classes including classes with excluded minors (see e.g. [13, 1, 9, 8] and the references there), the existence of such algorithms was not known before.

Let us finally remark that algorithmic meta theorems like ours are not meant to be practical, as usually the dependence of the running time on the formula size is nonelementary and the hidden constants are enormous. One reason for the interest in such results is that they often provide an easy way to quickly check if a concrete problem is fixed-parameter tractable (see our remarks on dominating set and independent set above). The more significant reason for our interest in such meta theorems is that they yield a better understanding of the limits of general algorithmic techniques and, in some sense, the limits of tractability. In particular, they clarify the interactions between logic and combinatorial structure, which we believe to be fundamental for computational complexity.

References

[1] J. Alber, H. Fernau, and R. Niedermeier. Parameterized complexity: exponential speed-up for planar graph problems. *Journal of Algorithms*, 52(1):26–56, 2004.

[2] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 194 – 242. Elsevier, 1990.

[3] B. Courcelle, J. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Systems*, 33:125–150, 2000.

[4] E. Demaine, M. Hajiaghayi, and K. Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Symposium on Foundations of Computer Science (FOCS)*, pages 637–646, 2005.

[5] E. Demaine and M.T. Hajiaghayi. Equivalence of local treewidth and linear local treewidth and its algorithmic applications. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 840–849, 2004.

[6] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27:275–291, 2000.

[7] J. Flum and M. Grohe. Fixed-parameter tractability, definability, and model checking. *SIAM Journal on Computing*, 31:113 – 145, 2001.

[8] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[9] F. Fomin and D.M. Thilikos. A simple and fast approach for solving problems on planar graphs. In *Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 56–67, 2004.

[10] M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM*, 48:1184 – 1206, 2001.

[11] H. Gaifman. On local and non-local properties. In J. Stern, editor, *Herbrand Symposium, Logic Colloquium '81*, pages 105 – 135. North Holland, 1982.

[12] M. Grohe. Local tree-width, excluded minors, and approximation algorithms. *Combinatorica*, 23(4):613–632, 2003.

[13] M. Hajiaghayi. *The bidimensionality Theory and Its Algorithmic Applications*. PhD thesis, Department of Mathematics, Massachusetts Institute of Technology, 2005.

[14] N. Robertson and P.D. Seymour. Graph minors XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63:65–110, 1995.

[15] N. Robertson and P.D. Seymour. Graph minors XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 77:1–27, 1999.

[16] D. Seese. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science*, 2:505 – 526, 1996.

[17] J. Nešetřil and P. Ossona de Mendez. The grad of a graph and classes with bounded expansion. In *International Colloquium on Graph Theory*, pages 101 – 106, 2005.