Computing Excluded Minors

Isolde Adler Martin Grohe Stephan Kreutzer

Institut für Informatik, Humboldt Universität zu Berlin
{adler,grohe}@informatik.hu-berlin.de

Oxford University Computing Laboratory
kreutzer@comlab.ox.ac.uk

Abstract

By Robertson and Seymour's graph minor theorem, every minor ideal can be characterised by a finite family of excluded minors. (A *minor ideal* is a class of graphs closed under taking minors.) We study algorithms for computing excluded minor characterisations of minor ideals. We propose a general method for obtaining such algorithms, which is based on definability in monadic second-order logic and the decidability of the monadic second-order theory of trees. A straightforward application of our method yields algorithms that, for a given k, compute excluded minor characterisations for the minor ideal $\mathcal{T}_k$ of all graphs of tree width at most k, the minor ideal $\mathcal{B}_k$ of all graphs of branch width at most k, and the minor ideal $\mathcal{G}_k$ of all graphs of genus at most k.

Our main results are concerned with constructions of new minor ideals from given ones. Answering a question that goes back to Fellows and Langston [11], we prove that there is an algorithm that, given excluded minor characterisations of two minor ideals $\mathcal{C}$ and $\mathcal{D}$, computes such a characterisation for the ideal $\mathcal{C} \cup \mathcal{D}$. Furthermore, we obtain an algorithm for computing an excluded minor characterisation for the class of all apex graphs over a minor ideal $\mathcal{C}$, given an excluded minor characterisation for $\mathcal{C}$. (An apex graph over $\mathcal{C}$ is a graph G from which one vertex can be removed to obtain a graph in $\mathcal{C}$.) A corollary of this result is a uniform ftp-algorithm for the "distance k from planarity" problem.

1 Introduction

One of the most stunning consequences of Robertson and Seymour's graph minor theory is the algorithmic result stating that every minor ideal has a cubic time membership test. Here a *minor ideal* is a class of graphs closed under taking minors. A puzzling feature of this result is that it merely states the existence of an algorithm deciding membership in a minor ideal, without giving an explicit algorithm. For example, the class of all graphs that can knotlessly[1] be embedded into 3-space is a minor ideal and hence has a cubic time membership test by Robertson and Seymour's theorem, but to the best of our knowledge still no explicit membership test for this minor ideal is known. Let us analyse how this comes about. The cubic time membership test for minor ideals is the direct consequence of two other results due to Robertson and Seymour. The first is the well-known graph minor theorem [22], stating that every minor ideal is characterised by a finite family of excluded minors, that is, for every minor ideal $\mathcal{C}$ there are an $n \geq 1$ and graphs $H_1, \ldots, H_n$ such that a graph G belongs to $\mathcal{C}$ if and only if none of the graphs $H_1, \ldots, H_n$ is a minor of G. We call $\{H_1, \ldots, H_n\}$ an *excluded minor characterisation* of $\mathcal{C}$.[2] The second result states that for every graph H there is a cubic time algorithm deciding whether a given graph G has H as a minor [21]. Let us call such an algorithm an *H-minor test*. The algorithmic result is perfectly constructive; for every H, Robertson and Seymour construct an explicit cubic H-minor test. However, the graph minor theorem is not constructive; in general, we do not know how to obtain an excluded minor characterisation of a given minor ideal. For example, we do not know such a characterisation for the minor ideal of knotlessly embeddable graphs. Fellows and Langston [12] observed that there is no algorithm that, given a Turing machine which is a membership test for a minor ideal, computes an excluded minor characterisation of the ideal. Later, Courcelle et al. [5] showed that there is no algorithm that, given a sentence of monadic second-order logic which defines a minor ideal, computes an excluded minor characterisation of the ideal.

Fellows and Langston [9, 10, 11, 12] were the first to study these algorithmic issues related to the graph minor theorem (also see [15, 6]). They devised several strategies to overcome the "non-constructiveness" of algorithms based on the minor theorem. The most obvious, of course, is to compute excluded minor char-

[1] A *knotless embedding* of a graph into 3-space is an embedding where no cycle of the graph is non-trivially knotted. The class of knotlessly embeddable graphs only serves as examples in this introduction and will play no further role in the paper.

[2] Instead of *excluded minor characterisation*, the term *obstruction set* is often used in the literature.

acterisations of the classes in question. In [11], Fellows and Langston proposed a general method for computing excluded minor characterisations. It is based on a generalisation of the Myhill-Nerode theorem of formal language theory to certain "graph languages". Here, we propose a similar method that, instead of on Myhill-Nerode congruences, is based on definability in monadic second order logic and generalisations of the Büchi-Elgot-Trakhtenbrot theorem to graphs [4, 23]. Using our method, we can easily show that there are algorithms that, for a given k, compute excluded minor characterisations of the minor ideal $\mathcal{T}_k$ of all graphs of tree width at most k, the minor ideal $\mathcal{B}_k$ of all graphs of branch width at most k, and the minor ideal $\mathcal{G}_k$ of all graphs of genus at most k. These results are not new: The computability of excluded minor characterisations of the families $\mathcal{T}_k$ of bounded tree width follows easily from [16], and for the families $\mathcal{B}_k$ of bounded branch width it follows from [13] (and is also mentioned, without proof, in [2]). For the families $\mathcal{G}_k$ of bounded genus, it follows from [24] or a combination of [11] and [25] (cf. [18], p.222).

Our main results are concerned with constructions of new minor ideals from given ones. Answering a question that goes back to Fellows and Langston [11] (and later was also asked by Courcelle et al. [5]), we prove that there is an algorithm that, given excluded minor characterisations for two minor ideals $\mathcal{C}$ and $\mathcal{D}$, computes an excluded minor characterisation of the minor ideal $\mathcal{C}\cup\mathcal{D}$. (Note that it is trivial to obtain such a characterisation for the minor ideal $\mathcal{C}\cap\mathcal{D}$.) Fellows and Langston [11] observed that an algorithm computing an excluded minor characterisation of $\mathcal{C}\cup\mathcal{D}$ exists if an effective bound on the tree width of the excluded minors for $\mathcal{C}\cup\mathcal{D}$ could be obtained, but left open the problem of obtaining such a bound. Using a deep result due to Robertson and Seymour [21, 19], we give such a bound. Cattell et al. [3] had earlier established the special case of our result where one of the ideals $\mathcal{C},\mathcal{D}$ is of bounded pathwidth.

We say that a graph G is an *apex graph* over a class $\mathcal{C}$ of graphs if there is a vertex $v \in V(G)$ such that the graph $G \setminus \{v\}$ obtained from G by deleting v belongs to $\mathcal{C}$. Let us denote the class of all apex graphs over $\mathcal{C}$ by $\mathcal{C}^{\text{apex}}$. Observe that if $\mathcal{C}$ is a minor ideal then so is $\mathcal{C}^{\text{apex}}$. We show that there is an algorithm that, given an excluded minor characterisation of a minor ideal $\mathcal{C}$, computes an excluded minor characterisation of $\mathcal{C}^{\text{apex}}$. Iterating the algorithm, of course we can also compute excluded minor characterisations of the classes of all graphs from which we can delete a fixed number k of vertices to obtain a graph in $\mathcal{C}$.

As a corollary, we obtain a uniform fixed-parameter tractable algorithm for the problem of deciding whether a graph has "distance k" from the class of planar graphs or from any other minor ideal. More precisely, for every minor ideal $\mathcal{C}$ we obtain an algorithm that, given a graph G and a nonnegative integer k, decides in time $f(k)\cdot n^3$ if there is a set X of at most k vertices of G such that $G \setminus X \in \mathcal{C}$. Here f is some computable function on the integers. Note that it is an immediate consequence of the graph minor theorem that for every fixed k there is a cubic time algorithm that decides if a given graph has distance at most k from planarity; the point here is the *uniformity*, which means that we have one algorithm which works for all k. As was pointed out to us by Mike Fellows, the result can also be derived by a technique from [12] called *effectivization by self reduction* (cf. [6], Sec. 7.9.1). For planar graphs, it has independently been obtained by Marx and Schlotter [17].

2 Preliminaries

In this section we briefly recall basic notions from graph minors and logic and prove some basic lemmas.

2.1 Graphs, Minors, and Minor Ideals Graphs in this paper are finite, simple, and undirected. We denote the vertex set of a graph G by $V(G)$ and its edge set by $E(G)$. We always assume that $V(G)\cap E(G) = \varnothing$. The degree of a vertex v in a graph G is denoted by $\deg_G(v)$, and the maximum degree of G is denoted by $\Delta(G)$.

A *tree decomposition* of a graph G is a pair $\mathcal{T} := (T, B)$, 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 $B^{-1}(v) = \{t \in V(T) : v \in B(t)\}$ is connected in T. Here, $G[B_t]$ denotes the subgraph of G induced by B_t. The sets B_t, for $t \in V(T)$, are called the *bags* of the decomposition $\mathcal{T}$. The *width* $\text{width}(D)$ of a tree decomposition D is defined as $\max\{|B_t|-1 : t \in V(T)\}$ and the *tree width* $\text{tw}(G)$ of a graph G is defined as $\min\{\text{width}(\mathcal{T}) : \mathcal{T}$ tree decomposition of $G\}$. Finally, a class $\mathcal{C}$ of graphs has *bounded tree width*, if there is a constant $k \in \mathbb{N}$ such that $\text{tw}(G) \leq k$ for all $G \in \mathcal{C}$. We denote the class of graphs of tree width $\leq k$ by $\mathcal{T}_k$.

For the rest of this paper we assume that the tree T underlying a tree decomposition (T, D) is sub-cubic, i.e. a tree of maximum degree at most 3. This is w.l.o.g. as any tree decomposition can easily be converted into one where the underlying tree is sub-cubic.

A graph H is a *minor* of G, in terms $H \preccurlyeq G$, if for each $v \in V(H)$ there is a non-empty connected subgraph $T_v \subseteq G$ such that for all $u, v \in V(H)$, if $u \neq v$ then $V(T_v) \cap V(T_u) = \varnothing$ and if there is an edge $e \in E(H)$ between u and v then there is an edge

$i_e \in E(G)$ with one endpoint in T_u and one endpoint in T_v. The subgraph $\bigcup_{v \in V(H)} T_v \cup \bigcup_{e \in E(H)} i_e$ of G is called a *model* of H in G and the function taking $v \in V(H)$ to T_v and $e \in E(H)$ to $i_e \in E(G)$ is called the corresponding *minor map*. By a *minimal model* of H in G we mean a model where all T_v are trees with $\deg_H(v)$ many leaves. Clearly, if $H \preccurlyeq G$, then there is a minimal model of H in G.

A *minor ideal* is a class $\mathcal{C}$ of graphs that is downward closed under the minor relation $\preccurlyeq$, that is, if $G \in \mathcal{C}$ and $H \preccurlyeq G$, then $H \in \mathcal{C}$.

DEFINITION 2.1. *Let $\mathcal{C}$ be a minor ideal. A set $\mathcal{F}$ of graphs is a set of excluded minors for $\mathcal{C}$ if*

- *$H \not\preccurlyeq G$ for all $G \in \mathcal{C}$ and $H \in \mathcal{F}$*
- *for every $G \notin \mathcal{C}$ there is a $H \in \mathcal{F}$ with $H \preccurlyeq G$*

We define an ordering $\leq_{\mathcal{C}}$ among the sets of excluded minors for $\mathcal{C}$ where $\mathcal{F}_1 := \{F_1, \ldots, F_k\} \leq_{\mathcal{C}} \mathcal{F}_2 := \{H_1, \ldots, H_l\}$ if either $f_1 := \sum_{i=1}^{k} |V(F_i)| < f_2 := \sum_{i=1}^{l} |V(H_i)|$ or $f_1 = f_2$ and $\sum_{i=1}^{k} |E(F_i)| < \sum_{i=1}^{l} |E(H_i)|$.

A set $\mathcal{F}$ of excluded minors for $\mathcal{C}$ is minimal*, if it is minimal with respect to $\leq_{\mathcal{C}}$.*

THEOREM 2.1. (GRAPH MINOR THEOREM [22]) *Every minor ideal has a finite set of excluded minors.*

The following Lemma is easy to see.

LEMMA 2.1. *Up to isomorphism, there is only one minimal set of excluded minors for any minor ideal $\mathcal{C}$.*

We omit the straightforward proof.

For a minor ideal $\mathcal{C}$ we denote the minimal set of excluded minors by $\mathcal{F}(\mathcal{C})$.

LEMMA 2.2. *Let $k > 0$, and let $H \preccurlyeq G$ with $|V(H)| = k$. Then any minimal model G' of H in G has $\operatorname{tw}(G') \leq k^2 + 1$.*

We omit the straightforward proof.

COROLLARY 2.2. *For all graphs H, G, if $H \preccurlyeq G$ then there is a subgraph G' of G with $H \preccurlyeq G'$ satisfying $\operatorname{tw}(G') \leq |V(H)|^2 + 1$.*

Observe that by the graph minor theorem and Corollary 2.2, for every minor ideal $\mathcal{C}$ there is a k such that every graph $G \notin \mathcal{C}$ has a subgraph $G' \subseteq G$ such that $G' \notin \mathcal{C}$ and $\operatorname{tw}(G') \leq k$.

DEFINITION 2.2. *Let $\mathcal{C}$ be a minor ideal. The least k such that for all $G \notin \mathcal{C}$ there is a $G' \subseteq G$ with $G' \notin \mathcal{C}$ and $\operatorname{tw}(G') \leq k$ is called the* width *of $\mathcal{C}$ and denoted by* $\operatorname{width}(\mathcal{C})$.

Although every minor ideal has bounded width, it is not clear how to compute an upper bound for the width of a class. An upper bound follows from Lemma 2.2 if we are given the excluded minors. Otherwise, it is far from clear how to compute the width and indeed much of the present paper will be spent on proving upper bounds for the width of various minor ideals.

2.2 Monadic Second-Order Logic We briefly recall monadic second-order logic (see e.g. [7]). A *signature* is a finite set $\tau := \{R_1, \ldots, R_k\}$ of relation symbols R_i, each equipped with an arity $ar(R_i)$. A τ-structure is tuple $\mathfrak{A} := (U, (R)^{\mathfrak{A}}_{R \in \tau})$ consisting of a finite universe U and for each $R \in \tau$ a relation $R^{\mathfrak{A}} \subseteq U^{ar(R)}$. With every graph G we associate a relational structure $\mathfrak{G} := (V(G) \mathbin{\dot\cup} E(G), V^{\mathfrak{G}}, E^{\mathfrak{G}}, I^{\mathfrak{G}})$, over the signature $\tau_{\text{graph}} := \{V, E, I\}$, where $V^{\mathfrak{G}} := V(G)$, $E^{\mathfrak{G}} := E(G)$, and $I^{\mathfrak{G}} := \{(v, e) : e \in E(G), v \in e \cap V(G)\}$. In the sequel, we do not distinguish notationally between a graph and the associated relational structure.

Monadic Second-Order Logic (MSO) is the extension of first-order logic by quantification over sets. E.g. formulas of MSO over τ_{graph} are built up inductively from atoms $x \in V, e \in E$, and $(x, e) \in I$ using Boolean connectives, first-order quantifiers $\exists x, \forall x$ and second-order quantifiers $\exists X, \forall X$. Here, $x, y, e, ...$ are first-order variables interpreted by single elements form $V(G) \cup E(G)$ and $X, Y, ...$ are second-order variables interpreted by sets of elements. As the universe of structures $\mathfrak{G}$ considered here contains edges and vertices, we can use set quantification over sets of edges and sets of vertices. We illustrate the definition by an example.

Example. Let H be a finite graph with $V(H) := \{v_1, \ldots, v_k\}$. Let φ_H be the MSO-sentence $\exists X_1 \ldots \exists X_k \psi$ where ψ is defined as

$$\left(\begin{array}{l} \bigwedge_{i \neq j} X_i \cap X_j = \varnothing \wedge \bigwedge_i (\varnothing \subsetneq X_i \subseteq V \wedge \\ \qquad\qquad\qquad\qquad \text{"}X_i \text{ is connected"}) \wedge \\ \bigwedge_{\{v_i, v_j\} \in E(H)} \exists x_i \in X_i \exists x_j \in X_j \exists z ((x_i, z) \in I \wedge \\ \qquad\qquad\qquad (x_j, z) \in I), \end{array} \right)$$

where "X_i is connected" can easily be expressed in MSO. The formula is true in a graph G if there is a minor map φ from H into G, where the sets X_i represent the images $\varphi(v_i)$.

Hence, for every graph G, $G \models \varphi_H$ if, and only if, $H \preccurlyeq G$. If $\mathcal{H}$ is a finite set of graphs then $\varphi_{\mathcal{H}} := \bigvee_{H \in \mathcal{H}} \varphi_H$ is true in a graph G if, and only if, at least one graph of $\mathcal{H}$ is a minor of G. Hence, $\neg \varphi_{\mathcal{H}}$ defines the minor ideal $\mathcal{F}(\mathcal{H})$.

THEOREM 2.3. (SEESE [23]) *For each $k \in \mathbb{N}$, it is decidable for a given* MSO*-formula whether it is true in a graph of tree width at most k.*

2.3 Extending Structures by Tree Decompositions To define certain properties of graphs in MSO, it is sometimes more convenient if the MSO-formulas can explicitly refer to tree decompositions of graphs. We will therefore equip graphs with a tree-decomposition, i.e. we consider structures consisting of a graph and a tree-decomposition of the graph. Recall that we only consider tree-decompositions in this paper where the underlying graph is sub-cubic.

DEFINITION 2.3. (TREE-DEC EXPANSIONS) *Let G be a graph and $\mathcal{T} := (T, B)$ be a tree decomposition of G. Let $\tau_{\text{ex}} := \{V, E, I, T, E_T, I_T, B\}$, where V, E, T, E_T are unary relation symbols and I, I_T, B are binary.*

The tree-dec expansion $\text{TreeExp}(G, \mathcal{T})$ *of G and $\mathcal{T}$ is the τ_{ex}-structure*

$$(V(G) \dot\cup E(G) \dot\cup V(T) \dot\cup E(T), \quad V^{\mathfrak{G}}, E^{\mathfrak{G}}, I^{\mathfrak{G}}, T^{\mathfrak{G}}, E_T^{\mathfrak{G}}, I_T^{\mathfrak{G}}, B^{\mathfrak{G}}),$$

where $V^{\mathfrak{G}}, E^{\mathfrak{G}}, I^{\mathfrak{G}}$ are as above and $T^{\mathfrak{G}} := V(T)$, $E_T^{\mathfrak{G}} := E(T)$, $I_T^{\mathfrak{G}} := \{(t, e) : e \in E(T), t \in e \cap V(T)\}$ and $B^{\mathfrak{G}} := \{(t, u) : t \in V(T), u \in B_t \cap V(G)\}$.

Intuitively, we think of tree-dec expansions $\mathfrak{G}$ as of two-sorted structures consisting of the graph G and the tree decomposition $\mathcal{T}$. The proof of the following lemma is straightforward.

LEMMA 2.3. *(i) If G is a graph and $\mathcal{T}$ a tree decomposition of G of width k, then the tree width of* $\text{TreeExp}(G, \mathcal{T})$ *is at most $k + 2$.*

(ii) There is an MSO*-sentence φ_{ex} so that for any τ_{ex}-structure $\mathfrak{G}$, $\mathfrak{G} \models \varphi_{\text{ex}}$ if, and only if, $\mathfrak{G}$ is a Tree-Dec Expansion of a graph G.*

We denote the class of structures $\text{TreeExp}(G, \mathcal{T})$, for $G \in \mathcal{T}_k$ and $\mathcal{T}$ a tree decomposition of G of width at most k, by $\text{TreeExp}(\mathcal{T}_k)$. It follows from Lemma 2.3 that $\text{TreeExp}(\mathcal{T}_k)$ is a class of structures of tree width at most $k + 2$.

COROLLARY 2.4. *For all $k \in \mathbb{N}$, it is decidable whether a given* MSO*-formula is satisfied in some $\mathfrak{G} \in$* $\text{TreeExp}(\mathcal{T}_k)$.

2.4 Dynamic Programming and MSO-definability MSO is a powerful logic on graphs and many interesting properties of graphs can be expressed naturally in this logic. In fact, for various problems, their mathematical formalisations only use set and individual quantification and therefore *are* already MSO-definitions. However, sometimes algorithms for a certain problem on graph classes of bounded tree width are already known and we don't want to prove an equivalent MSO-definition.

In this section we therefore develop a strong relationship between dynamic programming algorithms as they are typically employed on graph classes of bounded tree width and MSO-definability.

A *typical linear time dynamic programming algorithm* on classes of graphs of bounded tree width, *tlp-algorithm* for short, is an algorithm that, given a graph G and a tree decomposition $\mathcal{T}$ of G of width at most k as input, computes a data record for each node of $\mathcal{T}$, starting at the leaves and proceeding bottom up, so that

- whether or not the algorithm accepts G only depends on the record computed for the root of $\mathcal{T}$
- there are only constantly many possible data records
- the record computed for a leaf t of $\mathcal{T}$ only depends on the isomorphism type of B_t
- the record computed for an inner node t can be computed in constant time from the isomorphism type of B_t and the records computed for the children.

Many algorithms developed for graphs of small tree width are actually of this form, e.g. standard algorithms for 3-colourability, dominating set, vertex cover and many more. For instance, for the 3-colourability problem, the data record computed for a node $t \in V(T)$ of a tree decomposition (T, B) of a graph G consists of all possible 3-colourings of B_t that can be extended to a 3-colouring of the subgraph of G induced by the vertices contained in bags in the subtree of T rooted at t. Another example is the algorithm given by Bodlaender and Kloks in [1] which, given a graph G and a tree decomposition of G of (possibly non-optimal) width w, and a number k decides whether G has tree width at most k in time exponential in w and linear in the size of G.

LEMMA 2.4. *Let A be a tlp-algorithm. There is an* MSO *sentence φ_A such that given a structure* $\text{TreeExp}(G, \mathcal{T})$ *of a graph G and a tree decomposition $\mathcal{T}$ of G, $\mathfrak{G}$ satisfies φ_A if, and only if, A accepts the pair $(G, \mathcal{T})$.*

Proof. Let $C := \{C_1, \ldots, C_l\}$ be the constant set of possible data records used by A. The sentence φ_A is defined as $\varphi_A := \exists C_1 \ldots \exists C_l \psi_A$, where ψ_A is a first-order sentence stating that every node of $\mathcal{T}$ is contained in exactly one set C_i and that the colours C_i assigned to the nodes of $\mathcal{T}$ are consistent with the data records computed by A on $\mathcal{T}$. The latter can be defined in FO, as the data record, and hence the colour of a node t, only depends on the isomorphism type of B_t and the data records/colours of the successors. Hence, this is finite information that can be coded directly in the first-order formula. □

Recall from above that the Bodlaender, Kloks algorithm from [1] is a tlp-algorithm. Hence, we immediately obtain the following corollary.

COROLLARY 2.5. *For every $k \in \mathbb{N}$ there is an* MSO-*sentence φ_k such that a structure* $\mathrm{TreeExp}(G, \mathcal{T}) \in \mathrm{TreeExp}(\mathcal{T}_l)$*, for some $l \geq k$, is a model of φ_k if, and only if,* $\mathrm{tw}(G) = k$.

The following strengthening of Seese's Theorem 2.3 will be useful later on.

THEOREM 2.6. *For every k it is decidable whether a given* MSO*-formula is satisfied by a graph of tree width exactly k.*

Proof. Let φ be an MSO-sentence and $k \in \mathbb{N}$. To decide if φ has a model of tree width exactly k we test whether $\varphi \wedge \varphi_{\mathrm{ex}} \wedge \varphi_k$ is satisfiable in $\mathrm{TreeExp}(\mathcal{T}_k)$. Here, φ_{ex} is the MSO-sentence defined in Lemma 2.3 and φ_k is the sentence defined in Corollary 2.5. By Corollary 2.4, this is decidable. □

We remark that by the proof of Courcelle's theorem (see [4]), every MSO-definable problem can be solved by a tlp-algorithm on graph classes of bounded tree width.

COROLLARY 2.7. *Let $\mathcal{C}$ be a class of bounded tree width. A problem can be solved by a tlp-algorithm on $\mathcal{C}$ if, and only if, it is* MSO*-definable on the class* $\mathrm{TreeExp}(\mathcal{C})$.

3 Computing Excluded Minors

In this section we develop a general machinery for computing minimal sets of excluded minors that will be applied to various minor ideals in Sections 4 and 5.

DEFINITION 3.1. *A class $\mathcal{C}$ of graphs is* layerwise MSO-definable*, if for every $k \in \mathbb{N}$ we can compute an* MSO-*formula φ_k defining $\mathcal{C} \cap \mathcal{T}_k$ in* $\mathrm{TreeExp}(\mathcal{T}_k)$.

We will show next that if $\mathcal{C}$ is a minor ideal that is layerwise MSO-definable and for which we are given an upper bound on its width, then we can compute the minimal set of excluded minors for $\mathcal{C}$. We will state this result in two equivalent forms, a logical form that we actually prove (cf. Lemma 3.1) and an equivalent algorithmic form.

LEMMA 3.1. *The minimal set of excluded minors is computable for every layerwise* MSO*-definable class of graphs, provided we are given an upper bound on its width.*

Formally, there is an algorithm that, given $w \in \mathbb{N}$ and (an algorithm computing) a computable function $g : \mathbb{N} \to \mathrm{MSO}$ so that there is a minor ideal $\mathcal{C}$ with

- *for every $k \in \mathbb{N}$, $\varphi_k := g(k)$ defines $\mathcal{C} \cap \mathcal{T}_k$ in* $\mathrm{TreeExp}(\mathcal{T}_k)$ *and*
- *the width of $\mathcal{C}$ is at most w,*

then the algorithm computes $\mathcal{F}(\mathcal{C})$.

Proof. We first show that for any given set $\mathcal{F} := \{F_1, \ldots, F_m\}$ we can decide if $\mathcal{F}$ is a set of excluded minors for $\mathcal{C}$. For this, we show that the following two statements are decidable.

(1) Is $(G \in \mathcal{C} \wedge \exists\, i : F_i \preccurlyeq G)$ unsatisfiable?

(2) Is $(G \notin \mathcal{C} \wedge \forall\, i : F_i \not\preccurlyeq G)$ unsatisfiable?

Towards (1), it suffices to check whether $F_i \in \mathcal{C}$ for some $1 \leq i \leq m$. However, as there are no assumptions on the membership problems of $\mathcal{C}$, we give a general method that works for all classes $\mathcal{C}$ satisfying the prerequisites of the lemma. Suppose there is a $G \in \mathcal{C}$ with a minor $F_i \preccurlyeq G$. By Lemma 2.2, there is $G' \subseteq G$ of tree width at most $|V(F_i)|^2 + 1$ with $F_i \preccurlyeq G'$. As $\mathcal{C}$ is closed under taking subgraphs, $G' \in \mathcal{C}$. Hence, $(G \in \mathcal{C} \wedge \exists i : F_i \preccurlyeq G)$ is satisfiable if, and only if, it is satisfiable by a graph of tree width at most k, where $k := \max\{|V(F_i)|^2 + 1 : F_i \in \mathcal{F}\}$.

By assumption, the class $\mathcal{C} \cap \mathcal{T}_k$ is MSO-definable in $\mathrm{TreeExp}(\mathcal{T}_k)$ by $\varphi_{\mathcal{C}} := g(k)$. Let $\varphi_{\mathcal{F}}$ be the MSO-sentence from Example 2.2 saying that every model of $\varphi_{\mathcal{F}}$ has at least one graph $F \in \mathcal{F}$ as a minor. Then there is a $G \in \mathcal{C}$ with $F_i \preccurlyeq G$, for some $1 \leq i \leq m$ if, and only if, the sentence $\varphi_{\mathcal{C}} \wedge \varphi_F$ is satisfiable in $\mathrm{TreeExp}(\mathcal{T}_k)$ – which is decidable by Corollary 2.4.

Towards (2), suppose there is a $G \notin \mathcal{C}$ and $F_i \not\preccurlyeq G$ for all $1 \leq i \leq m$. As the width of $\mathcal{C}$ is bounded by w, G has a subgraph G' of tree width $\mathrm{tw}(G') \leq w$ so that $G \notin \mathcal{C}$. Clearly, $F_i \not\preccurlyeq G'$ for all $1 \leq i \leq m$.

Let $\varphi_w := g(w)$ be the MSO-sentence that defines $\mathcal{C} \cap \mathcal{T}_w$ in $\mathrm{TreeExp}(\mathcal{T}_w)$. Then there is a graph $G \notin \mathcal{C}$ of tree width at most w with no minor $F_i \in \mathcal{F}$ if, and only if, $\neg\varphi_w \wedge \neg\varphi_{\mathcal{F}}$ is satisfiable in $\mathrm{TreeExp}(\mathcal{T}_w)$. This is decidable by Corollary 2.4.

Hence, we can decide for each finite set $\mathcal{F}$ of graphs if it is a set of excluded minors for $\mathcal{C}$. Enumerating all finite sets $\mathcal{F}$ of graphs in a way respecting the ordering $\leq_{\mathcal{C}}$ of Definition 2.1, the first set $\mathcal{F}$ of excluded minors for $\mathcal{C}$ will be $\mathcal{F}(\mathcal{C})$. □

For the algorithmic variant of the lemma, we say that a class $\mathcal{C}$ of graphs is *layerwise decidable*, if for every $k \in \mathbb{N}$ we can compute a tlp-algorithm that decides for any graph G of tree width at most k if $G \in \mathcal{C}$.

COROLLARY 3.1. *The set $\mathcal{F}(\mathcal{C})$ of excluded minors is computable for every layerwise decidable class $\mathcal{C}$ of graphs given an upper bound w of its width.*

Often we can use a slightly simpler form of the previous lemma and corollary.

COROLLARY 3.2. *There is an algorithm that, given an* MSO*-formula φ and $w \in \mathbb{N}$, so that φ defines a minor ideal $\mathcal{C}$ of width at most w, computes the set $\mathcal{F}(\mathcal{C})$.*

Proof. This follows from Lemma 3.1 by setting $g(k) := \varphi$ for all k. □

4 Computing Excluded Minors for Specific Classes of Graphs

In this section we present some immediate applications of the machinery developed in Section 3 and show that the set of excluded minors are computable for various natural families of minor ideals.

4.1 Bounded Tree Width and Branch-Width

As a first and simple example we show that, given $k \in \mathbb{N}$, the set of excluded minors is computable for the class of graphs of tree width at most k and the class of graphs of branch-width at most k. The following theorem is also an immediate consequence of a result due to Lagergren [16] giving an upper bound on the size of the excluded minors for the class $\mathcal{T}_k$.

THEOREM 4.1. ([16]) *There is an algorithm that, given $k > 1$, computes $\mathcal{F}(\mathcal{T}_k)$.*

Proof. Let $k > 1$. Clearly, $\mathcal{T}_k$ has width $k + 1$, as every graph G of tree width $\geq k + 1$ has a subgraph of tree width $k + 1$. Furthermore, by Corollary 2.5, for every $l > 1$ we can compute an MSO-sentence φ_l defining $\mathcal{T}_k \cap \mathcal{T}_l$ in TreeExp$(\mathcal{T}_l)$. Hence, the theorem follows from Lemma 3.1. □

Another type of graph decompositions related to tree width is the notion of *branch-decompositions* and the associated width measure *branch-width.* Whereas a tree decomposition is a decomposition of the vertex set of a graph, a branch-decompositions decomposes its edge set. We refrain from giving details here and refer the reader to [20]. Let $\mathcal{B}_k$ denote the class of graphs of branch-width at most k.

The branch-width and tree width of a graph are within a constant factor but clearly the sets of excluded minors for $\mathcal{T}_k$ and $\mathcal{B}_k$ differ. In [2], Bodlaender and Thilikos give for any fixed $k \in \mathbb{N}$ a linear-time algorithm for deciding whether a graph has branch-width exactly k. It follows from their result that there is an tlp-algorithm to decide the branch width of a graph given a tree decomposition of potentially non optimal width. Hence, an analogous reasoning as for Theorem 4.1 establishes the following theorem. The theorem has already been stated in [2], and it also follows easily from an upper bound on the size of the excluded minors for matroids of bounded branch width [13].

THEOREM 4.2. ([2, 13]) *There is an algorithm that, given $k > 1$, computes $\mathcal{F}(\mathcal{B}_k)$.*

4.2 Bounded Genus

The *(orientable) genus* of a graph G is the minimum genus of an orientable surface S such that G can be embedded into S. (We only consider orientable surfaces here, but the same results hold for nonorientable surfaces as well.) Let $\mathcal{G}_k$ be the class of all graphs of genus at most k.

It is not hard to prove that for every k there is a sentence φ_k of monadic second-order logic that defines the class $\mathcal{G}_k$: We construct φ_k inductively. φ_0 just says that a graph does not contain K_5 or $K_{3,3}$ as a minor. In the inductive step, we use the fact that a graph has genus $k + 1$ if any only if we can "cut the graph along some cycle" such that the resulting graph has genus k. If we choose the shortest such cycle, we can further assume that this cycle is chordless. In the MSO-sentence φ_{k+1}, we existentially guess such a cycle and also which edges incident with the cycle go on which side of the cut, and then use φ_k to express that the graph obtained by cutting along this cycle is in $\mathcal{G}_k$.

A graph G is *critical* for $\mathcal{G}_k$ if $G \notin \mathcal{G}_k$, but $G' \in \mathcal{G}_k$ for every proper subgraph $G' \subset G$. Thomassen [25] proved that there is a computable function g such that every graph G that is critical for $\mathcal{G}_k$ has tree width at most $g(k)$. Now the following theorem is an immediate consequence of Lemma 3.1. It was observed in [18] that it can also be derived from Thomassen's result by the techniques of [11]. Furthermore, it follows from an unpublished result due to Seymour [24] giving an upper bound on the excluded minors for graphs of bounded genus.

THEOREM 4.3. ([11, 18, 24, 25]) *There is an algorithm that, given $k > 1$, computes $\mathcal{F}(\mathcal{G}_k)$.*

5 Constructions

In this section we give the main new applications of our method for computing excluded minors. Whereas in the previous section we showed that excluded minors are computable for specific classes of graphs, we now study ways to compute the excluded minors for derived classes of graphs given the excluded minors for the base classes.

5.1 Apices For any given class $\mathcal{C}$ of graphs we can construct the class of graphs G such that there is a vertex $v \in V(G)$ for which $G \setminus v \in \mathcal{C}$. We call this vertex an *apex* of G with respect to $\mathcal{C}$. We denote the class of graphs having an apex with respect to $\mathcal{C}$ by $\mathcal{C}^{\text{apex}}$. The aim of this section is to show the following theorem.

THEOREM 5.1. *If $\mathcal{C}$ is a minor ideal whose set of excluded minors is given, then we can compute the excluded minors of $\mathcal{C}^{\text{apex}}$.*

By iterating this construction, we can compute for any k the set of excluded minors for the class of graphs G for which there are k vertices $v_1, \dots, v_k$ such that $G \setminus \{v_1, \dots, v_k\} \in \mathcal{C}$.

We now prove Theorem 5.1. Let $\mathcal{C}$ be a minor ideal. As before, we aim at applying Lemma 3.1. Using Example 2.2, it is easily seen that for each $k \in \mathbb{N}$, $\mathcal{C}^{\text{apex}} \cap \mathcal{T}_k$ is MSO-definable over TreeExp($\mathcal{T}_k$). Hence, it remains to show that $\mathcal{C}^{\text{apex}}$ has effectively bounded width. We first need some preparations.

5.1.1 Walls, Layouts, and Linkages An *elementary wall of height $h \geq 1$* is a graph defined as in Figure 1(a). A *wall of height h* is a subdivision of an elementary wall of height h. The *perimeter* of a wall is the boundary cycle. A wall *in* a graph G is a wall W 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.

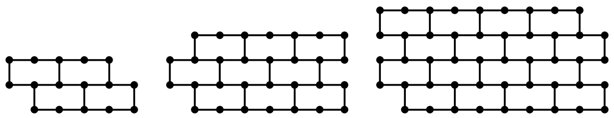

Figure 1: Elementary walls of height 2–4

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

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

We let $\overline{C}$ be the graph obtained from C by adding new vertices $d_1, \dots, d_m$ and, for $1 \leq i \leq m$, edges between d_i to the vertices in $\partial^G D_i$ and edges between all vertices in $\partial^G D_i$. Hence, for each $i \in \{1, \dots, m\}$ 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 $D_1, \dots, D_m$ its *extensions.* The layout $(C, D_1, \dots, D_m)$ 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 W (as shown in Figure 1), because the wall W has a unique embedding into the sphere. We call the wall W *flat* (in G) if the compass of W has a flat layout.

The following lemma is (essentially) Lemma (9.8) of [21]. Concerning the uniformity, see the remarks at the end of [21] (on page 109).

TRINITY LEMMA (ROBERTSON AND SEYMOUR [21]). *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 W of height h in $G \setminus X$, and a flat layout $(C, D_1, \dots, D_m)$ of the compass of W 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 $g(k, h) \cdot |V(G)|^2$.

A *linkage* L in a graph G is a subgraph whose components are paths. For a set $Z \subseteq V(G)$, the *effect* of L on Z is defined as the partition of $Z \cap V(L)$ where two vertices belong to the same class if they belong to the same component of L.

5.1.2 Models in Critical Graphs

A graph G is called *critical* for $\mathcal{C}^{\text{apex}}$, if $G \notin \mathcal{C}^{\text{apex}}$ but $G' \in \mathcal{C}^{\text{apex}}$ for every proper subgraph $G' \subset G$. Let $\mathcal{F}(\mathcal{C}) := \{H_1, \ldots, H_m\}$. To show that $\mathcal{C}^{\text{apex}}$ has effectively bounded width we prove that the tree width of its critical graphs is bounded by $f(H_1, \ldots, H_m)$ for a computable function f from finite sets of graphs to natural numbers.

Let G be a critical graph for $\mathcal{C}^{\text{apex}}$ and let $c := \max\{|H_i| : 1 \leq i \leq m\}$. We set $k := c + 2$ and let h be "large". We will explain how to choose h as we go along. For the following arguments, assume that h is "large enough".

By the Trinity Lemma, there is a computable function $f(k, h)$ such that G either has (1) tree width at most $f(k, h)$, or (2) a K_k-minor, or (3) a subset $X \subseteq V(G)$ with $|X| < \binom{k}{2}$, a wall W of height h in $G \setminus X$, and a flat layout $(C, D_1, \ldots, D_m)$ of the compass of W in $G \setminus X$ such that the tree width of each of the extensions $D_1, \ldots, D_m$ is at most $f(k, h)$.

Clearly, Case 2 is impossible, as then G would have a proper subgraph containing a K_{c+1} minor. We show next that Case 3 is impossible as well and hence the tree width of G is bounded by $f(k, h)$.

Suppose Case 3 applies. Let P be the perimeter of the wall W. As $G \notin \mathcal{C}^{\text{apex}}$, there is a model of some H_i in G. Let H be a minimal model of H_i in G. We use H both for the minor map and the model of H_i in G. A vertex $v \in V(H) \cap C$ is called *important*, if it is in $C \setminus P$ and

1. v has degree at least 3 in H, or
2. $v \in H(e)$ for some $e \in E(H_i)$, or
3. v has a neighbour in $X \cap V(H)$, or
4. there is a path in H from v to a vertex $u \in V(H)$ in an extension D, so that the internal vertices of the path are all contained in D and the degree of u in H is at least 3, or $u \in H(e)$ for some $e \in E(H_i)$, or there is an edge in H from u to a vertex $x \in X$.

From the minimality of the model H of H_i in G and the fact that each $x \in X \cap V(H)$ can have at most $|H_i|$ neighbours in $V(H)$, we can infer that the number of important vertices is bounded by $\mathcal{O}(|H_i|)$. Hence, as h is large, by the pigeonhole principle there is a subwall W' of W whose compass contains no important vertex (and which is still large enough). It follows that the intersection of H with the compass of W' is a set L_H of paths whose endpoints are outside of W'. Now we replace the compass of W' in G by its planar core $\overline{C}'$, i.e. we contract each extension D_i to a new single vertex d_i and add edges between any pair of vertices in ∂D_i. Let G' be the new graph. As $|\partial D_i| \leq 3$ for all D_i, each extension can be traversed by at most one path in L_H. Hence, L_H induces a unique linkage L in G' with the same effect on the set of important vertices. L is a linkage between important vertices and hence the number of paths in L is bounded by $\mathcal{O}(|H|)$. We show next that we can find another linkage L' with the same effect on the set of important vertices that only uses the outermost $g(|L|)$ *shells* of the wall W' (cf. Figure 2). The next lemma follows immediately from 3.1 in [19] letting $\Gamma := W$, $K := G \setminus (W - P)$, and choosing Z as the set of endpoints of L, as any vertex of W not in the outer $g(|L|)$ circuits is $g(|L|)$-insulated from K.

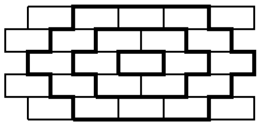

Figure 2: The shells of a wall of height 5

Lemma 5.1. *There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that if $W \subseteq G$ is a wall that is flat in a graph G and L is a linkage in G whose endpoints are disjoint from W, then there is a linkage L' in G with the same effect on the endpoints as L such that $(G \setminus W) \cap L' \subseteq L$ and every vertex $v \in V(W) \cap V(L')$ is contained in the outer $g(|L|)$ circuits of W.*

The linkage L' also induces a model H' of H_i in G that has an empty intersection with the inner shells of W'. Hence, there is a "large" subwall W'' of W' that has an empty intersection with L' (and H').

For any graph G we let the *folio* of G be the class of all minors of G. For $\delta \geq 0$, we say that a graph H has *detail* at most δ if $|E(H)| \leq \delta$ and $|V(H)| \leq \delta$ and define the *δ-folio* of G to be the class of all minors of G of detail at most δ.

In the following, we let $\delta := \max\{|V(H_i)|, |E(H_i)|\}$. The following lemma is an immediate consequence of (10.3) and the algorithm in (10.4) in [21].

Lemma 5.2. *There is a computable function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ so that if the height of W'' is at least $f(|X|, \delta)$, then there is a subwall W''' of W'' of which both middle vertices are irrelevant for the δ-folio of G.*

The subwall W''' guaranteed by the previous lemma is called *homogeneous* in [21]. We remark that a homogeneous wall in a graph G remains homogeneous if we delete a vertex w_0 from G that is not contained in the compass of the wall. This is used below.

Let v_0 be one of the middle vertices of W''' and let $G' := G \setminus v_0$. As G is a critical graph, $G' \in \mathcal{C}^{\text{apex}}$. Hence, there is a vertex w_0 in $V(G')$ so that $G' \setminus w_0 \in \mathcal{C}$. Recall that H' is a model of H_i in G that does not contain v_0. Hence, H' is a model of H_i in G' and therefore $w_0 \in V(H')$. Thus, w_0 is not in the compass of W'''.

Let $G'' := G \setminus w_0$. As $G \notin \mathcal{C}^{\text{apex}}$, $G'' \notin \mathcal{C}$. Therefore, there is a model of some $H_i \in \{H_1, \dots, H_m\}$ in G''. As w_0 is not contained in the compass of W''' in G, W''' is still a flat homogeneous subwall of G'' and hence the two middle vertices of W''' are still irrelevant for the δ-folio of G''. It follows, that $G'' \setminus v_0$ still contains a model of H_i and therefore $G'' \notin \mathcal{C}$. But this is a contradiction, as $G'' \setminus v_0 = G' \setminus w_0$.

This shows that if initially we choose h large enough so that after applying the various constructions above the wall W'' can be guaranteed to be large enough to apply Lemma 5.2 for our value of δ, then Case 3 of the Trinity Lemma is impossible as well. Hence, the tree width of G is bounded by $f(k, h)$, where f is the function from the Trinity Lemma. This shows that critical graphs have bounded tree width and concludes the proof of Theorem 5.1.

5.2 Excluded Minors for Unions of Classes The last application of our method we give is to show that we can compute the set of excluded minors for the union of two minor ideals, provided that we are given their excluded minors.

THEOREM 5.2. *There is an algorithm that, given two classes $\mathcal{C}_1, \mathcal{C}_2$ of finite graphs represented by their sets of excluded minors $\mathcal{F}(\mathcal{C}_1) := \{H_1, \dots, H_l\}$ and $\mathcal{F}(\mathcal{C}_2) := \{I_1, \dots, I_r\}$, computes the set of excluded minors for the union $\mathcal{C} := \mathcal{C}_1 \cup \mathcal{C}_2$.*

Proof. We aim at applying Corollary 3.2. Towards this aim, we first establish an effective bound on the width of $\mathcal{C}$. Let $G \notin \mathcal{C}$. Hence, there are $H_i \in \{H_1, \dots, H_l\}$ and $I_j \in \{I_1, \dots, I_r\}$ such that $H_i \preccurlyeq G$ and $I_j \preccurlyeq G$. Let H and I be models of H_i and I_j in G such that $|I \cup H|$ is as small as possible. We use H both for the model mapping and the image $H \subseteq G$ and likewise for I. It will always be clear what is meant. Let $G' := H \cup I \subseteq G$. Further, let $k := \max\{|H_i|, |I_j|\}$ and choose h "large enough", where, as in the previous section, the meaning of large enough will become clear in the course of the proof.

By the Trinity Lemma, G' either has a) tree width bounded by some $f(k, h)$ for some computable function f, or b) a K_k minor, or c) there is a subset $X \subseteq V(G)$ with $|X| < \binom{k}{2}$, a wall W of height h in $G \setminus X$, and a flat layout of the compass of W in $G \setminus X$.

Suppose Case c) applies. Note that every vertex in G' has degree at most $2k$. If, initially, we choose h large enough, we can argue as in the previous section and use Lemma 5.2 to find a large homogenous subwall W' of W whose compass is disjoint from X. Again by Lemma 5.2, the middle vertex v_0 of W' is irrelevant for the k-folio of G' and we can remove it to obtain a graph $G'' := G \setminus v_0$ that still has I_j and H_i as a minor, contradicting the minimality of H and I. Hence, Case 3) is impossible.

For Case b), Lemma 2.2 implies that G', and hence G, has a subgraph of tree width at most k^2+1 containing a K_k minor and therefore also an I_j and an H_i minor.

Finally, in Case a) the tree width of G' is bounded by a computable function in h and k. In either case, we have found a uniform upper bound for the tree width of a subgraph of G containing an I_j and an H_i minor. This shows that $\mathcal{C}$ has effectively bounded with. To conclude the proof we need to show that $\mathcal{C}$ is MSO-definable. But this follows immediately from Example 2.2. $\square$

6 Conclusion and Open Problems

We have introduced a general method for computing excluded minors for minor ideals that is based on MSO-definability on graphs of small tree width and the notion of the width of a minor ideal. As straightforward applications of this method we showed that excluded minors for several natural minor ideals are computable, in particular for the classes of graphs of tree width at most k, branch-width at most k, or of bounded genus.

We have also demonstrated the usefullness of our method by giving far more advanced examples. In particular, we showed that excluded minors can be computed for the class of apex graphs over a minor ideal whose excluded minors we know already, or for the union of two minor ideals, provided we are given the excluded minors of the base classes. The latter answers an open question from [5].

We conclude the paper by stating some open problems. In [8], Eppstein introduced minor ideals with the *diameter tree width property*, which is now more commonly known as bounded local tree width. We briefly recall the definition. Let G be a graph. For every vertex $v \in V(G)$ and every $r \geq 1$ we define the *r-neighbourhood $N_r^G(v)$ of v* as the set of vertices of distance at most r from v. The *local tree width* of G is the function $\mathrm{ltw}(G, \cdot)$ defined by $\mathrm{ltw}(G, r) = \max\left\{ \mathrm{tw}\left(G[N_r^G(v)]\right) : v \in V(G)\right\}$. For all nonnegative integers λ, μ we let $\mathcal{L}(\lambda) = \left\{G : \forall H \preccurlyeq G\ \forall r \geq 0 : \mathrm{ltw}(H, r) \leq \lambda \cdot r\right\}$, and $\mathcal{L}(\lambda, \mu) = \left\{G : \exists X \subseteq V(G) \text{ s. th. } |X| \leq \mu \text{ and } G \setminus X \in \mathcal{L}(\lambda)\right\}$.

Minor ideals of bounded local tree width have many interesting algorithmic properties. Another reason these classes are interesting is the fact that for every $k \in \mathbb{N}$ there are $\lambda, \mu \in \mathbb{N}$ such that every graph G with $K_k \not\preccurlyeq G$ has a tree decomposition over $\mathcal{L}(\lambda, \mu)$ (see [14]). We do not currently know how to compute upper bounds w_λ for the width of the classes $\mathcal{L}(\lambda)$. It is, however, easily seen that they are layerwise MSO definable. Hence, we obtain the following consequence.

Corollary 6.1. *For each $\lambda \in \mathbb{N}$, an upper bound on the width of $\mathcal{L}(\lambda)$ is computable if, and only if, the set $\mathcal{F}(\mathcal{L}(\lambda))$ is computable.*

Observe that, by the results of Section 5.1, we can compute $\mathcal{F}\big(\mathcal{L}(\lambda, \mu)\big)$ given $\mathcal{F}\big(\mathcal{L}(\lambda)\big)$.

Open Problem. Is there a computable function $f : \mathbb{N} \to \mathbb{N}$ such that $f(\lambda)$ is an upper bound for the width of $\mathcal{L}(\lambda)$, for all $\lambda \in \mathbb{N}$?

Another natural open problem concerns the class of graphs having a tree decomposition over a fixed minor ideal $\mathcal{C}$. Again the hard part is not to show that this class is layerwise MSO-definable but to show that we can uniformly bound its width in terms of the excluded minors for $\mathcal{C}$.

Open Problem. Let $\mathcal{D}$ be the class of graphs having a tree decomposition over a fixed minor ideal $\mathcal{C}$. Is there an algorithm computing an upper bound for the width of $\mathcal{D}$ given the set $\mathcal{F}(\mathcal{C})$ as input?

References

[1] H.L. Bodlaender and T. Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21:358–402, 1996.

[2] H.L. Bodlaender and D.M. Thilikos. Constructive linear time algorithms for branchwidth. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proceedings of the 24th International Colloquium on Automata, Languages and Programming*, volume 1256 of *Lecture Notes in Computer Science*, pages 627–637. Springer-Verlag, 1997.

[3] K. Cattell, M.J. Dinneen, R.G. Downey, M.R. Fellows, and M.A. Langston. On computing graph minor obstruction sets. *Theoretical Computer Science*, 233:107–127, 2000.

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

[5] B. Courcelle, R.G. Downey, and M.R. Fellows. A note on the computability of graph minor obstruction sets for monadic second order ideals. *Journal of Universal Computer Science*, 3:1194–1198, 1997.

[6] R.G. Downey and M.R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.

[7] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer, 2nd edition, 1999.

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

[9] M.R. Fellows and M.A. Langston. Nonconstructive advances in polynomial-time complexity. *Information Processing Letters*, 26(3):155–162, 1987.

[10] M.R. Fellows and M.A. Langston. Nonconstructive tools for proving polynomial-time decidability. *Journal of the ACM*, 35, 1988.

[11] M.R. Fellows and M.A. Langston. An analogue of the Myhill-Nerode theorem and its use in computingfinite-basis characterizations. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 520–525, 1989.

[12] M.R. Fellows and M.A. Langston. On search, decision and the efficiency of polynomial-time algorithms (extended abstract). In *Proceedings of the 21st ACM Symposium on Theory of Computing*, pages 501–512, 1989.

[13] J.F. Geelen, A.M.H. Gerards, N. Robertson, and G. Whittle. On the excluded minors for the matroids of branch-width k. *Journal of Combinatorial Theory, Series B*, 88:261–265, 2003.

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

[15] M. Dinneen K.Cattell and M. Fellows. Forbidden minors to graphs with small feedback sets. *Discrete Mathematics*, 230:215–252, 2001.

[16] J. Lagergren. Upper bounds on the size of obstructions and intertwines. *Journal of Combinatorial Theory, Series B*, 73:7–40, 1998.

[17] D. Marx and I. Schlotter. Obtaining a planar graph by vertex deletion. In *Proceedings of the 33rd Workshop on Graph-Theoretic Concepts in Computer Science*, 2007.

[18] B. Mohar and C. Thomassen. *Graphs on Surfaces.* Johns Hopkins University Press, 2001.

[19] N. Robertson and P.D. Seymour. Graph minors XXII. Irrelevant vertices in linkage problems. To appear.

[20] N. Robertson and P.D. Seymour. Graph minors X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52:153–190, 1991.

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

[22] N. Robertson and P.D. Seymour. Graph minors XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92:325–357, 2004.

[23] D. Seese. The structure of models of decidable monadic theories of graphs. *Annals of Pure and Applied Logic*, 53:169–195, 1991.

[24] P. Seymour. A bound on the excluded minors for a surface, 1995. Unpublished manuscript.

[25] C. Thomassen. A simpler proof of the excluded minor theorem for higher surfaces. *Journal of Combinatorial Theory, Series B*, 70:306–311, 1997.