Definable Tree Decompositions

Martin Grohe
Humboldt-Universität zu Berlin
grohe@informatik.hu-berlin.de

Abstract

We introduce a notion of definable tree decompositions *of graphs. Actually, a definable tree decomposition of a graph is not just a tree decomposition, but a more complicated structure that represents many different tree decompositions of the graph. It is definable in the graph by a tuple of formulas of some logic. In this paper, only study tree decomposition definable in fixed-point logic. We say that a definable tree decomposition is* over *a class of graphs if the pieces of the decomposition are in this class.*

We prove two general theorems lifting definability results from the pieces of a tree decomposition of a graph to the whole graph. Besides unifying earlier work on fixed-point definability and descriptive complexity theory on planar graphs and graphs of bounded tree width, these general results can be used to prove that the class of all graphs without a K_5-minor is definable in fixed-point logic and that fixed-point logic with counting captures polynomial time on this class.

1. Introduction

The question of whether there is a logic that captures polynomial time is the central open problem in descriptive complexity theory. It was first asked by Chandra and Harel [4] in the context of database theory, and later in a slightly different form by Gurevich [13] in the context of finite model theory. We say that a logic L *captures polynomial time (on a class $\mathscr{C}$ of structures)* if the polynomial time decidable properties of structures (in the class $\mathscr{C}$) are precisely those definable in L. Actually, the exact definition of a logic capturing PTIME is a bit more subtle; I refer the reader to the short survey [11] in these proceedings for the definition and more background on the quest for a logic capturing PTIME.

A natural logic that was considered a candidate for a logic capturing PTIME for a while is *inflationary fixed-point logic with counting*, IFP+C. Although Cai, Fürer, and Immerman [3] proved that IFP+C does not capture PTIME on all structures, in my LICS-paper 10 years ago [9] I showed that IFP+C captures PTIME on the class of all planar graphs. With Julian Mariño [12], we also proved the corresponding result for classes of structures of bounded tree width. The key step in the proofs of these results was to show that the class of planar graphs and all classes of bounded tree width are IFP+C-*canonisable*, that is, they admit a canonisation mapping definable in IFP+C by means of a syntactical interpretation. A *canonisation mapping* for a class $\mathscr{C}$ of structures may be viewed as a mapping that associates with each structure $A \in \mathscr{C}$ a canonical ordered copy of the structure.

The present paper is a continuation of the line of work of [9, 10, 12]. The question we address here is the following: Suppose we have structures that can be decomposed into simpler structures. Then how can we lift definability results from the simpler structures to the decomposable structures. The technical notion of decomposition we use here is that of tree decompositions of graphs over simpler graphs, which are called the *torsi* of the decomposition. For example, all graphs have a tree decomposition whose torsi are the *3-connected components* of the graphs. Graphs have *bounded tree width* if and only if they have a tree decomposition whose torsi are of bounded size. A well-known theorem due to Wagner [27] states that K_5-*free* graphs, that is, graphs that do not contain the complete 5-vertex graph K_5 as a minor, have a tree decomposition whose torsi are either planar graphs or subgraphs of one exceptional 8-vertex graph L (shown in Figure 4.2). Wagner's theorem was generalised by Robertson and Seymour [25] to a powerful structure theorem for arbitrary classes of graphs with excluded minors, which states that for every $k \geq 1$, all K_k-free graphs have a tree decomposition whose torsi are almost embeddable into some fixed surface.

If we want to lift definability results from the torsi of a structure's tree decomposition to the whole structure, we need to be able to define the tree decomposition. The main contribution of this paper is to come up with a notion of *definable tree decomposition* that allows us to lift definability results and results about canonisation and capturing polynomial time. The technical problem in trying to define tree decompositions is that definable sets are invariant under isomorphisms, whereas tree decompositions usually are not — think of a tree decomposition of a cycle. Therefore, definable tree decompositions are more complicated structures than normal tree decompositions. Intuitively, they may be viewed as directed acyclic graphs whose nodes carry the torsi of the decomposition. These structures do not just represent one tree decomposition of a graph, but many different tree decompositions of parts of the graph.

We prove that the tree decompositions of graphs into their 3-connected components, of bounded tree width graphs into torsi of bounded order, and of K_5-free graphs into planar graphs and subgraphs of L are all definable in *inflationary fixed-point logic* IFP. I believe that Robertson and Seymour's decomposition of K_k-free graphs into almost embeddable torsi is also definable in IFP, but this requires considerable additional work related to the definability of "almost embeddable" graphs, which will not be carried out in this paper.

We prove two general theorems about lifting definability results: The first states that if a class of torsi is definable in IFP, then the class of all graphs with a decomposition over these torsi is also definable in IFP. Once the framework is set up, this result is not difficult to prove. Nevertheless, it allows us to prove that the class of all K_5-free graphs is definable in IFP. It is not obvious how to obtain an IFP-definition of this class in a direct way.

The second result, which is much deeper, is concerned with definable canonisation and descriptive complexity theory. For a class $\mathscr{C}$ of graphs, we let $\mathscr{S}(\mathscr{C})$ be the class of all structures whose Gaifman graph is in $\mathscr{C}$. We prove that if $\mathscr{A},\mathscr{C}$ are classes of graphs, and we have an IFP+C-definable tree decomposition of the graphs in $\mathscr{C}$ with torsi in $\mathscr{A}$, then if $\mathscr{S}(\mathscr{A})$ is IFP+C-canonisable then so is $\mathscr{S}(\mathscr{C})$. To obtain this result, we need to work with two-sorted relational structures with one finite universe of vertices and a universe of all nonnegative integers, and with mixed relations involving vertices and numbers. A consequence of this general result is that IFP+C captures PTIME on the class of K_5-free graphs.

Our general approach using definable tree decomposition also unifies the result from [12] that all classes of structures of bounded tree width are IFP+C-canonisable and a lemma from [9] lifting the canonisability of 3-connected planar graphs to arbitrary finite graphs. Although it was obvious that these two results were proved by very similar ideas, the proofs in [9, 12] were very ad-hoc, and for a long time it was not clear to me how a common generalisation might look.

There is one last consequence of our results that I would like to mention. It is known [3] that a simple combinatorial algorithm known as the Weisfeiler-Leman (WL) algorithm can be used as a polynomial time isomorphism test on classes of graphs that admit IFP+C-definable canonisation. Hence it follows from our results that the WL-algorithm provides an isomorphism test for K_5-free graphs.[1]

[1]It is known that isomorphism for all graph classes with excluded minors, such as the class of K_5-free graphs, can be decided in polynomial time [24]. It seems that the algorithm of [24], which unfortunately is only published in Russian, uses algebraic techniques. Thus the simple combinatorial algorithm we obtain here for K_5-free graphs may be of some interest.

Organisation of the paper

The paper is divided into two parts. In the first part, we introduce definable tree decompositions and prove the definability results. The second part is concerned with definable canonisation. Due to space limitations, we have to omit most of the proofs.[2] To make this conference version more interesting, I decided to put an emphasis on the first part. But this means that the second part has become very condensed and mainly consists of definitions and a statement of the main result.

[2]A full version of this paper is available on my webpage.

Notation

$\mathbb{Z}_{\geq 0}$, and $\mathbb{N}$ denote the sets of nonnegative integers and natural numbers (that is, positive integers), respectively. For $m,n \in \mathbb{Z}_{\geq 0}$, we let $[m,n] := \{\ell \in \mathbb{Z}_{\geq 0} \mid m \leq \ell \leq n\}$ and $[n] := [1,n]$.

We often denote tuples $(v_1,\dots,v_k)$ by $\vec{v}$. If $\vec{v}$ denotes the tuple $(v_1,\dots,v_k)$, then by $\tilde{v}$ we denote the set $\{v_1,\dots,v_k\}$. If $\vec{v} = (v_1,\dots,v_k)$ and $\vec{w} = (w_1,\dots,w_\ell)$, then by $\vec{v}\vec{w}$ we denote the tuple $(v_1,\dots,v_k,w_1,\dots,w_\ell)$. By $|\vec{v}|$ we denote the length of a tuple $\vec{v}$, that is, $|(v_1,\dots,v_k)| = k$.

PART I. DEFINABILITY

2. Preliminaries

2.1. Graphs

Graphs in this paper are always finite, nonempty, and simple, where simple means that there are no loops or parallel edges. Unless explictly called "directed", graphs are undirected. The vertex set of a graph G is denoted by $V(G)$ and the edge set by $E(G)$. We view graphs as relational structures with $E(G)$ being a binary relation on $V(G)$. However, we often find it convenient to view edges (of undirected graphs) as 2-element subsets of $V(G)$ and use notations like $e = \{u,v\}$ and $v \in e$. $\mathscr{G}$ denotes the class of all graphs. Subgraphs, induced subgraphs, union, and intersection of graphs are defined in the usual way. We write $G[W]$ to denote the induced subgraph of G with vertex set $W \subseteq V(G)$, and we write $G \setminus W$ to denote $G[V(G) \setminus W]$. A *minor* of a graph G is a graph that is isomorphic to a graph obtained from a subgraph of G by contracting edges. A graph is *H-free* if it does not contain H as a minor. The *order* of a graph, denoted by $|G|$, is the number of vertices of G.

For every finite nonempty set V, we let $K[V]$ be the *complete graph* with vertex set V. We let $K_n := K\big[[n]\big]$, and we let $K_{m,n}$ be a complete bipartite graph with parts of size

m,n. A *clique* in a graph G is a set $W \subseteq V(G)$ such that $G[W]$ is a complete graph, and an *independent set* in G is a set $W \subseteq V(G)$ such that $E(G[W]) = \emptyset$. *Paths* and *cycles* in graphs are defined in the usual way. The *length* of a path or cycle is the number of its edges. An *internal* vertex of a path is a vertex that is not an endpoint, and a path from W to W' is a path with one endpoint in W and one endpoint in W' and no internal vertex in $W \cup W'$. Two paths are *(internally) disjoint* if they have no (internal) vertex in common. *Connectedness* and *connected* components are defined in the usual way. Let G be a graph. A set $W \subseteq V(G)$ is *connected* if $W \neq \emptyset$ and $G[W]$ is connected. A graph G is *k-connected*, for some $k \geq 1$, if $|G| > k$ and for every $W \subseteq V(G)$ with $|W| < k$ the graph $G \setminus W$ is connected. A set $S \subseteq V(G)$ is a *separator* of G, or *separates* G, if $G \setminus S$ has more than one connected component. S is a *minimal separator* if S, but no proper subset of S, is a separator. The *order* of a separator is its cardinality $|S|$. For sets $W_1, W_2 \subseteq V(G)$, a set $S \subset V(G)$ *separates* W_1 *from* W_2, or is a (W_1,W_2)*-separator*, if there is no path from a vertex in $W_1 \setminus S$ to vertex in $W_2 \setminus S$ in the graph $G \setminus S$. S is a *minimal* (W_1,W_2)*-separator* if S, but no proper subset of S, is a (W_1,W_2)-separator. Recall that by Menger's Theorem there is a family of k disjoint paths from W_1 to W_2 if and only if there is no (W_1,W_2)-separator of order less than k.

A *forest* is an undirected acyclic graph, and a *tree* is a connected forest. It will be a useful convention to call the vertices of trees and forests *nodes*. A *rooted tree* is a triple $T = (V(T),E(T),r(T))$, where $(V(T),E(T))$ is a tree and $r(T) \in V(T)$ is a distinguished node called the *root*. A node s of a rooted tree T is the *ancestor* of a node t, and t is a *descendant* of s, if s appears on the unique path from the root $r(T)$ to t. *Parents* and *children* of a node are ancestors and descendants adjacent to the node.

2.2. Inflationary fixed-point logic

I assume that the reader has a solid background in logic and, in particular, is familiar with the standard fixed-point logics used in finite model theory. Background material can be found in [5, 8, 17, 19]. In the first part of this paper, we shall work with *inflationary fixed-point logic* IFP over graphs. IFP-formulas are build from atomic formulas Exy, expressing incidence, and $x = y$ by the usual propositional connectives, existential and universal quantification over vertices, and a fixed-point operator with inflationary semantics. To follow the first part of the paper, it is sufficient to know that basic graph properties involving connectivity and separators can be expressed in IFP.

Let me just mention one nonstandard piece of notation here: We write $\varphi(x_1,\ldots,x_k)$ to denote that the free variables of the formula φ are among $x_1,\ldots,x_k$. For a graph G and vertices $v_1,\ldots,v_k$, we write $G \models \varphi[v_1,\ldots,v_k]$ to denote that G satisfies φ if x_i is interpreted by v_i for $i \in [k]$. For $i < k$, we let $\varphi[G;v_1,\ldots,v_i,x_{i+1},\ldots,x_k]$ denote the $(k-i)$-ary relation consisting of all tuples $(w_1,\ldots,w_{k-i}) \in V(G)^{k-i}$ such that $G \models \varphi[v_1,\ldots,v_i,w_1,\ldots,w_{k-i}]$.

3. Tree decompositions

A *tree decomposition* of a graph G is a pair (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 for every $v \in V(G)$ the set $\{t \in V(T) \mid v \in B_t\}$ is connected in T, and for every $e \in E(G)$ there is a $t \in V(T)$ such that $e \subseteq B_t$. The sets B_t, for $t \in V(T)$, are called the *bags* of the decomposition. It is sometimes convenient to have the tree T in a tree decomposition rooted; we always assume it is. The *width* of a tree decomposition (T,B) is $\max\{|B_t| - 1 \mid t \in V(T)\}$. The *tree width* of a graph G, denoted by $\mathrm{tw}(G)$, is the minimum of the widths of all tree decompositions of G. The *adhesion* of a tree decomposition (T,B) is $\max\{|B_s \cap B_t| \mid \{s,t\} \in E(T)\}$.

Let (T,B) be a tree decomposition of a graph G and $t \in V(T)$ such that $B_t \neq \emptyset$. The *torso* $[\![B_t]\!]$ at t is the graph

$$[\![B_t]\!] := G[B_t] \quad \cup \bigcup_{s \text{ with } \{s,t\} \in E(T)} K[B_s \cap B_t].$$

(T,B) is a tree decomposition *over* a class $\mathscr{C}$ of graphs if all its torsi belong to $\mathscr{C}$. Note that torsi are only defined for nodes with nonempty bags. For every node $t \in V(T)$, we let

$$B_{\geq t} := \bigcup_{s=t \text{ or } s \text{ descendant of } t} B_s.$$

3.1. Definable tree decompositions

Definition 3.1. Let L be a logic

(1) A *k-ary TD-scheme in* L is a tuple

$$\Theta = \big(\theta_V(\vec{x}), \theta_E(\vec{x},\vec{x}'), \theta_{sep}(\vec{x},y), \theta_{comp}(\vec{x},y)\big)$$

of L-formulas in the language of graphs, where $\vec{x},\vec{x}'$ are k-tuples of variables.

In the following, let G be a graph and Θ a k-ary TD-scheme.

(2) All tuples $\vec{v} \in V(G)^k$ with $G \models \theta_V[\vec{v}]$ are called Θ-*nodes (in G)*, and for all Θ-nodes $\vec{v},\vec{v}'$, if $G \models \theta_E[\vec{v},\vec{v}']$ then $\vec{v}'$ is a called Θ-*child* of $\vec{v}$. For every Θ-node $\vec{v}$ we let

$$S^{\Theta}_{\vec{v}} := \theta_{sep}[G;\vec{v},y],$$
$$C^{\Theta}_{\vec{v}} := \theta_{comp}[G;\vec{v},y],$$
$$B^{\Theta}_{\geq \vec{v}} := S^{\Theta}_{\vec{v}} \cup C^{\Theta}_{\vec{v}}.$$

We call Θ-nodes $\vec{v},\vec{v}'$ Θ-*equivalent* if $S^{\Theta}_{\vec{v}} = S^{\Theta}_{\vec{v}'}$ and $C^{\Theta}_{\vec{v}} = C^{\Theta}_{\vec{v}'}$.

(3) Θ *defines a tree decomposition on G* if the following conditions are satisfied:

   (i) For every Θ-node $\vec{v}$, the set $C^\Theta_{\vec{v}}$ is the vertex set of a connected component of $G \setminus S^\Theta_{\vec{v}}$.

   (ii) For every connected component C of G there is at least one Θ-node $\vec{v}$ with $C^\Theta_{\vec{v}} = V(C)$.

   Each such node is called a *root node for C*.

   (iii) If a Θ-node $\vec{v}'$ is a Θ-child of a Θ-node $\vec{v}$, then $B^\Theta_{\geq \vec{v}'} \subseteq B^\Theta_{\geq \vec{v}}$ and $C^\Theta_{\geq \vec{v}'} \subset C^\Theta_{\geq \vec{v}}$.

   (iv) For all Θ-nodes $\vec{v}$ and all Θ-children $\vec{v}_1, \vec{v}_2$ of $\vec{v}$, either $\vec{v}_1$ and $\vec{v}_2$ are Θ-equivalent, or $B^\Theta_{\geq \vec{v}_1} \cap B^\Theta_{\geq \vec{v}_2} = S^\Theta_{\vec{v}_1} \cap S^\Theta_{\vec{v}_2}$.

In the following, we assume that Θ defines a tree decomposition on G, and we let $\vec{v}$ be a Θ-node.

(4) The *bag defined by Θ at $\vec{v}$* is the set

$$B^\Theta_{\vec{v}} := B^\Theta_{\geq \vec{v}} \setminus \bigcup_{\vec{v}' \ \Theta\text{-child of } \vec{v}} C^\Theta_{\vec{v}'}.$$

(5) The *torso defined by Θ at $\vec{v}$* is the graph

$$\left[\!\left[B^\Theta_{\vec{v}} \right]\!\right] := G[B^\Theta_{\vec{v}}] \cup K[S^\Theta_{\vec{v}}] \cup \bigcup_{\vec{v}' \ \Theta\text{-child of } \vec{v}} K[S^\Theta_{\vec{v}'}].$$

(6) The *adhesion* of the decomposition defined by Θ on G is $\max\left\{ |S_{\vec{v}}| \;\middle|\; \vec{v}\ \Theta\text{-node} \right\}$. ⌟

Remark 3.2. In this paper, we only consider TD-schemes in the logic IFP. Hence from now on, all TD-schemes will be be in IFP, and we will not mention this anymore when we introduce them.

It is straightforward to generalise the definition of definable tree decompositions from graphs to arbitrary structures. But as the classes of structures we study here are always defined in terms of their Gaifman graphs, we would not gain much from this generalisation. ⌟

Remark 3.3. While the definition of definable tree decomposition may seem quite generic, it took me a while to arrive at this definition. Two important aspects of the definition that are not entirely obvious are: (A) The definition is based on the sets $B_{\geq t}$ rather than the bags B_t, and (B) the nodes $\vec{v}$ are not linked to the separators $S_{\vec{v}}$ or the bags in a direct way. (A) will be crucial in the proof of Lemma 3.6; via Definition 3.1 (iii) it guarantees the acyclicity of the graph defined by the Θ-child relation. (B) is important in the proofs of Theorem 4.1 and Lemma 4.8, where the Θ-nodes $\vec{v}$ will be chosen in such a way that they do not only control the bag at the current node, but also the children of the node. The following example illustrates this point. ⌟

Example 3.4. Suppose we want to define a tree decomposition of some graph G by a TD-scheme Θ, and suppose that at some point during the decomposition we are at a Θ-node $\vec{v}$ where $B^\Theta_{\geq \vec{v}}$ is a cycle of odd length, say with vertices $w_0, \ldots, w_{2n}$ in cyclic order, and we have $S^\Theta_{\vec{v}} = \{w_0\}$ and hence $C_{\vec{v}} = \{w_1, \ldots, w_{2n}\}$. The obvious way to decompose the cycle would be to pick an $i \in [2n]$ and attach two children $\vec{v}_1, \vec{v}_2$ to $\vec{v}$ with separators $S^\Theta_{\vec{v}_1} = S^\Theta_{\vec{v}_2} = \{w_0, w_i\}$ and components $C^\Theta_{\vec{v}_1} = \{w_1, \ldots, w_{i-1}\}$, $C^\Theta_{\vec{v}_2} = \{w_{i+1}, \ldots, w_{2n}\}$. But which i shall we choose? There is no distinguished choice, not even a unique "middle" or "first" vertex on the cycle, as there is an automorphism that keeps w_0 fixed and maps w_1 to w_{2n} and w_n to w_{n+1}. It seems as if we can only resolve this by choosing several i and create two Θ-children for all of them. But this would violate condition (iv) of Definition 3.1.

The solution is to make the choice already at the node $\vec{v}$. That is, the tuple $\vec{v}$ will already contain one of the vertices w_i for $i \in [2n]$ that determines the choice of the children at $\vec{v}$. Surprisingly, this solves our problem, because we can simply add one node $\vec{v}_i$ to our decomposition for each choice of w_i. All these nodes will have the same separator and component, hence be equivalent, and therefore not violate Definition 3.1 (iv).

This idea appears, in slightly different forms, in the proofs of Theorem 4.1 and Lemma 4.8. ⌟

Definition 3.5. Suppose that a TD-scheme Θ defines a tree decomposition on a graph G. Then a tree decomposition (T, B) of G is *compatible* with Θ if it satisfies the following conditions:

(i) $B_{r(T)} = \emptyset$ for the root $r(T)$.

(ii) All nodes $t \in V(T) \setminus \{r(T)\}$ are Θ-nodes.

(iii) All children of the root $r(T)$ are root Θ-nodes.

(iv) If a node t' is a child of node $t \neq r(T)$, then t' is a Θ-child of t.

(v) For all nodes $t \in V(T) \setminus \{r(T)\}$, the torso $[\![B_t]\!]$ of the decomposition (T, B) at t is equal to the torso $[\![B^\Theta_t]\!]$ defined by Θ at t. ⌟

Lemma 3.6. *Suppose that a TD-scheme Θ defines a tree decomposition on a graph G. Then there is a tree decomposition (T, B) of G that is compatible with Θ.*

Proof. To distinguish between nodes of the tree T to be defined and Θ-nodes, we call the former *T-nodes*. Similarly, we speak of *T-children*. We define T inductively:

– We create a root node $r = r(T)$, which is not a Θ-node. All other T-nodes will be Θ-nodes.

– For every connected component C of G, we arbitrarily choose a Θ-node $\vec{v}$ with $C^\Theta_{\vec{v}} = V(C)$ and make it a child of the root r.

– For every T-node $\vec{v} \neq r$, we arbitrarily choose Θ-children $\vec{v}_1, \ldots, \vec{v}_m$ of $\vec{v}$ such that for every Θ-child $\vec{v}'$ of

$\vec{v}$ there is exactly one $i \in [m]$ such that $\vec{v}'$ is Θ-equivalent to $\vec{v}_i$. We let $\vec{v}_1, \ldots, \vec{v}_m$ be the T-children of $\vec{v}$.

Formally, children are only defined in trees and it is not clear yet that T really is a tree. Therefore, the definition should be read as the definition of a directed graph, where "$\vec{v}'$ is a T-child of $\vec{v}$" simply means that there is an edge in T from $\vec{v}$ to $\vec{v}'$. *Directed paths* in T are paths in this directed graph. In Claim 2, we shall proof that T is a tree.

Observe that if $\vec{v}'$ is a Θ-child of $\vec{v}$, then

$$B^{\Theta}_{\vec{v}} \cap B^{\Theta}_{\geq \vec{v}'} = S^{\Theta}_{\vec{v}'}. \tag{3.1}$$

Claim 1: Let $\vec{v}_1, \vec{v}_2 \in V(T) \setminus \{r\}$ such that $\vec{v}_1$ is an ancestor of $\vec{v}_2$ in T. Then for every T-node $\vec{v}$ on a directed path in T from $\vec{v}_1$ to $\vec{v}_2$ it holds that $B^{\Theta}_{\vec{v}_1} \cap B^{\Theta}_{\geq \vec{v}_2} \subseteq S^{\Theta}_{\vec{v}}$.

Proof. This follows from (3.1) by a straightforward induction. ⌟

Claim 2: T is a rooted tree.

Proof. It follows immediately from the definition of T that all nodes are reachable from the root via a directed path. It follows from Definition 3.1 (iii) that T has no directed cycle. Hence if T is not a tree, then there is a T-node $\vec{v}_0$ with T-children $\vec{v}_1$ and $\vec{v}_2$ that have a common descendant $\vec{v}_3$. Then by Definition 3.1 (iii) and (iv)

$$B^{\Theta}_{\geq \vec{v}_3} \subseteq B^{\Theta}_{\geq \vec{v}_1} \cap B^{\Theta}_{\geq \vec{v}_2} \subseteq S^{\Theta}_{\vec{v}_1} \cap S^{\Theta}_{\vec{v}_2} \subseteq B^{\Theta}_{\vec{v}_0}.$$

Hence by Claim 1,

$$B^{\Theta}_{\geq \vec{v}_3} = B^{\Theta}_{\geq \vec{v}_3} \cap B^{\Theta}_{\vec{v}_0} \subseteq S^{\Theta}_{\geq \vec{v}_3}.$$

It follows that $C^{\Theta}_{\vec{v}_3} = \emptyset$, which is a contradiction. ⌟

We let $B_r := \emptyset$ and $B_{\vec{v}} := B^{\Theta}_{\vec{v}}$ for every $\vec{v} \in V(T) \setminus \{r\}$. For every $w \in V(G)$, we let $B^{-1}(w) := \{\vec{v} \in V(T) \mid w \in B_{\vec{v}}\}$.

Claim 3: For every $w \in V(G)$ the set $B^{-1}(w)$ is connected in T.

Proof. To prove that $B^{-1}(w)$ is nonempty, let $\vec{v} \in V(T)$ such that $w \in B^{\Theta}_{\geq v}$ and $w \notin B^{\Theta}_{\geq v'}$ for all T-children $\vec{v}'$ of $\vec{v}$. Suppose for contradiction that $w \notin B_{\vec{v}}$. Then by the definition of $B^{\Theta}_{\vec{v}}$, there is a Θ-child $\vec{v}''$ of $\vec{v}$ such that $w \in C^{\Theta}_{\vec{v}''}$. But then there also is a T-child $\vec{v}'$ of $\vec{v}$ such that $w \in C^{\Theta}_{\vec{v}'} \subseteq B^{\Theta}_{\geq v'}$. This is a contradiction, which proves that $B^{-1}(w)$ is nonempty.

To see that $B^{-1}(w)$ is connected, suppose that $w \in B_{\vec{v}_1} \cap B_{\vec{v}_2}$ for two T-nodes $\vec{v}_1, \vec{v}_2$. We shall prove that $w \in B_{\vec{v}}$ for every $\vec{v}$ on the (undirected) path in T from $\vec{v}_1$ to $\vec{v}_2$. This follows immediately from Claim 1 if $\vec{v}_1$ is an ancestor of $\vec{v}_2$ or vice-versa. Otherwise, let $\vec{v}_3$ be the last common ancestor of $\vec{v}_1$ and $\vec{v}_2$, and for $i = 1, 2$, let $\vec{v}_i'$ be the T-child of $\vec{v}_3$ on the path from $\vec{v}_3$ to $\vec{v}_i$. Then $\vec{v}_1' \not\equiv^{\Theta} \vec{v}_2'$, and we have

$$w \in B_{\vec{v}_1} \cap B_{\vec{v}_2} \subseteq B^{\Theta}_{\geq \vec{v}_1'} \cap B^{\Theta}_{\geq \vec{v}_1'} \subseteq S^{\Theta}_{\vec{v}_1'} \cap S^{\Theta}_{\vec{v}_2'} \subseteq B_{\vec{v}_3}.$$

It follows from Claim 1 that $w \in B_{\vec{v}}$ for every $\vec{v}$ on the path in T from $\vec{v}_3$ to $\vec{v}_1$ and for every $\vec{v}$ on the path in T from $\vec{v}_3$ to $\vec{v}_2$. Hence $w \in B_{\vec{v}}$ for every $\vec{v}$ on the path in T from $\vec{v}_1$ to $\vec{v}_2$. ⌟

Claim 4: For every edge $e \in E(G)$ there is a node $\vec{v} \in V(T)$ such that $e \subseteq B_{\vec{v}}$.

Proof. Let $\vec{v} \in V(T)$ such that $e \subseteq B^{\Theta}_{\geq v}$ and $e \not\subseteq B^{\Theta}_{\geq v'}$ for all T-children $\vec{v}'$ of $\vec{v}$. Suppose for contradiction that $e \not\subseteq B_{\vec{v}}$. Arguing similarly as in the proof of Claim 3, we find a T-child $\vec{v}'$ of $\vec{v}$ such that $e \cap C^{\Theta}_{\vec{v}'} \neq \emptyset$. As $e \not\subseteq B^{\Theta}_{\geq v'} = S^{\Theta}_{\vec{v}'} \cup C^{\Theta}_{\vec{v}'}$, this contradicts Definition 3.1 (i). ⌟

Hence (T, B) is a tree-decomposition of G. It follows immediately from the definitions that (T, B) is compatible with Θ. ☐

Definition 3.7. Let Θ be a TD-scheme. Furthermore, let $\mathscr{A}, \mathscr{C}$ be classes of graphs.

(1) Θ defines a tree decomposition on a graph G *over* $\mathscr{A}$ if all torsi defined by Θ on G are in $\mathscr{A}$.

(2) Θ *defines a tree decomposition on* $\mathscr{C}$ *over* $\mathscr{A}$ if for every $G \in \mathscr{C}$ the scheme Θ defines a tree decomposition on G over $\mathscr{A}$.

(3) The class $\mathscr{C}$ *admits an* IFP*-definable tree decomposition over* $\mathscr{A}$ if there is a TD-scheme that defines a tree decomposition on $\mathscr{C}$ over $\mathscr{A}$. ⌟

Corollary 3.8. *If a TD-scheme Θ defines a tree decomposition on a graph G over a class $\mathscr{A}$ of graphs, then there is a tree decomposition of G over $\mathscr{A}$.*

We close this section with two simple lemmas, whose straightforward proofs we omit.

Lemma 3.9. *Let $\mathscr{A}, \mathscr{B}, \mathscr{C}$ be classes of graphs such that $\mathscr{C}$ admits an* IFP*-definable tree decomposition over $\mathscr{B}$ and $\mathscr{B}$ admits an* IFP*-definable tree decomposition over $\mathscr{A}$. Then $\mathscr{C}$ admits an* IFP*-definable tree decomposition over $\mathscr{A}$.*

Lemma 3.10. *There is a TD-scheme Θ such that for all graphs G, the scheme Θ defines a tree decomposition on G and the torsi defined by Θ on G are precisely the connected components of G.*

3.2. Definability results

Lemma 3.11. *Let Θ be a k-ary TD-scheme.*

(1) There is an IFP*-sentence td_{Θ} such that for all graphs G we have $G \models td_{\Theta}$ if and only if Θ defines a tree decomposition on G.*

(2) There is an IFP*-formula $bag_{\Theta}(\vec{x}, y)$ such that for every graph G and every tuple $\vec{v} \in V(G)^k$, if Θ defines a tree decomposition on G and $\vec{v}$ is a Θ-node, then $bag_{\Theta}[G; \vec{v}, y] = B^{\Theta}_{\vec{v}}$.*

(3) There is an IFP*-formula* $torso_\Theta(\vec{x},y,z)$ *such that for every graph G and every tuple* $\vec{v} \in V(G)^k$*, if* Θ *defines a tree decomposition on G and* $\vec{v}$ *is a* Θ*-node, then* $torso_\Theta[G;\vec{v},y,z]$ *is the edge relation of the torso* $[\![B^\Theta_{\vec{v}}]\!]$.

Proof. This follows immediately from the fact that the sets $S^\Theta_{\vec{v}}$, $C^\Theta_{\vec{v}}$, $B^\Theta_{\geq\vec{v}}$ and the Θ-child relation are definable by the formulas appearing in Θ (by definition) and that graph reachability is definable in IFP. □

Theorem 3.12 (First Lifting Theorem). *Let* Θ *be a TD-scheme. Let* $\mathscr{A}$ *be an* IFP*-definable class of graphs, and let* $\mathscr{T}_\Theta(\mathscr{A})$ *be the class of all graphs G such that* Θ *defines a tree decomposition on G over* $\mathscr{A}$*.*

Then $\mathscr{T}_\Theta(\mathscr{A})$ *is* IFP*-definable.*

Proof. Follows easily from the previous lemma. □

Corollary 3.13. *Let* $\mathscr{A}$ *be an* IFP*-definable class of graphs, and let* $\mathscr{T}(\mathscr{A})$ *be the class of all graphs that have a tree decomposition over* $\mathscr{A}$*. Suppose that* $\mathscr{T}(\mathscr{A})$ *admits an* IFP*-definable tree decomposition over* $\mathscr{A}$*.*

Then $\mathscr{T}(\mathscr{A})$ *is* IFP*-definable.*

Definition 3.14. A TD-scheme Θ defines a tree decomposition on a graph G *weakly over* a class $\mathscr{A}$ if there is a tree decomposition of G over $\mathscr{A}$ that is compatible with Θ. ⌟

Lemma 3.15. *Let* $\mathscr{A}$ *be an* IFP*-definable class of graphs. Then for every TD-scheme* Θ *there is a TD-scheme* Θ' *such that for all graphs G, if* Θ *defines a tree decomposition on G that is weakly over* $\mathscr{A}$ *then* Θ' *defines a tree decomposition on G over* $\mathscr{A}$*.*

Proof. Let G be a graph such that Θ defines a tree decomposition on G that is weakly over $\mathscr{A}$. Let (T,B) be a tree decomposition of G that is compatible with Θ.

We inductively define sets $\mathscr{N}_i$, for $i \in \mathbb{Z}_{\geq 0}$, of Θ-nodes as follows:

- $\mathscr{N}_0 := \emptyset$.
- $\mathscr{N}_{i+1}$ is the set of all Θ-nodes $\vec{v}$ such that $[\![B^\Theta_{\vec{v}}]\!] \in \mathscr{A}$,

$$B^\Theta_{\vec{v}} = B^\Theta_{\geq\vec{v}} \setminus \bigcup_{\vec{v}' \in \mathscr{N}_i,\, G \models \theta_E[\vec{v},\vec{v}']} C^\Theta_{\vec{v}'},$$

and

$$[\![B^\Theta_{\vec{v}}]\!] = G[B^\Theta_{\vec{v}}] \cup K[S_{\vec{v}}] \cup \bigcup_{\vec{v}' \in \mathscr{N}_i,\, G \models \theta_E[\vec{v},\vec{v}']} K[S_{\vec{v}'}].$$

A straightforward induction shows that all nodes of T except the root are in $\bigcup_{i\geq 0}\mathscr{N}_i$.

Furthermore, it is easy to define an IFP-formula $\varphi(\vec{x})$ (not depending on G) such that $\varphi[G;\vec{x}] = \bigcup_{i\geq 0}\mathscr{N}_i$. We let $\theta'_V(\vec{x}) := \theta_V(\vec{x}) \wedge \varphi(\vec{x})$, and $\theta'_E := \theta_E$, $\theta'_{sep} := \theta_{sep}$, $\theta'_{comp} := \theta_{comp}$. Then $\Theta' := (\theta'_V, \theta'_E, \theta'_{sep}, \theta'_{comp})$ defines a tree decomposition on G over $\mathscr{A}$. □

4. Applications

4.1. Graphs of bounded tree width

For every $k \in \mathbb{N}$, let $\mathscr{G}_k$ denote the class of all graphs of order at most k, and let $\mathscr{T}_{k-1}$ be the class of all structures of tree width at most $k-1$. Observe that a structure is in $\mathscr{T}_{k-1}$ if and only if it has a tree decomposition over $\mathscr{G}_k$.

Theorem 4.1. *For every* $k \in \mathbb{N}$*, the class* $\mathscr{T}_{k-1}$ *admits an* IFP*-definable tree decomposition over* $\mathscr{G}_k$*.*

Corollary 4.2 (Grohe and Mariño [12]). *For every* $k \in \mathbb{Z}_{\geq 0}$*, the class* $\mathscr{T}_k$ *is* IFP*-definable.*

4.2. Hinges

Definition 4.3. A *hinge* of a graph G is a minimal separator S of G such that for every connected component C of $G \setminus S$ the graph $G\big[V(C) \cup S\big] \cup K[S]$ is a minor of G. ⌟

The basic idea that we pursue in the following is to recursively separate a graph along small hinges until no more small hinges are left. This gives us a tree decomposition where the torsi have no small hinges and are minors of the graph. Thinking of a graph without small hinges as "highly connected", we have thus decomposed our graph into highly connected components that are all minors of the graph. If we are lucky, this decomposition is even definable. Unfortunately, this idea only works for hinges of order at most 3. Hinges of order 1 and 2 are just minimal separators, hence a graph without hinges of order 1 and 2 is 3-connected in the usual sense. The analogue for hinges of order 3 does not holds; there are minimal separators of order 3 that are no hinges. However, the following lemma shows that in 3-connected graphs, minimal separators of order 3 that are no hinges can just separate single vertices from a graph. This will be good enough for us.

Lemmas 4.4 and 4.6 contain the graph theory that goes into the proof of Lemma 4.8, the main result of this section.

Lemma 4.4. *Let G be a 3-connected graph and S a separator of G of order* 3*. Then either S is a hinge, or S is an independent set and* $G \setminus S$ *has exactly two connected components, one of which has order* 1*.*

Proof. Let $S = \{v_1, v_2, v_3\}$. As G is 3-connected, S is minimal

We first prove that if S is not an independent set, then S is a hinge. To see this, suppose that $\{v_1, v_2\} \in E(G)$, and let C, C' be two connected components of $G \setminus S$. Let $w \in C'$. As G is 3-connected, there are internally disjoint paths P_i, for $i \in [3]$, from w to v_i. By deleting all vertices not in $V(C) \cup S \cup \bigcup_{i=1}^{3} V(P_i)$, contracting P_3 to a single vertex, and contracting P_1 and P_2 to single edges, we see that $G[V(C) \cup S] \cup K[S]$ is a minor of G.

It follows that if $G \setminus S$ has more than two connected components, then S is a hinge, because by contracting one of the components we can obtain an edge between vertices in S.

So suppose that $G \setminus S$ has precisely two connected components, say, C_1 and C_2, and that $|C_1| \geq 2$ and $|C_2| \geq 2$. Let $w \in V(C_2)$. As G is 3-connected, there are internally disjoint paths P_i, for $i \in [3]$, from w to v_i. At least one of these paths has length ≥ 2. If not, $\{w\}$ is a separator of G, which contradicts the 3-connectedness of G. Say, P_1 has length at least 2. Then $V(P_1) \setminus \{v_1, w\} \neq \emptyset$. Let Q be a path from $V(P_1) \setminus \{v_1, w\}$ to $V(P_2) \cup V(P_3)$ in $G \setminus \{v_1, w\}$; such a path exists because G is 3-connected. Say, $w_1 \in V(P_1)$ and $w_2 \in V(P_2)$ are the endpoint of Q. By contracting the piece of P_1 from w_1 to v_1 and the piece P_2 from w_2 to v_2, and contracting Q to a single edge, we obtain an edge between v_1 and v_2. Now we can argue as above to prove that $G[V(C_1) \cup S] \cup K[S]$ is a minor of G. By symmetry, $G[V(C_2) \cup S] \cup K[S]$ is also minor of G. Hence S is a hinge. □

Corollary 4.5. *For all $k \in [3]$ there is an* IFP*-formula $hinge_k(x_1, \ldots, x_k)$ such that for every k-connected graph G and every tuple $\vec{v} \in V(G)^k$ we have*

$$G \models hinge_k[\vec{v}] \iff \vec{v} \text{ is a hinge.}$$

Proof. The proofs for $k = 1, 2$ are straightforward because hinges of order 1 and 2 are just minimal separators, and the statement for $k = 3$ follows from Lemma 4.4. □

Let G be a graph and $W \subseteq V(G)$. We say that a hinge S of G is *inseparable* from W if there is no hinge S' such that $|S'| \leq |S|$, $W, S \not\subseteq S'$, and S' separates S from W. The *sphere around* W is the set $S(W)$ of all vertices in $V(G) \setminus W$ that are adjacent to a vertex in W. For a subgraph $H \subseteq G$, we let $S(H) := S(V(H))$. A *hinge extension of W of order at most k* is a vertex $v \in V(G) \setminus W$ such that for every connected component C of $G \setminus (W \cup \{v\})$, the sphere $S(C)$ around C is a hinge of order at most k.

Lemma 4.6. *Let $k \in [3]$. Let G be a k-connected graph and $W \subseteq V(G)$ a clique of order k in G such that there is no hinge extension of W of order k. Let S_1, S_2 be hinges of order k of G such that:*

(i) $S_1 \neq S_2$ and $W \not\subseteq S_i$ for $i = 1, 2$.

(ii) For $i = 1, 2$, the hinge S_i is inseparable from W.

For $i = 1, 2$, let C_i be a connected component of $G \setminus S_i$ such that $W \cap V(C_i) = \emptyset$. Then $V(C_1) \cap V(C_2) = \emptyset$.

Proof. We only prove the lemma for $k = 3$. The proofs for $k = 1$ and $k = 2$ are similar, but simpler.

For $i = 1, 2$, let A_i be the connected component of $G \setminus S_i$ with $W \cap V(A_i) \neq \emptyset$, and let B_i be the union of all other

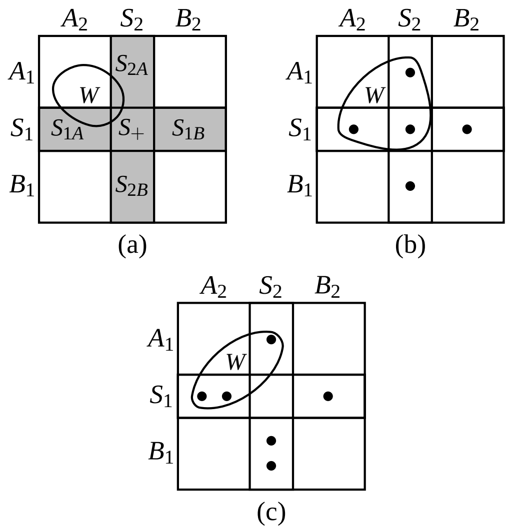

Figure 4.1.

connected components of $G \setminus S_i$. Then $C_i \subseteq B_i$. To simplify the notation, for the rest of this proof we do not distinguish between the graphs A_i and B_i and their vertex sets. Let $S_+ := S_1 \cap S_2$ and $S_{iA} := S_i \cap A_{3-i}$, $S_{iB} := S_i \cap B_{3-i}$ for $i = 1, 2$. Figure 4.1(a) illustrates the situation.

As S_i does not separate S_{3-i} from W, we have

$$|S_{1A}| \geq 1, \qquad |S_{2A}| \geq 1. \tag{4.1}$$

Suppose for contradiction that $V(C_1) \cap V(C_2) \neq \emptyset$. Then $B_1 \cap B_2 \neq \emptyset$, and hence $S_+ \cup S_{1B} \cup S_{2B}$ is a separator of G. Since G is 3-connected, this implies

$$|S_+ \cup S_{1B} \cup S_{2B}| \geq 3. \tag{4.2}$$

Case 1: $W \subseteq S_1 \cup S_2$.
Then either $|W \cap S_1| \geq 2$ or $|W \cap S_2| \geq 2$. Without loss of generality we assume that $|W \cap S_1| \geq 2$. Figures 4.1 (b) and (c) show the only two configurations that are consistent with all the constraints we have derived or imposed so far.

I claim that in both configurations the unique vertex in S_{1B} is a hinge extension of W of order 3. To see this, let C be a connected component of $G \setminus (W \cup S_{1B})$. Suppose first that $S(C) = W \cup S_{1B}$. Then there is a path P with internal vertices in C from S_{1A} to S_{1B}. This path must intersect S_2, which is only possible in S_{2B}. Hence $S_{2B} \cap V(C) \neq \emptyset$. But then there is a path from S_{2B} to S_{2A} which does not intersect S_1, which is impossible. Thus $S(C) \subset W \cup S_{1B}$ and hence $|S(C)| = 3$. As $|S_{1B}| = 1$, it follows that $S(C)$ contains at least two vertices of the clique W. Hence $S(C)$ is not an independent set and therefore a hinge by Lemma 4.4.

Thus the vertex in S_{1B} is indeed a hinge extension of W. This is a contradiction, because we assumed that W has no hinge extensions of order 3. ⌟

Case 2: $W \not\subseteq S_1 \cup S_2$.
Then $A_1 \cap A_2 \neq \emptyset$. Hence $S := S_+ \cup S_{1A} \cup S_{2A}$ separates $A_1 \cap A_2$ from $B_1 \cap B_2$. By the 3-connectedness of G it follows that $|S| \geq 3$, and by (4.2) we obtain $|S| = 3$. I claim that S is a hinge. If $|W \cap S| \geq 2$, then S is not an independent set. If $|W \cap S| \leq 1$ and G has exactly two connected components, then one of these components contains $W \setminus S$ and the other contains the nonempty sets S_{1B} and S_{2B}. Hence both components have order at least 2. Otherwise, $G \setminus S$ has more than two connected components. In all cases, S is a hinge by Lemma 4.4. As S separates S_1 from W, this contradicts assumption (ii). □

Corollary 4.7. *Let $k \in [3]$. Let G be a k-connected graph. Let S be a hinge of order k in G and C a connected component of $G \setminus S$ such that there is no hinge extension of order k of S in C. Let $S_1, S_2 \subseteq V(C) \cup S$ be hinges of order k of G such that:*

(i) $S_1 \neq S_2$ and $S_i \neq S$ for $i = 1, 2$.

(ii) For $i = 1, 2$, the hinge S_i is inseparable from S.

For $i = 1, 2$, let C_i be a connected component of $G \setminus S_i$ such that $S \cap C_i = \emptyset$. Then $V(C_1) \cap V(C_2) = \emptyset$.

Proof. Apply Lemma 4.6 to the graph $G[V(C) \cup S] \cup K[S]$. □

Lemma 4.8. *For all $k \in [3]$, there is a TD-scheme Θ that for all k-connected graphs G defines a tree decomposition on G such that for all torsi $[\![B]\!]$ defined by Θ on G:*

(1) $[\![B]\!]$ is a minor of G.

(2) $[\![B]\!]$ has no hinges of order $\leq k$.

Proof. We only prove the lemma for $k = 3$. The proofs for $k = 1$ and $k = 2$ are similar, but simpler. For simplicity, all hinges and hinge extensions in this proof are assumed to be of order 3.

It suffices to define a Θ that defines a tree decomposition with the desired properties on all 3-connected graphs G with at least one hinge, because by Corollary 4.5 the class of 3-connected graphs with at least one hinge is IFP-definable.

We shall define a 6-ary TD-scheme that for every 3-connected graph G with at least one hinge defines a tree decomposition on G with the desired properties.

To simplify the notation, for 6-tuples $\vec{v} = (v_1, \ldots, v_6)$ we let $\tilde{v}_I := (v_1, v_2, v_3)$ and $\vec{v}_{II} := (v_4, v_5, v_6)$.

Let G be a 3-connected graph with at least one hinge. The IFP-formulas we shall define in the following will not depend on G. The decomposition defined by Θ on G has three kinds of nodes: root nodes, e-nodes, and h-nodes. *Root nodes* are tuples $\vec{v} \in V(G)^6$ where $\tilde{v}_I$ is a hinge and $\vec{v}_{II} = \vec{v}_I$. *E-nodes* are tuples $\vec{v} \in V(G)^6$ where $\tilde{v}_I$ is a hinge and $v_4 = v_5 = v_6$ is a hinge extension of $\tilde{v}_I$. *H-nodes* are tuples $\vec{v} \in V(G)^6$ where $\tilde{v}_I \neq \tilde{v}_{II}$ are both hinges with the following properties: There is a connected component C of $G \setminus \tilde{v}_I$ such that

- there is no hinge extension of $\tilde{v}_I$ in C;
- $\tilde{v}_{II} \subseteq V(C) \cup \tilde{v}_I$;
- $\tilde{v}_{II}$ is inseparable from $\tilde{v}_I$.

It is easy to define IFP-formulas *root*$(\vec{x})$, *e-node*$(\vec{x})$, and *h-node*$(\vec{x})$ that define the root nodes, h-nodes, and e-nodes. We let $\theta_V(\vec{x})$ be the disjunction of these three formulas.

We define the IFP-formula $\theta_{sep}(\vec{x}, y)$ in such a way that for all root nodes $\vec{v}$ we have $\theta_{sep}[G; \vec{v}, y] = \emptyset$, and for all e-nodes or h-nodes we have $\theta_{sep}[G; \vec{v}, y] = \tilde{v}_I$. We define the formula $\theta_{comp}(\vec{x}, y)$ in such a way that for all root nodes $\vec{v}$ we have $\theta_{comp}[G; \vec{v}, y] = V(G)$, and for all e-nodes or h-nodes, $\theta_{comp}[G; \vec{v}, y]$ is the vertex set of the connected component of $G \setminus \tilde{v}_I$ that contains $\tilde{v}_{II} \setminus \tilde{v}_I$.

In the following, for all nodes $\vec{v}$ we let $S_{\vec{v}} := \theta_{sep}[G; \vec{v}, y]$, $C_{\vec{v}} := \theta_{comp}[G; \vec{v}, y]$, and $B_{\geq \vec{v}} := S_{\vec{v}} \cup C_{\vec{v}}$. This is the same notation as in Definition 3.1, except that we omit the superscript $^{\Theta}$.

We define the formula $\theta_E(\vec{x}, \vec{x}')$ in such a way that for all nodes $\vec{v}, \vec{v}'$ we have $G \models \theta_E[\vec{v}, \vec{v}']$ if and only if the following conditions are satisfied:

- $\vec{v}'$ is an e-node or h-node.
- $S_{\vec{v}'}$ separates $S_{\vec{v}}$ from $C_{\vec{v}'}$.
- If $\vec{v}$ is a root node or an h-node, then $\vec{v}'_I = \vec{v}_{II}$.
- If $\vec{v}$ is an e-node, then $\tilde{v}'_I \subseteq \tilde{v}$.

Claim 1: Θ defines a tree decomposition on G.

Proof. Condition (i) Definition 3.1 is obviously satisfied. Condition (ii) is satisfied because G has hinges and hence there are root nodes.

To see that condition (iii) holds, let $\vec{v}'$ be a child of $\vec{v}$. Then $S_{\vec{v}'} \subseteq S_{\vec{v}} \cup C_{\vec{v}}$ by the definition of the nodes, and hence $C_{\vec{v}'} \subseteq C_{\vec{v}}$, because $S_{\vec{v}'}$ separates $S_{\vec{v}}$ from $C^{\Theta}_{\vec{v}'}$. Hence $B_{\geq \vec{v}'} \subseteq B_{\geq \vec{v}}$. Furthermore, $C_{\vec{v}'} \neq C_{\vec{v}}$ because $S_{\vec{v}'} \cap C_{\vec{v}} \neq \emptyset$.

It remains to verify condition (iv). Let $\vec{v}$ be a node with children $\vec{v}_1, \vec{v}_2$. Let $S := S_{\vec{v}}$, $C := C_{\vec{v}}$, and $S_i := S_{\vec{v}_i}$, $C_i := C_{\vec{v}_i}$ for $i = 1, 2$. If $\vec{v}$ is a root node, then $S_1 = S_2 = \tilde{v}$, and both C_1 and C_2 are vertex sets of connected components of $G \setminus S_1$. Hence either $C_1 = C_2$ or $C_1 \cap C_2 = \emptyset$.

If $\vec{v}$ is an e-node, then v_4 is a hinge extension of S in C, and both C_1 and C_2 are vertex sets of connected components of $G \setminus (S \cup \{v_4\})$. Hence C_1 and C_2 are either equal or disjoint. If they are equal, then so are $S_1 = S(C_1)$ and $S_2 = S(C_2)$, and hence $\vec{v}_1$ and $\vec{v}_2$ are Θ-equivalent.

If $\vec{v}$ is an h-node, then there are no hinge extensions of S in C. Hence $C_1 \cap C_2 = \emptyset$ by Corollary 4.7. ⌟

Claim 2: The torsi defined by Θ on G have no hinges.

Proof. Consider the torso $[\![B_{\vec{v}}]\!]$ defined by Θ at a node $\vec{v}$. If $\vec{v}$ is a root node, then $S := \tilde{v}$ is a hinge, and for every connected component C of $G \setminus S$ there is a child $\vec{v}'$ of $\vec{v}$ with

$C_{\vec{v}'} = V(C)$. Hence $B_{\vec{v}} = S$ and $[\![B_{\vec{v}}]\!] = K[S]$, which clearly does not have a hinge.

If $\vec{v}$ is an e-node, then v_4 is a hinge extension of $S_{\vec{v}}$ in $C_{\vec{v}}$. Then for every connected component C of $G \setminus (S_{\vec{v}} \cup \{v_4\})$ with $V(C) \subseteq C_{\vec{v}}$ there is a child $\vec{v}'$ of $\vec{v}$ with $C_{\vec{v}'} = V(C)$. Hence $B_{\vec{v}} = S \cup \{v_4\}$. Furthermore, $[\![B_{\vec{v}}]\!] = K[S_{\vec{v}} \cup \{v_4\}]$ because by the 3-connectedness of G there are three internally disjoint paths from v_4 to the vertices of $S_{\vec{v}}$. Again, $[\![B_{\vec{v}}]\!]$ does not have a hinge.

Finally, let $\vec{v}$ be an h-node. Suppose for contradiction that $[\![B_{\vec{v}}]\!]$ has a hinge S. If S is inseparable from $S_{\vec{v}}$, then for every connected component C of $G \setminus S$ that has an empty intersection with $S_{\vec{v}}$ there is a child $\vec{v}'$ of $\vec{v}$ with $S_{\vec{v}'} = S$ and $C_{\vec{v}'} = V(C)$. Then S does not separate $[\![B_{\vec{v}}]\!]$, which is a contradiction. If S is separable from $S_{\vec{v}}$, let S' be a hinge that is inseparable from $S_{\vec{v}}$ and that separates S from $S_{\vec{v}}$. Then there is a child $\vec{v}'$ of $\vec{v}$ with $S_{\vec{v}'} = S'$ and $C_{\vec{v}'} \cap S \neq \emptyset$. Then $S \not\subseteq B_{\vec{v}}$, which is also a contradiction. ⌟

Claim 3: The torsi defined by Θ on G are minors of G.

Proof. This follows from the fact that if S is a hinge and C the vertex set of a connected component of $G \setminus S$, then $G[C \cup S] \cup K[S]$ is a minor of G. ☐

We state one last lemma about hinges that we will need in Section 4.4.

Lemma 4.9. *For every $k \geq 1$, if a graph G has a hinge of order k, then K_{k+1} is a minor of G.*

4.3. Decomposition into 3-connected components

For every $k \geq 1$, let $\mathscr{X}_k$ be the class of all k-connected graphs, and let $\mathscr{X}_k^*$ be the union of $\mathscr{X}_k$ with all complete graphs of order at most k. Note that $\mathscr{X}_1^*$ is the class of all connected graphs. For all classes of graphs, we let $\mathscr{X}_k^{(*)}(\mathscr{C}) := \mathscr{X}_k^{(*)} \cap \mathscr{C}$.

Theorem 4.10. *Let $\mathscr{C}$ be a class of graphs that is closed under taking minors. Then $\mathscr{C}$ admits an* IFP*-definable tree decomposition over $\mathscr{X}_3^*(\mathscr{C})$.*

Proof. By Lemma 3.10, $\mathscr{C}$ admits an IFP-definable tree decomposition over $\mathscr{X}_1^*(\mathscr{C})$. By Lemma 4.8 applied to $k = 1, 2$, the class $\mathscr{X}_k(\mathscr{C})$ admits an IFP-definable tree decomposition over $\mathscr{X}_{k+1}^*(\mathscr{C})$. Here we use the fact that all minimal separators of order 1 and 2 are hinges. We can combine the definable tree decompositions to a decomposition of $\mathscr{C}$ over $\mathscr{X}_3^*(\mathscr{C})$ by Lemma 3.10. ☐

Remark 4.11. For $k = 1, 2$, the proof of Lemma 4.8 actually gives stronger results than stated. For $k = 1$, all torsi defined by Θ are induced subgraphs of G, which are known as the *blocks* of G, and for $k = 2$, the torsi are topological minors of G, which are known as the *3-connected components* of G.

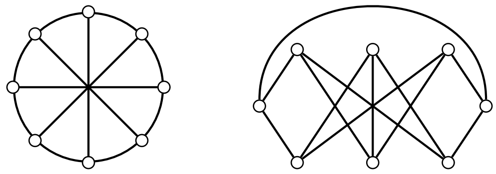

Figure 4.2. Two drawings of the graph L

Hence Theorem 4.10 can be strengthened to all classes $\mathscr{C}$ of graphs closed under topological minors. Furthermore, every class $\mathscr{C}$ of graphs closed under taking induced subgraphs admits an IFP-definable tree decomposition over $\mathscr{X}_2^*(\mathscr{C})$. ⌟

4.4. K_5-free graphs

Let L be the graph displayed (twice) in Figure 4.2. The second drawing of L in Figure 4.2 shows that $K_{3,3}$ is a minor of L. Hence L is not planar. However, L does not contain K_5 as a minor. (To see this, note that to obtain K_5 from L, we have to contract at least 5 edges to generate 5 vertices of degree 4. But then only 3 vertices remain.) Let $\mathscr{L} := \{G \mid G \cong H \text{ for some } H \subseteq L\}$, and let $\mathscr{P}$ denote the class of planar graphs. Let us call a K_5-free graph G *edge-maximal* if there is no K_5-free graph G' with $V(G') = V(G)$ and $E(G') \supset E(G)$.

Theorem 4.12 (Wagner [27]). *Let G be an edge-maximal K_5-free graph that has no hinges. Then either G is planar or $G \cong L$.*

It is an easy consequence of Wagner's theorem that every K_5-free graph has a tree decomposition over the class of planar graphs and subgraphs of L. We prove that there even is a definable tree decomposition:

Theorem 4.13. *The class of K_5-free graphs admits an* IFP*-definable tree decomposition over $\mathscr{X}_3^*(\mathscr{P} \cup \mathscr{L})$.*

Proof. It follows from Lemma 4.8 and Lemma 4.14 below that the class of 3-connected K_5-free graphs admits an IFP-definable tree decomposition over $\mathscr{X}_3^*(\mathscr{P} \cup \mathscr{L})$. Then the theorem follows from Theorem 4.10 and Lemma 3.9. ☐

Lemma 4.14. *Let G be a 3-connected K_5-free graph that has no hinges of order 3. Then either G is planar or G is isomorphic to a subgraph of L.*

Proof. Let $G' \supseteq G$ with $V(G') = V(G)$ be edge-maximal K_5-free. If G' is planar or $G' \cong L$, then G is planar or isomorphic to a subgraph of L. Otherwise, by Wagner's Theorem 4.12, G' has at least one hinge. By the edge-maximality of G', all hinges of G' are cliques.

We now define a sequence of induced subgraphs $G_1, \ldots, G_m$ of G' and hinges $S_1, \ldots, S_{m-1}$, where S_i is a hinge of G_i. We let $G_1 = G'$. Suppose we have defined

G_i. If G_i has no hinge or is planar, we let $m = i$. Otherwise, we let S_i be an arbitrary hinge of G_i and we let G_{i+1} be the connected component of G_i of maximum order. An easy induction shows that all G_i are 3-connected and edge-maximal K_5-free graphs and all the hinges S_i are hinges of G'. By Lemmas 4.9, $|S_i| \leq 3$. Then by Lemma 4.4, $G \setminus S_i$ has precisely two connected components, one of which consists of a single vertex v_i.

By Wagner's Theorem 4.12, either $G_m \cong L$ or G_m is planar. However, $G_m \cong L$ is impossible, because S_{m-1} induces a triangle in G_m. Hence G_m is planar. Let $i \in [m]$ be minimum such that G_i is planar. If $i = 1$, then G' is planar and hence G is also planar. So suppose that $i \geq 2$. We fix some planar embedding of G_i. Then the triangle S_{i-1} becomes a cycle in the plane. Since we cannot extend the embedding to G_{i-1}, both faces bounded by this cycle must be nonempty. Hence S_{i-1} separates G_i, and therefore, $G_{i-1} \setminus S_{i-1}$ has at least three connected components. Then $G \setminus S_i$ also has at least three connected components, which is a contradiction. □

Theorem 4.15 (Grohe [9]). *The class of 3-connected planar graphs is* IFP*-definable.*

Corollary 4.16. *The class of K_5-free graphs is* IFP*-definable.*

PART II. DESCRIPTIVE COMPLEXITY

5. More preliminaries

5.1. Structures and their Gaifman graphs

We work with two-sorted relational structures. Elements of the two sorts are called *vertices* and *numbers*. The vertex set of a structure A, denoted by $V(A)$, is an arbitrary finite set, and the number set of every structure is $\mathbb{Z}_{\geq 0}$. Relations may be mixed, but are always required to be finite. Vocabularies are finite sets of relation symbols, each with a prescribed sort. The interpretation of a relation symbol R is denoted by $R(A)$. Structures whose relations are entirely over the vertex part are called *plain*. We may identify the usual one sorted relational structures with plain structures in our framework. For example, we may view graphs as plain structures of vocabulary $\{E\}$. The class of all structures is denoted by $\mathscr{S}$, and the class of all plain structures by $\mathscr{S}_P$. If $\mathscr{C}$ is a class of structures and σ a vocabulary, then $\mathscr{C}[\sigma]$ is the class of all σ-structures in $\mathscr{C}$. Hence $\mathscr{G} \subseteq \mathscr{S}[\{E\}]$. An *ordered* structure is a structure whose vocabulary contains the distinguished relation symbol $\leqslant$ that is interpreted by a linear order of the vertex set. The class of all ordered structures is denoted by $\mathscr{O}$.

The *Gaifman graph* of a σ-structure A is the graph G_A with vertex set $V(G_A) := V(A)$ and edges between all pairs of vertices v, w such that there is a relation symbol $R \in \sigma$ and a tuple $\vec{a} \in R(A)$ such that $v, w \in \tilde{a}$. Note that the Gaifman graph only reflects the structure induced on the vertices and completely ignores the part of the relations on the number set. For every class $\mathscr{C}$ of graphs, $\mathscr{S}(\mathscr{C})$ denotes the class of all structures whose Gaifman graph is in $\mathscr{C}$.

5.2. Inflationary fixed-point logic with counting

To introduce the logic IFP+C, *inflationary fixed-point logic with counting*, we enhance IFP by a counting mechanism that allows us to define the cardinalities of definable sets by terms of the numerical sort. We allow mixed fixed-points over both sorts. To avoid undecidability, all variables ranging over numbers must be bound by terms when they are introduced. It is not hard to prove that our logic IFP+C has the same expressive power as the more common versions of fixed-point logic with counting that can be found in the literature (e.g. [6, 7, 21, 22, 23]).

6. Capturing polynomial time

A *Boolean query* is an isomorphism-closed class of σ-structures, for some plain vocabulary σ. Let us say that a logic L *captures* PTIME on a class $\mathscr{C}$ of structures if for every Boolean query $\mathscr{Q} \subseteq \mathscr{C}$, the query $\mathscr{Q}$ is definable in L if and only if it is decidable in PTIME. This definition is overly simplistic, as there are artificial logics capturing PTIME in this sense. However, the simple definition is sufficient for this paper, where we are only concerned with the logics IFP and IFP+C anyway. For more details, I refer the reader to one of the textbooks [5, 8, 17, 19] or to the short survey [11] in this volume.

Theorem 6.1 (Immerman-Vardi Theorem [16, 26]). IFP *captures* PTIME *on the class $\mathscr{O}$ of all ordered structures.*

6.1. Definable orders

We want to apply the Immerman-Vardi Theorem to structures that are not ordered. Sometimes this is easy because a linear order is *definable* on the structures we are considering. We say that a formula $\varphi(\vec{x}, y, z)$ of some logic *defines an order* on a class $\mathscr{C}$ of σ-structures (with parameters $\vec{x}$) if for every structure $A \in \mathscr{C}$ there is a tuple $\vec{v} \in V(A)^{|\vec{x}|}$ such that the binary relation $\varphi[A; \vec{v}, y, z]$ is a linear order on $V(A)$. We say that a class $\mathscr{C}$ of structures is IFP-*orderable* if for every vocabulary σ there is an IFP-formula that defines an order on $\mathscr{C}[\sigma]$. It is a straightforward consequence of the Immerman-Vardi Theorem that if a class $\mathscr{C}$ of structures is IFP-orderable then IFP captures PTIME on $\mathscr{C}$.

Theorem 6.2 (Grohe [9]). *The class of $\mathscr{Z}_3(\mathscr{P})$ of 3-connected planar graphs is* IFP*-orderable.*

Observe that if a class $\mathscr{C}$ of graphs is IFP-orderable, then the class $\mathscr{S}(\mathscr{C})$ is also IFP-orderable. Furthermore, if a class $\mathscr{C}$ of structures is IFP-orderable and $\mathscr{C}' \supseteq \mathscr{C}$ such that for every vocabulary σ the difference $\mathscr{C}'[\sigma] \setminus \mathscr{C}[\sigma]$ is finite up to isomorphism, then $\mathscr{C}'$ is also IFP-orderable.

6.2. Definable canonisation

We call variables ranging over elements of the numerical sort of our structures *number variables*.

Definition 6.3. Let σ and τ be vocabularies.

(1) A *numerical interpretation of σ in τ with parameters* $\vec{x}$ is a tuple

$$\Gamma(\vec{x}) = \big(\gamma_a(\vec{x}), \gamma_V(\vec{x},y), \big(\gamma_R(\vec{x},\vec{y}_R)\big)_{R\in\sigma}\big)$$

of IFP+C$[\tau]$-formulas, where y is a number variable $\vec{y}_R$ is a tuple of number variables whose length matches the arity of R, for each $R \in \sigma$.

In the following, let $\Gamma(\vec{x})$ be an interpretation of σ in τ. Furthermore, let A be a τ-structure, and let $\vec{a}$ be a tuple of elements of $V(A)$.

(2) $\Gamma(\vec{x})$ is *applicable* to $(A,\vec{a})$ if $\vec{a}$ has the same sort as $\vec{x}$ and $A \models \gamma_a[\vec{a}]$.

(3) If $\Gamma(\vec{x})$ is applicable to $(A,\vec{a})$, we let $\Gamma[A;\vec{a}]$ be the σ-structure with vertex set $V\big(\Gamma[A;\vec{a}]\big) := \gamma_V[A;\vec{a},y]$ and relations $R\big(\Gamma[A;\vec{a}]\big) := \gamma_R[A;\vec{a},\vec{y}_R]$, for $R \in \sigma$. ⌟

It is easy to prove that if $\Gamma(\vec{x})$ is a numerical interpretation of σ in τ, then for every IFP+C$[\sigma]$-sentence φ there is an IFP+C$[\tau]$-sentence φ' such that for all τ-structures A and all tuples $\vec{a}$ such that $\Gamma(\vec{x})$ is applicable to $(A,\vec{a})$ we have $A \models \varphi' \iff \Gamma[A;\vec{a}] \models \varphi$.

Definition 6.4. Let $\Gamma(x_1,\ldots,x_k)$ be a numerical interpretation of σ in σ.

(1) $\Gamma(\bar{x})$ *canonises* a σ-structure A if there is a tuple $\vec{a}$ over $V(A)$ such that $\Gamma(\vec{x})$ is applicable to $(A,\vec{a})$, and if for each such tuple $\vec{a}$ it holds that $\Gamma[A,\vec{a}] \cong A$.

(2) $\Gamma(\vec{x})$ *canonises* a class $\mathscr{C}$ of σ-structures if $\Gamma(\vec{x})$ canonises each structure in $\mathscr{C}$.

(3) A class $\mathscr{C}$ of structures is IFP+C-*canonisable* if for every vocabulary σ there is a numerical IFP+C-interpretation $\Gamma(\vec{x})$ that canonises $\mathscr{C}[\sigma]$. ⌟

The following lemma is used in [9, 12], and it is implicit in earlier work such as [23] and [18]:

Lemma 6.5. *Let $\mathscr{C}$ be a class of structures that is* IFP+C-*canonisable. Then* IFP+C *captures* PTIME *on $\mathscr{C}$.*

7. Canonisation of decomposable structures

The main result of the second part of this paper states that we can lift an IFP+C-canonisation from the torsi of a definable tree decomposition to the whole decomposition. We need one additional technical restriction on the tree decompositions:

Definition 7.1. A class $\mathscr{C}$ of graphs admits an IFP-definable tree decomposition over a class $\mathscr{A}$ of graphs *of bounded adhesion* if there is a TD-scheme Θ and a $k \in \mathbb{Z}_{\geq 0}$ such that for every $G \in \mathscr{C}$ the scheme Θ defines a tree decomposition of adhesion at most k on G over $\mathscr{A}$. ⌟

Observe that if $\mathscr{A}$ is a class of graphs G of clique number at most k, then the adhesion of all tree definable decompositions over $\mathscr{A}$ is at most k. All classes $\mathscr{A}$ we consider here are of bounded clique number, hence tree decompositions over these classes are automatically of bounded adhesion.

Theorem 7.2 (Second Lifting Theorem). *Let $\mathscr{C},\mathscr{A}$ be classes of graphs such that $\mathscr{C}$ admits an* IFP+C-*definable tree decomposition over $\mathscr{A}$ of bounded adhesion. If $\mathscr{S}(\mathscr{A})$ is* IFP+C-*canonisable, then $\mathscr{S}(\mathscr{C})$ is* IFP+C-*canonisable.*

Remark 7.3. The use of arbitrary two-sorted structures with mixed relations, and not just plain structures, is crucial for our proof of Theorem 7.2. Essentially, mixed relations come in as follows: We canonise the decomposable structures inductively. So suppose we have a decomposition of the Gaifman graph G_A of a structure A defined by a TD-scheme Θ, and suppose that we have already canonised the substructures $A[B^{\Theta}_{\geq\vec{v}'}]$ for all children of a Θ-node $\vec{v}$. Note that a Θ-node may have an unbounded number of Θ-children. We can lexicographically sort the canonical structures for the children. Then we expand the structure $A[B_{\vec{v}}]$ by a mixed relation that contains tuples $\vec{v}\ell_{\vec{v}}$, where $\vec{v}$ is a Θ-node and $\ell_{\vec{v}} \in \mathbb{Z}_{\geq 0}$ such that for all $\vec{v},\vec{v}'$ it holds that $\ell_{\vec{v}} \leq \ell_{\vec{v}'}$ if and only if canonical copy of the structure $A[B^{\Theta}_{\geq\vec{v}}]$ is lexicographically smaller than the canonical copy of the structure $A[B^{\Theta}_{\geq\vec{v}'}]$. Then we canonise the expansion $A[B_{\vec{v}}]$, and this will give us a canonisation of $A[B_{\geq\vec{v}}]$. We can sort canonical structures because their universes are subsets of $\mathbb{Z}_{\geq 0}$.

It is an open problem whether the version of Theorem 7.2 for plain structures holds, that is, whether the IFP+C-canonisability of $\mathscr{S}_P(\mathscr{A})$ implies that of $\mathscr{S}_P(\mathscr{C})$.

It is actually conceivable that for every class $\mathscr{C}$ of graphs the IFP+C-canonisability of $\mathscr{S}_P(\mathscr{C})$ implies the IFP+C-canonisability of $\mathscr{S}(\mathscr{C})$ and hence that the two notions are equivalent. Of course a proof of this would be the nicest way to settle the questions. ⌟

Corollary 7.4. IFP+C *captures polynomial time on the class of K_5-free graphs.*

Let me close this paper by observing that our techniques actually prove slightly more general results than stated so far. For a class $\mathscr{C}$ of graphs, let $\mathscr{R}(\mathscr{C})$ be the class of all structures A whose vocabulary contains the symbol $\leqslant$ such that: $\leqslant^A$ is a linear order of a subset $W \subseteq V(A)$, and the induced subgraph $G_A \setminus W$ is in $\mathscr{C}$. It is not hard to prove that if the class $\mathscr{S}(\mathscr{C})$ is IFP-canonisable, then so is the class $\mathscr{R}(\mathscr{C})$. This gives us the following corollary:

Corollary 7.5. *Let $\mathscr{C}$ either be a class of graphs of bounded tree width or the class of all K_5-free graphs. Then* IFP+C *captures* PTIME *on* $\mathscr{R}(\mathscr{C})$.

8. Concluding remarks

We have introduced definable tree decompositions and proved two general theorems about lifting definability results from the torsi of a definable tree decomposition to the whole decomposed structure. As an application of these general theorems, we prove that the class of K_5-free graphs is definable in IFP and canonisable in IFP+C; the latter implies that IFP+C captures polynomial time on this class.

The ultimate goal of this line of research is to prove that all (nontrivial) classes of graphs defined by excluded minors are definable in IFP and IFP+C-canonisable. The present paper is a significant step towards this goal, because via Robertson and Seymour's structure theorem [25] for classes of graphs with excluded minors it enables us to reduce the definability questions to graphs that are almost embeddable into a fixed surface (in a precise technical sense).

In this paper, we were only interested in fixed-point definability, but definable tree decompositions could be studied for other logics as well, in particular for monadic second-order logic MSO. This may shed some new light on the relation between fixed-point definability and MSO-definability.

References

[1] L. Babai and E. Luks. Canonical labeling of graphs. In *Proceedings of the 15th ACM Symposium on Theory of Computing*, pages 171–183, 1983.

[2] H. Bodlaender. Polynomial algorithms for graph isomorphism and chromatic index on partial *k*-trees. *Journal of Algorithms*, 11:631–643, 1990.

[3] J. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12:389–410, 1992.

[4] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982.

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

[6] J. Flum and M. Grohe. On fixed-point logic with counting. *Journal of Symbolic Logic*, 65:777–787, 2000.

[7] E. Grädel and M. Otto. Inductive definability with counting on finite structures. In E. Börger, G. Jäger, H. K. Büning, S. Martini, and M. Richter, editors, *Computer Science Logic, 6th Workshop, CSL '92, San Miniato 1992, Selected Papers*, volume 702 of *Lecture Notes in Computer Science*, pages 231–247. Springer-Verlag, 1993.

[8] E. Grädel, P. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications*. Springer-Verlag, 2007.

[9] M. Grohe. Fixed-point logics on planar graphs. In *Proceedings of the 13th IEEE Symposium on Logic in Computer Science*, pages 6–15, 1998.

[10] M. Grohe. Isomorphism testing for embeddable graphs through definability. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 63–72, 2000.

[11] M. Grohe. The quest for a logic capturing PTIME. In *Proceedings of the 23rd IEEE Symposium on Logic in Computer Science*, 2008. These proceedings.

[12] M. Grohe and J. Mariño. Definability and descriptive complexity on databases of bounded tree-width. In C. Beeri and P. Buneman, editors, *Proceedings of the 7th International Conference on Database Theory*, volume 1540 of *Lecture Notes in Computer Science*, pages 70–82. Springer-Verlag, 1999.

[13] Y. Gurevich. Logic and the challenge of computer science. In E. Börger, editor, *Current trends in theoretical computer science*, pages 1–57. Computer Science Press, 1988.

[14] J. Hopcroft and J. Wong. Linear time algorithm for isomorphism of planar graphs. In *Proceedings of the 6th ACM Symposium on Theory of Computing*, pages 172–184, 1974.

[15] J. E. Hopcroft and R. Tarjan. Isomorphism of planar graphs (working paper). In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*. Plenum Press, 1972.

[16] N. Immerman. Upper and lower bounds for first-order expressibility. *Journal of Computer and System Sciences*, 25:76–98, 1982.

[17] N. Immerman. *Descriptive Complexity*. Springer-Verlag, 1999.

[18] N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. In A. Selman, editor, *Complexity theory retrospective*, pages 59–81. Springer-Verlag, 1990.

[19] L. Libkin. *Elements of Finite Model Theory*. Springer-Verlag, 2004.

[20] E. Luks. Isomorphism of graphs of bounded valance can be tested in polynomial time. *Journal of Computer and System Sciences*, 25:42–65, 1982.

[21] M. Otto. Ptime canonization for two variables with counting. In *Proceedings of the 10th IEEE Symposium on Logic in Computer Science*, pages 342–352, 1995.

[22] M. Otto. The expressive power of fixed-point logic with counting. *Journal of Symbolic Logic*, 61:147–176, 1996.

[23] M. Otto. *Bounded variable logics and counting – A study in finite models*, volume 9. Springer-Verlag, 1997.

[24] I. N. Ponomarenko. The isomorphism problem for classes of graphs that are invariant with respect to contraction. *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)*, 174(Teor. Slozhn. Vychisl. 3):147–177, 182, 1988. In Russian.

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

[26] M. Vardi. The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on Theory of Computing*, pages 137–146, 1982.

[27] K. Wagner. Beweis einer Abschwächung der Hadwiger-Vermutung. *Mathematische Annalen*, 153:139–141, 1964.