Fixed-Point Definability and Polynomial Time on Graphs with Excluded Minors

MARTIN GROHE, Humboldt University Berlin

We give a logical characterization of the polynomial-time properties of graphs embeddable in some surface. For every surface S, a property $\mathcal{P}$ of graphs embeddable in S is decidable in polynomial time if and only if it is definable in fixed-point logic with counting. It is a consequence of this result that for every surface S there is a k such that a simple combinatorial algorithm, namely "the k-dimensional Weisfeiler-Lehman algorithm", decides isomorphism of graphs embeddable in S in polynomial time.

We also present (without proof) generalizations of these results to arbitrary classes of graphs with excluded minors.

Categories and Subject Descriptors: F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Model theory*; F.1.3 [**Computation by Abstract Devices**]: Complexity Measures and Classes; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms: Theory, Algorithms

Additional Key Words and Phrases: Descriptive complexity, fixed-point logics, graph minors, graph isomorphism

ACM Reference Format:
Grohe, M. 2012. Fixed-Point definability and polynomial time on graphs with excluded minors. J. ACM 59, 5, Article 27 (October 2012), 64 pages.
DOI = 10.1145/2371656.2371662 http://doi.acm.org/10.1145/2371656.2371662

27

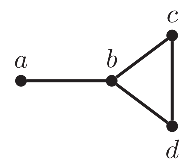
1. INTRODUCTION

There is a deep and fruitful connection between the descriptive complexity and the computational complexity of computational problems. Here descriptive complexity refers to the "language resources" required to express a problem in a formal language, or logic, whereas computational complexity refers to the computational resources needed to solve the problem. The connection goes back to the beginnings of computability theory, but descriptive complexity theory as we know it today started with Fagin's Theorem [Fagin 1974]. It states that a property of finite graphs (or other structures) is expressible in existential second-order logic if and only if it is decidable in nondeterministic polynomial time. More concisely, we say that existential second-order logic *captures* NP. Similar logical characterizations have been obtained for other complexity classes such as the polynomial-time hierarchy PH and PSPACE. However, no logic is known to capture the class PTIME or any natural complexity class below PTIME.

The question of whether there is a logic that captures PTIME has first been raised by Chandra and Harel [1982] in a fundamental paper on database query languages. Chandra and Harel asked for a query language for relational databases expressing precisely those queries that can be evaluated in polynomial time. Clearly, it would be very nice to have such a language, because it would allow the user to ask only

The author is currently affiliated with RWTH Aachen University.
Author's address: RWTH Aachen University, Aachen, Germany; email: grohe@informatik.rwth-aachen.de.

DOI 10.1145/2371656.2371662 http://doi.acm.org/10.1145/2371656.2371662



(a)

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

(b)

0100 1011 0101 0110
0010 0011 1101 0110
0011 0010 1101 1010

(c)

Fig. 1.1. Different adjacency matrices of a graph.

queries that can be answered efficiently, while at the same time guaranteeing that all such queries can be asked. Gurevich [1984] rephrased the question in terms of logic. He conjectured that there is no logic capturing PTIME. Observe that this conjecture implies PTIME $\neq$ NP, because by Fagin's Theorem there is a logic that captures NP. The question of whether there is a logic capturing PTIME is still open today, and it is viewed as one of the main open problems in finite model theory and database theory.

So why are we interested in a logic capturing PTIME? After all, logic is just another formalism that can be used to describe computation, like Turing machines, the lambda calculus, various grammars and rewriting systems, or "real" programming languages. But there is an important difference between logics capturing complexity classes and the other formalisms. The models in which we describe computational problems are usually finite structures such as graphs, transition systems, Boolean circuits, or relational databases. Logics can speak directly about such structures, whereas most other formalisms operate on encodings of structures by strings or terms. Hence a logical characterization of a complexity class is representation independent. This may seem like a minor technical issue, but actually is quite significant, mainly because we have no natural canonical representations of structures by strings or terms, but have to make arbitrary choices when we construct such representations.

Example 1.1. A standard way of representing graphs is by their adjacency matrices; once we have an adjacency matrix we can obtain a $\{0, 1\}$-string encoding the graph by concatenating the rows of the matrix.

Consider the graph displayed in Figure 1.1(a). Figure 1.1(b) shows three different adjacency matrices of this graph, leading to three different string encodings shown in Figure 1.1(c). Indeed, the graph has $4! = 24$ adjacency matrices, one for each linear order of the vertices (the three matrices in Figure 1.1 correspond to the orders *abcd*, *acbd*, *cabd*). Not all 24 matrices are different, but there is no distinguished one that we may declare to be the "right" representation.

We are used to thinking of graphs as abstract models and of properties of graphs as only depending on the graphs and not on their specific representations. Examples of graph properties are "connectedness", "acyclicity", or "Hamiltonicity" (i.e., having a cycle that traverses each vertex exactly once). By comparison, we would not view a statement like "the first vertex has degree 3" as describing a graph property, because a graph has no distinguished "first" vertex.

This level of abstraction is an integral part of our models. For example, if we use a graph model of a network then we abstract from the physical location of the nodes on purpose (otherwise we should use a different model). Similarly, it is a feature and not a bug of the relational database model to separate the logical from the physical level of a database.

There is a mismatch between these abstract models and the typical algorithms working on them. A graph algorithm does not get an abstract graph as input, but a specific representation of this graph, typically an adjacency matrix or adjacency lists. Nevertheless, its result is expected to be independent of the specific representation. But how can we guarantee that a particular graph algorithm is representation independent? Unfortunately, it is not decidable whether an algorithm has this property. As a matter of fact, many of the standard graph algorithms we know internally depend on the specific representation of the input and only happen to be representation independent in their output. A typical example is a connectivity algorithm based on depth-first search. At any vertex of the graph, a depth-first search chooses the "first" neighbor of the vertex to continue the search. But there is no canonical first neighbor; it depends on the specific representation of the input graph which neighbor comes first.

We may ask if there is a programming language that enforces representation independence, in the sense that all syntactically correct programs are representation independent, and that at the same time allows us to implement algorithms for all decidable properties of graphs. It is not too hard to come up with such a language, albeit not a very natural one. But the price to pay appears to be a huge loss in efficiency. The question of whether there is a logic capturing PTIME can be recast as asking for a programming language that enforces representation independence and at the same time allows us to implement polynomial-time algorithms for all properties of graphs that can be decided in polynomial time by conventional algorithms working on specific representations of graphs. (By polynomial-time algorithms in our new language we mean programs that can be executed in polynomial time on a conventional machine.)

It is worth mentioning that the question of whether there is a logic capturing PTIME is an instance of the general theme of finding syntactical characterizations of semantically defined complexity classes. Here the semantically defined complexity class consists of all properties of finite structures that are decidable in polynomial time. Other prominent examples of semantically defined complexity classes for which we do not know syntactical characterizations are NP $\cap$ co-NP and BPP.

As mentioned earlier, the question of whether there is a logic capturing PTIME is still open, but there are partial positive results for specific classes of structures. The main theorem of this article is such a result. Informally, a logic L *captures* PTIME *on a class* $\mathcal{C}$ of structures if precisely the polynomial-time properties of structures in $\mathcal{C}$ are definable in L (see Appendix A for a technical definition).

A *structure* consists of a universe, which is simply a finite set, and a finite collection of relations, functions, and constants on this universe.[1] An ordered structure is a structure that has one distinguished binary relation $\leqslant$ which is a linear order of the universe. In 1982, Immerman [1982] and Vardi [1982], independently, proved that least fixed-point logic LFP captures polynomial time on the class of all ordered structures. LFP is an extension of first-order predicate logic by a fixed-point operator that allows

[1]In general, structures may also have an infinite universe and infinitely many relations, functions, and constants, but all structures considered in this article are finite.

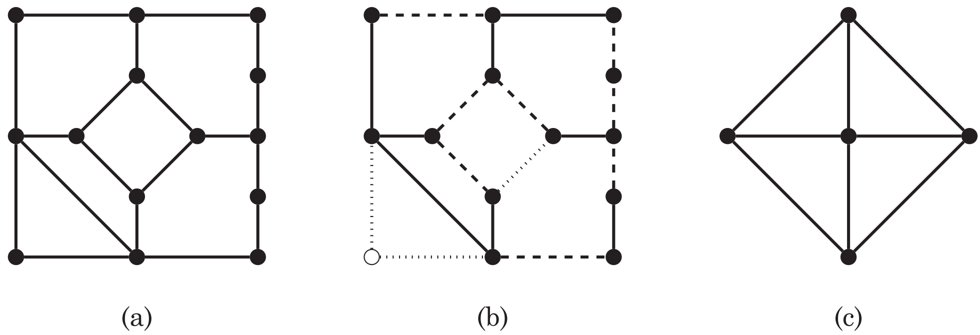


Fig. 1.2. The graph in (c) is a minor of the graph in (a) obtained by deleting the dotted edges and the white node and contracting the dashed edges in (b).

it to formalize inductive definitions. In this article, it will be more convenient to work with inflationary fixed-point logic IFP, which is known to have the same expressive power as LFP, even on infinite structures [Gurevich and Shelah 1986; Kreutzer 2002].

It can be shown by standard techniques that IFP does not capture polynomial time on all structures, the most blatant reason being that the logic lacks the ability to count. For example, the graph property of having an even number of vertices is not definable in IFP (see, for example, Immerman [1999]). In the late 1980s, Immerman [1987] conjectured that inflationary fixed-point logic with counting IFP+C, an extension of inflationary fixed-point logic by a mechanism that allows it to define the cardinalities of definable sets, might capture polynomial time. Although in 1992 Cai, Fürer, and Immerman [1992] refuted this conjecture, it has turned out since then that IFP+C does capture polynomial time on many interesting classes of structures. As a matter of fact, Hella, Kolaitis, and Luosto [1996] proved that IFP+C captures polynomial time on almost all structures (in a precise technical sense).

The first capturing result for a specific class of graphs is due to Immerman and Lander [1990]; it states that IFP+C captures PTIME on the class of trees. In this article, we prove the following theorem.

THEOREM 1.2. *For every surface* $\boldsymbol{S}$*,* IFP+C *captures* PTIME *on the class of all graphs embeddable in* $\boldsymbol{S}$.

The special case of this result for planar graphs first appeared in the conference paper [Grohe 1998]. The result for general surfaces has not been published before, but a slightly weaker result appeared in the conference paper [Grohe 2000].

A consequence of Theorem 1.2 is a simple polynomial-time isomorphism test for all classes of graphs embeddable in a fixed surface. For each surface $\boldsymbol{S}$, a generic combinatorial "colour refinement" algorithm, known as the k-dimensional Weisfeiler-Lehman algorithm, decides isomorphism of graphs embeddable in $\boldsymbol{S}$. Remarkably, the algorithm does not build on any of the special graph-theoretic properties the graphs embeddable in $\boldsymbol{S}$ may have, only the parameter k depends on $\boldsymbol{S}$. Neither does the algorithm use logic in any way; logic is only used to prove that it works. We describe the algorithm in Section 9, which can be read independently of the rest of the article.

Actually, Theorem 1.2 is only a special case of a far more general theorem. To state it, we need to define graph minors. A graph H is a *minor* of a graph G if H is isomorphic to a graph obtained from a subgraph of G by contracting edges (see Figure 1.2 for an example). We say that H is an *excluded minor* for a class $\mathcal{C}$ of graphs if H is not a minor of any graph in $\mathcal{C}$.

THEOREM 1.3 [GROHE 2012]. *Let $\mathcal{C}$ be a class of graphs with at least one excluded minor. Then* IFP+C *captures* PTIME *on $\mathcal{C}$.*

For every surface $\boldsymbol{S}$, the class of all graphs embeddable in $\boldsymbol{S}$ excludes some graph as a minor. Indeed, every complete graph K_k such that $(k-3)(k-4)/6$ is greater than the genus of S is an excluded minor (Ringel and Youngs, see Ringel [1974] or Mohar and Thomassen [2001]). Hence Theorem 1.3 is indeed a generalization of Theorem 1.2. Other examples of classes with excluded minors are the class of all graphs that can be embedded in $\mathbb{R}^3$ such that no cycle is nontrivially knotted (K_7 is an excluded minor [Conway and Gordon 1983]), for every k the class of all graphs that have a vertex cover of size at most k (K_{k+2} is an excluded minor), for every k the class of all graphs that have a feedback vertex set of size at most k (K_{k+3} is an excluded minor), and for every k the class of all graphs of tree width at most k (K_{k+2} is an excluded minor). It was known before that IFP+C captures PTIME on all classes of bounded tree width [Grohe and Mariño 1999]. To show the limitations of our theorem, let us also mention that the class of all cubic graphs (that is, graphs in which every vertex has exactly three neighbors) does not exclude any graph as a minor. While it is known that IFP+C does not capture PTIME on the class of cubic graphs [Cai et al. 1992], a natural example of a class of graphs that does not exclude a minor and on which IFP+C captures PTIME is the class of interval graphs [Laubner 2010].

It is a consequence of Theorem 1.3 that for every class $\mathcal{C}$ of graphs with at least one excluded minor there is a k such that the k-dimensional Weisfeiler-Lehman algorithm decides isomorphism of graphs embeddable in $\mathcal{C}$.

In this article, we only prove Theorem 1.2. Because of its length, it is not feasible to publish a full proof of Theorem 1.3 in a journal article. However, Theorem 1.2, or rather the Definable Structure Theorem for Embeddable Graphs 7.1 underlying Theorem 1.2, is an important building block of the proof of Theorem 1.3. Its proof already contains many of the key ideas of the proof of Theorem 1.3. We will give a high-level outline of the proof of Theorem 1.3 in Section 8. The full proof will be published in a monograph [Grohe 2012].

To prove Theorems 1.2 and 1.3, we develop a definable structure theory for graphs embeddable in a surface and graphs with excluded minors. It builds on the structure theory for graphs with excluded minors developed by Robertson and Seymour (see Section 8.1), but ultimately leads to quite different structural decompositions of the graphs. On the graph theoretic side, an important new aspect of our theory is that we need to identify structural properties of graphs with excluded minors that are invariant under automorphisms, which is a prerequisite for making them definable. On the logical side, we have to find a notion of definable decomposition that is flexible enough to be widely applicable, yet still carries enough information to enable us to prove Theorems 1.2 and 1.3. Definable (ordered) treelike decompositions, which will be introduced in Section 3, fulfil these requirements. Much of the definable structure theory we develop is fairly general and not tailored towards graphs with excluded minors, and it may find other applications.

This article is based on the conference papers [Grohe 1998, 2000, 2008, 2010]. A full proof of Theorem 1.3 can be found in [Grohe 2012]. A nontechnical presentation of the results can be found in Grohe [2011]. The proof of Theorem 1.2 we present here is self-contained, though some details are deferred to an appendix (available on the article's Web site in the ACM Digital Library). The proof presented here is streamlined compared to the proof presented in the monograph [Grohe 2012], which more extensively relies on the general theory developed to prove Theorem 1.3.

The rest of this article is organized as follows. In Section 2 we give the necessary preliminaries from graph theory and descriptive complexity theory. Then in Section 3, we introduce treelike decompositions and ordered treelike decompositions and develop their basic theory. Most of the proofs of the results presented in this section are long and technical, yet not very difficult. We only sketch them in the main article and defer details to the appendix. Section 4–7 are devoted to a proof of Theorem 1.2. Section 4 introduces the necessary background from surface topology and topological graph theory. In Section 5 we prove two technical lemmas, which we call "angle lemmas". In Section 6, we prove Theorem 1.2 for planar graphs. Finally, the full theorem is proved in Section 7. In Section 8, we give an outline of the proof of Theorem 1.3, and in Section 9 we discuss the application of our theorems to graph isomorphism testing.

2. BACKGROUND ON GRAPH THEORY AND DESCRIPTIVE COMPLEXITY

We briefly review basic notions and a few results from graph theory and descriptive complexity theory and at the same time introduce our notation.

$\mathbb{N}$ and $\mathbb{N}^+$ denote the sets of nonnegative integers and positive integers, respectively. For $m, n \in \mathbb{N}$, we let $[m, n] := \{\ell \in \mathbb{N} \mid m \leq \ell \leq n\}$ and $[n] := [1, n]$. We denote the power set of a set S by 2^S and the set of all k-element subsets of S by $\binom{S}{k}$.

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

2.1. Graphs

Graphs in this article are always finite, nonempty, and simple, where simple means that there are no loops or parallel edges. Unless explicitly called "directed", graphs are undirected. We view (directed) graphs G as relational structures with a vertex set $V(G)$ and a binary edge relation $E(G)$. For brevity we denote edges by vw instead of (v, w); in an undirected graph vw stands for both (v, w) and (w, v). 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 $G[V(G) \setminus W]$. It will often be convenient to let $G \setminus H := G \setminus V(H)$ for a graph H. The set $\{w \in V(G) \mid vw \in E(G)\}$ of *neighbors* of a node v is denoted by $N^G(v)$, or just $N(v)$ if G is clear from the context. For a subset $W \subseteq V(G)$ we let $N^G(W) := \{v \in V(G) \setminus W \mid \exists w \in W : \ vw \in E(G)\} = \big(\bigcup_{v \in W} N^G(v)\big) \setminus W$. For a subgraph H we let $N^G(H) := N^G(V(H))$. The *degree* of a vertex $v \in V(G)$ is the number $\deg^G(v) := |N(v)|$. A *branch vertex* of G is a vertex v with $\deg^G(v) \geq 3$. The *order* of a graph, denoted by $|G|$, is the number of vertices of G. A *homomorphism* from a graph G to a graph H is a mapping $h : V(G) \to V(H)$ that preserves adjacency, and an *isomorphism* is a bijective homomorphism whose inverse is also a homomorphism. We write $G \cong H$ to denote that G and H are isomorphic. An *automorphism* of a graph G is an isomorphism from G to itself. A *property* of graphs is a class of graphs that is closed under isomorphism.

For every finite nonempty set V, we let $K[V]$ be the *complete graph* with vertex set V, and we let $K_n := K\big[[n]\big]$. A *clique* in a graph G is a set $W \subseteq V(G)$ such that $G[W]$ is a complete graph. *Paths*, *cycles*, *trees*, and *forests* are defined in the usual way. The *length* of a path or cycle is the number of its edges. The *endvertices* and *internal vertices* of a path are defined in the obvious way. If P is a path and $v, w \in V(P)$, then vPw denotes the segment of P with endvertices v, w. A path or cycle *in* a graph G is a subgraph of G that is a path or cycle. An *isolated path* in G is a path $P \subseteq G$ such that no internal vertices of P are branch vertices of G. A *walk* in G is a sequence $W := v_0 e_1 v_1 e_2 \dots e_n v_n$, where $n \in \mathbb{N}$, $v_0, \dots, v_n \in V(G)$, $e_1, \dots, e_n \in E(G)$ such that $e_i = v_{i-1} v_i$ for all $i \in [n]$. The walk W is a *simple walk* if the vertices $v_0, \dots, v_n$ are pairwise distinct. W is a *closed* *walk* if $n \geq 1$ and $v_0 = v_n$, and a *simple closed walk* if it is a closed walk and $v_1, \ldots, v_n$ are pairwise distinct. With each path in G we can associate two walks, one for each direction, and with each cycle of length n in G we can associate $2n$ simple closed walks, one in each direction for each starting vertex.

Connectedness and *connected* components are defined in the usual way. The empty graph is disconnected by definition. A set $W \subseteq V(G)$ is *connected* in a graph G if $G[W]$ is connected. For sets $W_1, W_2 \subseteq V(G)$, a set $S \subset V(G)$ *separates* W_1 *from* W_2 if there is no path from a vertex in $W_1 \backslash S$ to a vertex in $W_2 \backslash S$ in the graph $G \backslash S$.

Recall that a graph G is a *minor* of a graph H if G can be obtained from a subgraph of H by contracting edges. *Contracting* an edge means deleting the edge and identifying its endvertices. Equivalently, and more formally, G is a minor of H if there is a mapping $f : V(G) \to 2^{V(H)}$ such that the sets $f(v)$, for $v \in V(G)$, are mutually disjoint connected subsets of $V(H)$ such that for all $vw \in E(G)$ there is an edge $v'w' \in E(H)$ with $v' \in f(v)$ and $w' \in f(H)$. To see the equivalence of the two definitions, note that if there is such a mapping f then G can be obtained from H by deleting all vertices in $V(H) \backslash \bigcup_{v \in V(G)} f(v)$ and deleting all edges between sets $f(v)$ and $f(w)$ for $vw \notin E(G)$, and by contracting all edges that have both endvertices in the same set $f(v)$ in the resulting subgraph. We write $G \preceq H$ to denote that G is a minor of H. A class $\mathcal{C}$ of a graphs is *closed under taking minors* if for every graph $H \in \mathcal{C}$ and every minor $G \preceq H$ we have $G \in \mathcal{C}$.

We use standard graph-theoretic terminology for directed graphs (for short: *digraphs*), without going through it in detail. Homomorphisms and isomorphisms of digraphs preserve the direction of the edges. Paths and cycles in a directed graph are always meant to be directed; otherwise we will call them "paths or cycles of the underlying undirected graph". We allow digraphs to have loops. Hence cycles in digraphs may have length 1 or 2. For a digraph D, we let $\trianglelefteq^D$ be the reflexive transitive closure of the edge relation $E(D)$ and $\triangleleft^D$ its irreflexive version. If D is acyclic then $\trianglelefteq^D$ is a partial order. By $N_+^D(v)$ we denote the set of "outgoing" neighbors of a vertex $v \in V(D)$. *Directed acyclic graphs* will be of particular importance in this article, and we introduce some additional terminology for them. Let D be a directed acyclic graph. A node w is a *child* of a node v, and v is a *parent* of w, if $vw \in E(D)$. A node v is a *leaf* if it has no outgoing neighbors and a *root* if it has no incoming neighbors. A *directed tree* is a directed acyclic graph T in which every node has at most one parent, and for which there is a vertex r called the *root* such that for all $t \in V(T)$ there is a path from r to t. The *height* of T is the maximum length of a path from the root to a leaf. A *subtree* of a directed tree is a subgraph that is a directed tree.

A *tree decomposition* of a graph G is a pair $\Delta = (T^\Delta, \beta^\Delta)$, where T^Δ is a tree and β^Δ a mapping that associates a set $\beta^\Delta(t) \subseteq V(G)$ of vertices of G with every node $t \in V(T)$ such that for every $v \in V(G)$ the set $\{t \in V(T) \mid v \in \beta^\Delta(t)\}$ induces a subtree of T, and for every $e \in E(G)$ there is a $t \in V(T)$ such that $e \subseteq \beta^\Delta(t)$. The sets $\beta^\Delta(t)$ are called the *bags* of the decomposition. It will be convenient for us to view the tree in a tree decomposition as being directed. The *torso* of Δ at a node $t \in V(T)$ is the graph

$$\tau^\Delta(t) := G[\beta^\Delta(t)] \cup K[\beta^\Delta(s) \cap \beta^\Delta(t)] \cup \bigcup_{u \in N_+^T(t)} K[\beta^\Delta(u) \cap \beta^\Delta(t)],$$

where s is the parent of t in T. If t is the root of T, the term $K[\beta^\Delta(s) \cap \beta^\Delta(t)]$ is omitted. The tree decomposition Δ is *over* a class $\mathcal{A}$ of graphs if all its torsos are in $\mathcal{A}$.

2.2. Relational Structures

A *vocabulary* is a finite set of *relation symbols*, each equipped with an *arity* in $\mathbb{N}$. Let τ be a vocabulary. A τ*-structure* A consists of a finite set $V(A)$ called the *universe* or *vertex set* of A and for each $R \in \tau$ a relation $R(A)$ over $V(A)$ of the appropriate arity. A *substructure* of a τ-structure A is a τ-structure B with $V(B) \subseteq V(A)$ and $R(B) \subseteq R(A)$ for all $R \in \tau$. An *induced substructure* of A is a substructure B with $R(B) = R(A) \cap V(B)^k$ for all k-ary $R \in \tau$. A homomorphism from a τ-structure A to a τ-structure B is a mapping that preserves membership in all relations. Then isomorphisms and automorphisms are defined as for graphs. A *property* of τ-structures is a class of τ-structures closed under isomorphism.

Throughout this article, we let E and $\leqslant$ be binary relation symbols. Instead of $\leqslant(A)$, we write $\leqslant^A$ to denote the interpretation of $\leqslant$ in a structure A. An *ordered structure* is a structure A whose vocabulary contains $\leqslant$ such that $\leqslant^A$ is a linear order of $V(A)$.

With few exceptions, the structures that will appear in this article are *(directed) graphs*, viewed as $\{E\}$-structures $G = \big(V(G), E(G)\big)$, and *ordered (directed) graphs*, viewed as $\{E, \leqslant\}$-structures $G = \big(V(G), E(G), \leqslant^G\big)$ where $(V(G), E(G))$ is a (directed) graph and $\leqslant^G$ is a linear order of the vertex set $V(G)$.

2.3. Logics

In this section, we will informally introduce the two main logics IFP and IFP+C used in this article. For background and a precise definition, I refer the reader to one of the textbooks [Ebbinghaus and Flum 1999; Grädel et al. 2007; Immerman 1999; Libkin 2004]. It will be convenient to start by briefly reviewing *first-order logic* FO. Formulae of first-order logic in the vocabulary of graphs are built from atomic formulae $E(x, y)$ and $x = y$, expressing adjacency and equality of vertices, by the usual Boolean connectives and existential and universal quantifiers ranging over the vertices of a graph. First-order formulae in the vocabulary of ordered graphs may also contain atomic formulae of the form $x \leqslant y$ with the obvious meaning, and formulae in other vocabularies may contain atomic formulae defined for these vocabularies. We use true as an abbreviation for the formula $\forall x\; x = x$ and false for $\neg$true. Moreover, we use $x \neq y$ as an abbreviation for $\neg x = y$.

We write $\varphi(x_1, \ldots, x_k)$ to indicate that the free variables of a 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 all $i \in [k]$. For $\ell \leq k$ and $v_1, \ldots, v_\ell$, by $\varphi[G, v_1, \ldots, v_\ell, x_{\ell+1}, \ldots, x_k]$ we denote the $(k - \ell)$-ary relation on on $V(G)$ that consists of all tuples $(v_{\ell+1}, \ldots, v_k)$ such that $G \models \varphi[v_1, \ldots, v_k]$. A *sentence* is a formula without free variables.

Example 2.1. The FO-sentence $\mathsf{graph} := \forall x \neg E(x, x) \wedge \forall x \forall y \big(E(x, y) \to E(y, x)\big)$ defines the class of all graphs. That is, for every $\{E\}$-structure G we have $G \models \mathsf{graph} \iff G$ is a graph.

Now consider the FO-formulae

$$\mathsf{iso}(x) := \forall y\; \neg E(x, y), \qquad \mathsf{no\text{-}iso} := \mathsf{graph} \wedge \neg \exists x\; \mathsf{iso}(x).$$

Then for every graph G and every vertex $v \in V(G)$ we have $G \models \mathsf{iso}[v]$ if and only if v is an isolated vertex in G. Hence $\mathsf{iso}[G, x]$ is the set of all isolated vertices of G. Furthermore, the sentence no-iso defines the class of all graphs without isolated vertices.

Example 2.2. Consider the FO-sentence

$$\mathsf{dagord} := \forall x \forall y \big(E(x, y) \to x \leqslant y\big)$$

in the vocabulary of ordered graphs. Then for every ordered directed graph $D = (V, E, \leqslant)$ we have $D \models \mathsf{dagord}$ if and only if the directed graph (V, E) is acyclic and the linear order $\leqslant^D$ is a refinement of the partial order $\trianglelefteq^{(V,E)}$.

Inflationary fixed-point logic IFP is the extension of FO by a fixed-point operator with an inflationary semantics. To introduce this operator, we first consider a special case and let $\varphi(X, \bar{x})$ be a formula that has a k-tuple $\bar{x} = (x_1, \dots, x_k)$ of free *individual variables* ranging over the vertices of a graph and a free k-ary *relation variable* X ranging over k-ary relations on the vertex set. (The general case is that φ has additional free variables.) For every graph G, we define a sequence of relations $R_i \subseteq V(G)^k$, for $i \in \mathbb{N}$, by

$$\begin{aligned} R_0 &:= \emptyset \\ R_{i+1} &:= R_i \cup \varphi[G, R_i, \bar{x}] \qquad\qquad \text{for all } i \in \mathbb{N}, \end{aligned}$$

where (in accordance with the notation introduced before) $\varphi[G, R_i, \bar{x}]$ denotes the set of all $\bar{v} \in V(G)^k$ such that G satisfies φ if the relation variable X is interpreted by R_i and the individual variables in $\bar{x}$ are interpreted by $\bar{v}$. Since we have $R_0 \subseteq R_1 \subseteq R_2 \subseteq \cdots \subseteq V(G)^k$ and $V(G)$ is finite, the sequence reaches a fixed point $R_n = R_{n+1}$, which we denote by R_∞. The *ifp-operator* applied to $\varphi, X, \bar{x}$ defines this fixed point. We use the following syntax.

$$\underbrace{\mathrm{ifp}(X\bar{x} \leftarrow \varphi)\bar{x}'}_{=:\psi(\bar{x}')} \tag{2.1}$$

Here $\bar{x}'$ is another k-tuple of individual variables, which may coincide with $\bar{x}$. The variables in the tuples $\bar{x}'$ are the free variables of the formula $\psi(\bar{x}')$, and for every tuple $\bar{v} \in V(G)^k$ of vertices we let $G \models \psi[\bar{v}] \iff \bar{v} \in R_\infty$. It is straightforward to extend this definition to the case that φ has additional free variables besides X and $\bar{x}$. These variables are usually referred to as *parameters* of the fixed-point definition, and they remain free variables of the formula $\mathrm{ifp}(X\bar{x} \leftarrow \varphi)\bar{x}'$. Now formulae of inflationary fixed-point logic IFP in the vocabulary of graphs are built from atomic formulae $E(x, y)$, $x = y$, and $X\bar{x}$ for relation variables X and tuples of individual variables $\bar{x}$ whose length matches the arity of X, by the usual Boolean connectives and existential and universal quantifiers ranging over the vertices of a graph, and the ifp-operator.

Example 2.3. Consider the IFP-formula

$$\mathsf{path}(x, y) := \mathrm{ifp}\Big(Y(y) \leftarrow \big(y = x \vee \exists y'(Y(y') \wedge E(y', y))\big)\Big)(y)$$

in the vocabulary of graphs. Then for all graphs G and all vertices $v, w \in V(G)$ we have $G \models \mathsf{path}[v, w]$ if and only if there is a path from v to w in G. Hence the IFP-sentence $\mathsf{conn} := \exists x\, x = x \wedge \forall x \forall y\, \mathsf{path}(x, y)$ says that a graph is connected. (Remember that the empty graph is disconnected.)

The formula $\mathsf{path}(x, y)$ keeps its meaning in directed graphs. Hence the IFP-sentence

$$\mathsf{dag} := \forall x \forall y \neg\big(E(y, x) \wedge \mathsf{path}(x, y)\big)$$

says that a directed graph is acyclic.

It can be shown that there are no FO-sentences equivalent to conn or dag.

Example 2.4. The FO-formulae

$$\begin{gathered} \mathsf{min}_{\leqslant}(x) := \forall y\, x \leqslant y, \qquad \mathsf{max}_{\leqslant}(x) := \forall y\, y \leqslant x, \\ \mathsf{succ}_{\leqslant}(x, y) := x \leqslant y \wedge x \neq y \wedge \forall z(z \leqslant x \vee y \leqslant z) \end{gathered}$$

define the first and last element of a linear order and the successor relation. Now consider the IFP-sentence

$$\begin{aligned} \mathsf{odd} := \exists x \exists y\, (&\mathsf{min}_{\leqslant}(x) \wedge \mathsf{max}_{\leqslant}(y) \\ &\wedge \mathrm{ifp}(Y(y) \leftarrow (y = x \vee \exists y' \exists y''(Y(y') \wedge \mathsf{succ}_{\leqslant}(y', y'') \wedge \mathsf{succ}_{\leqslant}(y'', y))))(y)) \end{aligned}$$

in the vocabulary of ordered graphs. Then for each ordered graph G we have $G \models \mathsf{odd}$ if any only if $|G|$ is an odd number.

It can be shown that there is no IFP-sentence φ in the vocabulary of graphs such that for each graph G it holds that $G \models \varphi$ if any only if $|G|$ is odd.

Our next goal is to define the *counting extension* IFP+C of IFP. The formal definition of this logic and many of the basic results about it go back to Grädel and Otto [1993]. We first introduce an intermediate logic IFP^+. To define this logic, we need to consider two sorted extensions of graphs or other relational structures by a numerical sort. For a structure G, we let $\mathit{Num}(G)$ be the initial segment $\big[0, |G|\big]$ of $\mathbb{N}$. To avoid confusion later, note that $|\mathit{Num}(G)| = |G| + 1$. We let G^+ be the two-sorted structure $G \cup (\mathit{Num}(G), \leq)$, where $\leq$ is the natural linear order on $\mathit{Num}(G)$. We call the elements of the first sort $V(G)$ *vertices* and the elements of the second sort $\mathit{Num}(G)$ *numbers*. Individual variables of our logic are either *vertex variables* ranging over the set $V(G)$ of vertices of G or *number variables* ranging over $\mathit{Num}(G)$. Two tuples $\bar{x} = (x_1, \dots, x_k), (y_1, \dots, y_\ell)$ of individual variables have the same *type* if $k = \ell$ and for all $i \in [k]$ either both x_i and y_i are vertex variables or both x_i and y_i are number variables. For every structure G, we let $G^{\bar{x}}$ be the set of all tuples $\bar{p} = (p_1, \dots, p_k) \in (V(G) \cup \mathit{Num}(G))^k$ such that for all $i \in [k]$ we have $p_i \in V(G)$ if x_i is a vertex variable and $p_i \in \mathit{Num}(G)$ if x_i is a number variable. Relation variables of the logic $\mathsf{IFP{+}C}^+$ range over sets of tuples of a prescribed type. Let IFP^+ be inflationary fixed-point logic equipped with the new kinds of variables and interpreted over the two-sorted extensions of structures. We may still view IFP^+ as a logic over plain structures, because the extension G^+ is uniquely determined by G. We say that a sentence φ of IFP^+ is satisfied by a structure G (and write $G \models \varphi$) if $G^+ \models \varphi$.

Example 2.5. All individual variables in this example are number variables. Consider the IFP^+-formulae

$$\begin{aligned}
\mathsf{min}(x) &:= \forall y\, x \leq y, \\
\mathsf{max}(x) &:= \forall y\, y \leq x, \\
\mathsf{succ}(x, y) &:= x \leq y \wedge x \neq y \wedge \forall z (z \leq x \vee y \leq z) \\
\mathsf{plus}(x_1, x_2, x_3) &:= \mathrm{ifp}\, \Big(Y(y, y') \leftarrow (\mathsf{min}(y) \wedge y' = x_2) \vee \\
&\qquad\qquad \exists z \exists z' \big(Y(z, z') \wedge \mathsf{succ}(z, y) \wedge \mathsf{succ}(z', y') \big) \Big)(x_1, x_3).
\end{aligned}$$

Then for all structures G and all $n_1, n_2, n_3 \in \mathit{Num}(G)$ we have $G^+ \models \mathsf{plus}[n_1, n_2, n_3]$ if and only if $n_1 + n_2 = n_3$. Moreover, we have

$$G \models \exists x \big(\mathsf{max}(x) \wedge \exists y\, \mathsf{plus}(y, y, x) \big)$$

if and only if $|G|$ is even.

Inflationary fixed-point logic with counting IFP+C is the extension of IFP^+ by *counting formulae* formed as follows. For every formula φ, every vertex variable x, and every number variable y, we may form a new formula $\#x\ \varphi = y$, which says that there are precisely y vertices x such that φ holds. More precisely, the free variables of $\#x\ \varphi = y$ are y together with the free variables of φ except x. If we write $\psi(y, \bar{z}) := \#x\ \varphi(x, \bar{z}) = y$ then for every structure G, every tuple $\bar{w} \in G^{\bar{z}}$, and every $n \in \mathit{Num}(G)$, we have $G \models \psi[n, \bar{w}]$ if any only if $|\varphi[G, x, \bar{w}]| = n$.

Example 2.6. In this example, x, x' are vertex variables and y, y' are number variables. Consider the IFP+C-formula

$$\mathsf{even\text{-}deg}(x) := \exists y \exists y' \big(\#x'\ E(x, x') = y \wedge \mathsf{plus}(y', y', y) \big)$$

in the vocabulary of graphs. For all graphs G and all vertices $v \in V(G)$ we have $G \models \textsf{even-deg}[v]$ if and only if $\deg^G(v)$ is even.

An *Eulerian cycle* in a graph is a closed walk on which every edge occurs exactly once. It is a well-known fact that a graph has an Eulerian cycle if and only if it is connected and every vertex has even degree. Then the following IFP+C-sentence defines the class of all graphs that have an Eulerian cycle:

$$\textsf{eulerian} := \textsf{conn} \wedge \forall x\ \textsf{even-deg}(x)$$

where conn is the IFP-sentence from Example 2.3.

2.4. Transductions

In the following, L is one of the logics IFP+C, IFP, or FO, and τ, υ are vocabularies such as the vocabulary $\{E\}$ of graphs or the vocabulary $\{E, \leqslant\}$ of ordered graphs. An $\mathsf{L}[\tau]$-*formula* is an L-formula in the vocabulary τ, and similarly for υ.

Definition 2.7.

(1) An $\mathsf{L}[\tau, \upsilon]$-*transduction* is a tuple

$$\Theta(\bar{x}) = \Big(\theta_{dom}(\bar{x}), \theta_V(\bar{x}, \bar{y}), \big(\theta_R(\bar{x}, \bar{y}_R)\big)_{R\in\upsilon}\Big),$$

of $\mathsf{L}[\tau]$-formulae, where $\bar{x}$, $\bar{y}$, and $\bar{y}_R$ for $R \in \upsilon$ are tuples of individual variables such that for every k-ary $R \in \upsilon$ the tuple $\bar{y}_R$ can be written as $\bar{y}_{R,1} \dots \bar{y}_{R,k}$, where the $\bar{y}_{R,i}$ have the same type as $\bar{y}$.
The variables in $\bar{x}$ are called the *parameters* of the transduction. Then length $|\bar{y}|$ of the tuple $\bar{y}$ is the *dimension* of the transduction.

In the following, let $\Theta(\bar{x})$ be an $\mathsf{L}[\tau, \upsilon]$-transduction.

(2) The domain of $\Theta(\bar{x})$ is the class $\mathcal{D}_{\Theta(\bar{x})}$ of all pairs $(G, \bar{p})$, where G is a τ-structure and $\bar{p} \in G^{\bar{x}}$ such that $G \models \theta_{dom}[\bar{p}]$.
(3) For all $(G, \bar{p}) \in \mathcal{D}_{\Theta(\bar{x})}$ we let $\Theta[G, \bar{p}]$ be the υ-structure with vertex set

$$V\big(\Theta[G, \bar{p}]\big) := \theta_V[G, \bar{p}, \bar{y}] \subseteq G^{\bar{y}}$$

and relations

$$R\big(\Theta[G, \bar{p}]\big) := V\big(\Theta[G, \bar{p}]\big)^k \cap \varphi_R[G, \bar{p}, \bar{y}_{R,1}, \dots, \bar{y}_{R,k}]$$

for each k-ary $R \in \upsilon$.

Example 2.8. Let Θ be the parameter-free 1-dimensional $\mathsf{IFP}[\{E\}, \{E\}]$-transduction defined by $\theta_{dom} := \textsf{true}$, $\theta_V(y) := \textsf{true}$, $\theta_E(y_1, y_2) := \textsf{path}(y_1, y_2)$, where path is the IFP-formula of Example 2.3.

Then for every directed graph D, the directed graph $\theta[D]$ has the same vertex set as D, and its edge relation is the reflexive transitive closure of the edge relation of D.

$\mathsf{L}[\tau, \upsilon]$-transductions map τ-structures to υ-structures. The crucial observation is that they also induce a reverse mapping from $\mathsf{L}[\upsilon]$-formulae to $\mathsf{L}[\tau]$-formulae.

FACT 2.9 (TRANSDUCTION LEMMA). *Let $\Theta(\bar{x})$ be an $\mathsf{L}[\tau, \upsilon]$-transduction. Then for every $\mathsf{L}[\upsilon]$-sentence φ there is an $\mathsf{L}[\tau]$-formula $\varphi^{-\Theta}(\bar{x})$ such that for all $(G, \bar{p}) \in \mathcal{D}_{\Theta(\bar{x})}$ we have*

$$G \models \varphi^{-\Theta}[\bar{p}] \iff \Theta[G, \bar{p}] \models \varphi.$$

A proof of this fact for first-order logic can be found in Ebbinghaus et al. [1994]. The proof for the other logics considered here is an easy adaptation of the one for first-order logic.

An L-*graph transduction* is an L$[\{E\}, \{E\}]$-transduction $\Theta(\bar{x})$ such that for all $(G, \bar{p}) \in \mathcal{D}_{\Theta(\bar{x})}$ the $\{E\}$-structures G and $\Theta[G, \bar{p}]$ are graphs.

Example 2.10. Let $k \in \mathbb{N}$ and $\bar{x} := (x_1, \dots, x_k)$. Consider the following IFP-graph transduction $\Theta(\bar{x})$ with $\theta_{dom}(\bar{x}) :=$ graph, $\theta_V(\bar{x}, y) := \bigwedge_{i=1}^{k} y \neq x_i$, $\theta_E(\bar{x}, y_1, y_2) := E(y_1, y_2)$. Then for all graphs G and all tuples $\bar{v} \in V(G)^k$ we have $(G, \bar{v}) \in \mathcal{D}_{\Theta(\bar{x})}$ and $\Theta[G, \bar{v}] = G \setminus \widetilde{v}$.

Let conn be an IFP-formula expressing connectedness (see Example 2.3). Then by the Transduction Lemma, for all graphs G and all $\bar{v} \in V(G)^k$ we have $G \models \mathsf{conn}^{-\Theta}[\bar{v}]$ if and only if $G \setminus \widetilde{v}$ is connected. Hence

$$G \models \forall x_1 \dots \forall x_k \, \mathsf{conn}^{-\Theta}(\bar{x}) \wedge \exists y_1 \dots \exists y_{k+2} \bigwedge_{\substack{i,j \in [k+2] \\ i \neq j}} y_i \neq y_j.$$

if and only if G is $(k+1)$-connected.

2.5. Logics Capturing Polynomial Time

We briefly review the necessary background from descriptive complexity theory. We assume that the reader is familiar with the basics of complexity theory; all we need is knowledge of the complexity class PTIME. In standard complexity theory, complexity classes are classes of decision problems described as languages over finite alphabets. In descriptive complexity theory, decision problems are described as *properties* of finite structures, that is, classes of finite structures closed under isomorphism. Hence complexity classes correspond to classes of properties of finite structures.

To establish the correspondence between formal languages and properties of structures, we need to encode structures by strings over some finite alphabet. As everywhere else in this article, we focus on graphs and encode them by their adjacency matrices. (The adjacency-matrix encoding of graphs can easily be generalized to other relational structures.) As discussed in the Introduction, adjacency matrices are not canonical; potentially, every linear order of the vertices of a graph yields a different adjacency matrix. For every graph G, we let $\mathcal{L}(G)$ be the set of all $\{0, 1\}$-strings obtained from some adjacency matrix of G by concatenating its rows. For a property $\mathcal{P}$ of graphs, we let

$$\mathcal{L}(\mathcal{P}) := \bigcup_{G \in \mathcal{P}} \mathcal{L}(G).$$

We say that an algorithm *decides* $\mathcal{P}$ if it decides $\mathcal{L}(\mathcal{P})$, and we say that $\mathcal{P}$ is *decidable in polynomial time* if there is a polynomial-time algorithm deciding $\mathcal{P}$, or equivalently, if the language $\mathcal{L}(\mathcal{P})$ is in PTIME.

Intuitively, a logic L captures polynomial time if for all properties $\mathcal{P}$ of structures, $\mathcal{P}$ is definable in L if and only if $\mathcal{P}$ is decidable in polynomial time. This definition is sufficient for the purposes of this article. For the reader's convenience, a precise definition of a logic capturing polynomial time can be found in Appendix A.

We need a relativised version of the definition. Let $\mathcal{C}$ be a class of structures and $\mathcal{P}$ a property. We say that an algorithm $\mathfrak{A}$ *decides* $\mathcal{P}$ *on* $\mathcal{C}$ if on input $w \in \mathcal{L}(G)$ for a structure $G \in \mathcal{C}$, the algorithm accepts if and only if $G \in \mathcal{P}$. The behavior of the algorithm on inputs $w \in \{0, 1\}^*$ that are not in $\mathcal{L}(G)$ for any structure $G \in \mathcal{C}$ is irrelevant. We say that an L-sentence φ *defines* $\mathcal{P}$ *on* $\mathcal{C}$ if for every structure $G \in \mathcal{C}$, the structure G satisfies φ if and only if $G \in \mathcal{P}$. Then L *captures* PTIME *on* $\mathcal{C}$ if for all properties $\mathcal{P}$ of structures, $\mathcal{P}$ is definable in L on $\mathcal{C}$ if and only if there is a polynomial-time algorithm that decides $\mathcal{P}$ on $\mathcal{C}$.

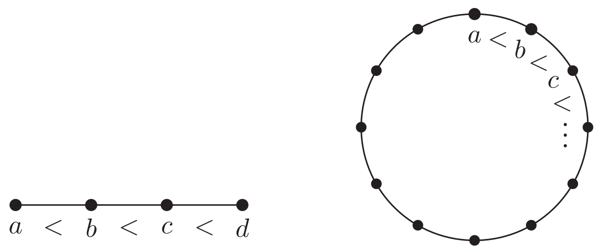


Fig. 2.1. Defining an order on a path and cycle; the white vertices are parameters of the definition.

2.6. Ordered Structures and the Immerman-Vardi Theorem

Recall that an *ordered structure* is a structure A whose vocabulary contains the binary relation symbol $\leqslant$ and in which $\leqslant^A$ is a linear order of $V(A)$. The prime example of ordered structures are ordered graphs. Descriptive complexity theory is easier on ordered structures, because ordered structures have a canonical string encoding. For example, we can encode an ordered graph G by the $\{0, 1\}$-string obtained by concatenating the rows of the adjacency matrix of the graph $(V(G), E(G))$ in which the rows and columns are ordered with respect to the linear order $\leqslant^G$. This can easily be generalized to arbitrary ordered structures. And indeed, there is a logic capturing PTIME on ordered structures.

FACT 2.11 (IMMERMAN-VARDI THEOREM [IMMERMAN 1986; VARDI 1982]). IFP *captures* PTIME *on the class of all ordered structures.*

The Immerman-Vardi Theorem can be used to prove that the logic IFP (and also stronger logics like IFP+C) captures PTIME not only on ordered structures, but on all classes of structures that admit a definable order. For example, on directed paths we can easily define linear orders in IFP. The transitive closure of the edge relation of a directed path happens to be a linear order of the path's vertices, and it is easy to see that the transitive closure is definable in IFP (refer to Example 2.3). On other structures, we cannot define linear orders directly, but need "parameters" in our definition. For example, on an undirected path, it is impossible to define a linear order without parameters, simply because there is no canonical way of defining which endpoint of the path comes first. More generally, if a graph has a nontrivial automorphism then it is impossible to define a linear order on this graph. However, if we fix one endpoint of an undirected path then we can define a linear order in the logic IFP. Similarly, if we fix two adjacent vertices in a cycle we can define a linear order on the cycle (see Figure 2.1).

Definition 2.12. Let L be a logic (like FO, IFP, or IFP+C).

(1) A formula $\varphi(\bar{x}, y_1, y_2)$ with vertex variables y_1, y_2 *defines a linear order* on a graph G *with parameters* $\bar{x}$ if there is a tuple $\bar{p} \in G^{\bar{x}}$ such that $\varphi[G, \bar{p}, y_1, y_2]$ is a linear order of $V(G)$.
(2) A class $\mathcal{C}$ of graphs *admits* L*-definable linear orders* if there is an L-formula $\varphi(\bar{x}, y_1, y_2)$ that defines a linear order on every graph $G \in \mathcal{C}$.

The following well-known lemma is an immediate consequence of the Immerman-Vardi Theorem (Fact 2.11).

LEMMA 2.13. *Let* L *be be one of the logics* IFP *or* IFP+C*, and let* $\mathcal{C}$ *be a class of graphs that admits* L*-definable linear orders. Then* L *captures* PTIME *on* $\mathcal{C}$.

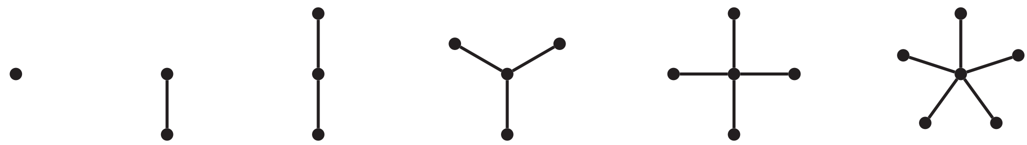


Fig. 2.2. A few stars.

2.7. Definable Canonization

We say that an ordered graph $H = (V(H), E(H), \leqslant^H)$ is an *ordered copy* of a graph G if $G \cong (V(H), E(H))$. Recall that two ordered graphs G, H are *isomorphic* if there is a mapping $f : V(G) \to V(H)$ such that for all $v, w \in V(G)$ we have $(vw \in E(G) \iff f(v)f(w) \in E(H))$ and $(v \leqslant^G w \iff f(v) \leqslant^H f(w))$, and note there is at most one isomorphism between two ordered graphs. A *canonization mapping* $\mathfrak{c}$ for a class $\mathcal{C}$ of graphs associates with every graph $G \in \mathcal{C}$ an ordered copy $\mathfrak{c}(G)$ of G such that for all $G, H \in \mathcal{C}$ we have $G \cong H \iff \mathfrak{c}(G) \cong \mathfrak{c}(H)$. We are interested in canonization mappings definable by $\mathsf{L}[\{E\}, \{E, \leqslant\}]$-transductions.

Definition 2.14. Let L be a logic (like FO, IFP, or $\mathsf{IFP{+}C}$).

(1) An $\mathsf{L}[\{E\}, \{E, \leqslant\}]$-transduction $\Theta(\bar{x})$ *canonizes* a graph G if there is at least one tuple $\bar{p} \in G^{\bar{x}}$ such that $(G, \bar{p}) \in \mathcal{D}_{\Theta(\bar{x})}$, and for all tuples $\bar{p} \in G^{\bar{x}}$ such that $(G, \bar{p}) \in \mathcal{D}_{\Theta(\bar{x})}$ the ordered graph $\Theta[G, \bar{p}]$ is an ordered copy of G.
$\Theta(\bar{x})$ *canonizes* a class $\mathcal{C}$ of graphs if it canonizes all graphs in $\mathcal{C}$.
(2) A class $\mathcal{C}$ of graphs *admits* L*-definable canonization* if there is an $\mathsf{L}[\{E\}, \{E, \leqslant\}]$-transduction that canonizes $\mathcal{C}$.

Note that a parameter-free transduction canonizing a class of graphs indeed defines a canonization mapping for this class. Transductions with parameters define not just one ordered copy of their input graphs, but polynomially many. This will still be good enough for our purposes.

The following example illustrates that defining a canonization is not the same as defining an order.

Example 2.15. A *star* is a tree of height 1 (refer to Figure 2.2). The class of all stars does not admit $\mathsf{IFP{+}C}$-definable orders, because even if we fix k vertices as parameters, a star with at least $k + 3$ vertices still has a nontrivial automorphism.

Let y, y_1, y_2 be number variables. Let Θ be the $\mathsf{IFP{+}C}[\{E\}, \{E, \leqslant\}]$-transduction with

—$\theta_{dom} := \mathsf{true}$,
—$\theta_V(y) := \neg\mathsf{max}(y)$, where $\mathsf{max}(y) := \forall y'\ y' \le y$,
—$\theta_E(y_1, y_2) := (\mathsf{min}(y_1) \wedge \neg\mathsf{min}(y_2)) \vee (\mathsf{min}(y_2) \wedge \neg\mathsf{min}(y_1))$, where $\mathsf{min}(y) := \forall y'\ y \le y'$,
—$\theta_{\leqslant}(y_1, y_2) := y_1 \le y_2$.

Observe that the transduction does not even look at the edge relation of the input graph. It just assumes that that graph is a star; and under this assumption the graph is determined up to isomorphism by its cardinality. The vertex set of the canonical copy $\Theta(S)$ of an input star S of order $n := |S|$ is the initial segment of $Num(S)$ (the number set associated with S) of length n, that is, the interval $[0, n-1] = Num(S) \setminus \{n\}$. The edge set $E(\Theta[S])$ is $\{0i \mid i \in [n-1]\}$, and the linear order $\leqslant^{\Theta[S]}$ is the natural linear order on $V(\Theta[S]) = [0, n-1]$. Hence Θ canonizes the class of stars.

The following lemma, which underlies all known results about capturing polynomial time on specific classes of graphs, is another straightforward consequence of the Immerman-Vardi Theorem.

Fig. 3.1. The cycle C_5 together with a tree decomposition of the cycle.

LEMMA 2.16. *Let* L *be one of the logics* IFP *or* IFP+C. *Let* $\mathcal{C}$ *be a class of graphs that admits* L*-definable canonization. Then* L *captures* PTIME *on* $\mathcal{C}$.

The idea of using definable canonization to capture polynomial time goes back to Immerman and Lander [1990]. It has been developed to great depth by Otto [1997]. Example 2.15 shows that the class of stars admits IFP+C-definable canonization. By applying the same idea recursively, Immerman and Lander [1990] were able to prove that the class of all trees admits IFP+C-definable canonization.

3. TREELIKE DECOMPOSITIONS

Suppose that we want to define a tree decomposition (T, β) of a graph G in some logic, say IFP. We could try to define an IFP-graph transduction Θ such that $\Theta[G] \cong T$, and then define the bags of the decomposition by a formula $\varphi_\beta(\bar{x}, y)$, where $\bar{x}$ is a tuple of individual variables whose length matches the dimension of Θ. Unfortunately, most interesting tree decompositions are not definable in this way, no matter what logic we use, because the decompositions are not invariant under automorphisms of the graph. What this means is that there may be an automorphism f of G for which we cannot find an automorphism g of T such that for all nodes t we have $\beta(g(t)) = f(\beta(t))$. As an example, consider the decomposition of the cycle C_5 displayed in Figure 3.1. The problem is that only invariant objects are logically definable in the graph.

We resolve this problem by introducing a more general notion of decomposition that we call *treelike*. In a treelike decomposition, we replace the tree T underlying a tree decomposition by a directed acyclic graph D. The idea is that certain restrictions of D to subtrees yield tree decompositions of G, and by including many such decompositions we can close the treelike decompositions under automorphisms of a graph. To get an impression how treelike decompositions look, consider Figure 3.2, which shows a treelike decomposition of the cycle C_5. The sets displayed in the nodes of the decomposition are the bags. Observe that the four grey nodes form exactly the tree decomposition of C_5 displayed in Figure 3.1. There are many tree decompositions of C_5 contained in the treelike decomposition in a similar way. Note the cyclic structure of the whole decomposition, which reflects the structure of the underlying cycle and is the reason for the invariance of the decomposition under automorphisms of the cycle. This cyclic structure is lost in a tree decomposition like the one in Figure 3.1.

In Section 3.4, we will extend treelike decompositions to *ordered treelike decompositions*, which one may think of as treelike decompositions together with linear orders of all bags. In our *definable structure theory*, we will show be that certain classes of graphs admit IFP-definable ordered treelike decompositions.

We will start by introducing a general notion of decomposition of a graph and then give axioms for treelike decompositions. It will be convenient to view decompositions from a slightly different angle than the usual one for tree decompositions. As a matter of

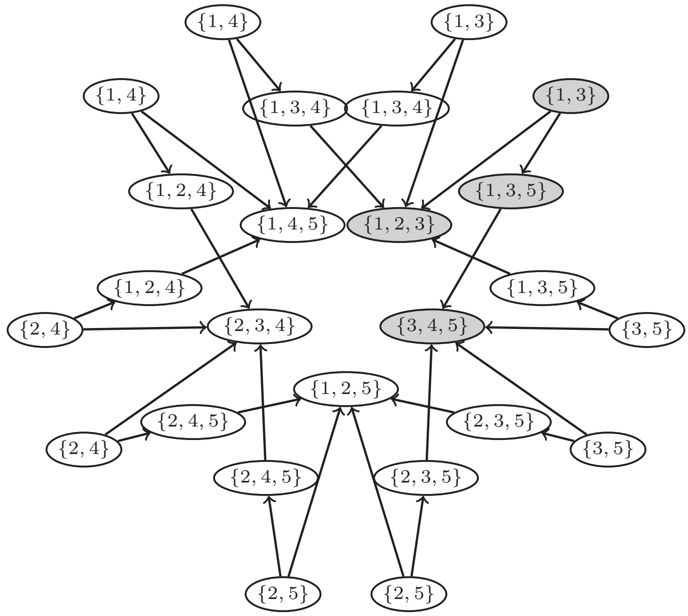


Fig. 3.2. An automorphism-invariant treelike decomposition of the cycle C_5.

fact, formally tree decompositions are not decompositions in our sense. To understand the definitions, it will thus be helpful to first review tree decompositions from the new angle (refer to Remark 3.4). So let (T, β) be a tree decomposition of a graph G. Recall that we view T as a directed tree and that $\trianglelefteq^T$ denotes the reflexive transitive closure of the edge relation of T, that is, the partial tree order where the root is the unique minimum. For every node $t \in V(T)$ we define the *cone* $\gamma(t)$ of T at t to be the union of $\beta(t)$ with all bags $\beta(u)$ for descendants u of t. We define the *separator* $\sigma(t)$ of T at t to be $\beta(t) \cap \beta(s)$ if s is the parent of t in T, and we let $\sigma(r) := \emptyset$ for the root r of T. Finally, we define the *component* $\alpha(t)$ to be $\gamma(t) \setminus \sigma(t)$. It is a basic property of tree decompositions that $\sigma(t)$ separates $\alpha(t)$ from $V(G) \setminus \gamma(t)$. It is not always the case that $\alpha(t)$ is the vertex set of a connected component of $G \setminus \sigma(t)$ (it may also be the vertex set of a union of connected components), but often it is—hence the name "component". Observe that we can define the bags and cones in terms of separators and components: we have $\gamma(t) = \sigma(t) \cup \alpha(t)$ and $\beta(t) = \gamma(t) \setminus \bigcup_u \alpha(u)$, where the union ranges over all children u of t. We choose separators and components rather than bags to be the basic objects in our decompositions.

In some sense, the main contributions of this section are the definitions. The proofs of the results are often long and technical, but not very difficult. For that reason, we only sketch many of them in the main article and defer full proofs to the appendix.

3.1. Definitions and Basic Lemmas

A *decomposition* of a graph G is a triple $\Delta = (D^\Delta, \sigma^\Delta, \alpha^\Delta)$ where D^Δ is a directed graph and $\sigma^\Delta, \alpha^\Delta$ are mappings from $V(D^\Delta)$ to $2^{V(G)}$. For every node $t \in V(D^\Delta)$ we let

$$\gamma^\Delta(t) := \alpha^\Delta(t) \cup \sigma^\Delta(t), \tag{3.1}$$

$$\beta^\Delta(t) := \gamma^\Delta(t) \setminus \bigcup_{u \in N_+(t)} \alpha^\Delta(u), \tag{3.2}$$

where we write $N_+(t)$ instead of $N_+^{D^\Delta}(t)$. The sets $\alpha^\Delta(t)$, $\beta^\Delta(t)$, $\gamma^\Delta(t)$, and $\sigma^\Delta(t)$ are called the *component*, *bag*, *cone*, and *separator*, respectively, of Δ at t. The *torso* of Δ at t is the graph

$$\tau^\Delta(t) := G\big[\beta^\Delta(t)\big] \cup K\big[\sigma^\Delta(t)\big] \cup \bigcup_{u \in N_+(t)} K\big[\sigma^\Delta(u)\big].$$

Two nodes $t, t' \in V(D^\Delta)$ are Δ-*equivalent* (we write $t \approx^\Delta t'$) if $\sigma^\Delta(t) = \sigma^\Delta(t')$ and $\alpha^\Delta(t) = \alpha^\Delta(t')$. Note that Δ-equivalent nodes have the same cones, but not necessarily the same bags, because the bags also depend on the children of the nodes.

In all these notations, we omit the index $^\Delta$ if Δ is clear from the context. Often, we leave decompositions unnamed and just introduce them as triples (D, α, σ). We frequently use implicit naming conventions such as the following: for a decomposition $\Delta' = (D', \sigma', \alpha')$ we let $\beta' := \beta^{\Delta'}$, $\gamma' := \gamma^{\Delta'}$, et cetera.

Definition 3.1. A *treelike decomposition* of a graph G is a decomposition $\Delta = (D, \sigma, \alpha)$ of G that satisfies the following axioms.

(TL.1) D is acyclic.
(TL.2) For all $t \in V(D)$ it holds that $\alpha(t) \cap \sigma(t) = \emptyset$ and $N^G(\alpha(t)) \subseteq \sigma(t)$.
(TL.3) For all $t \in V(D)$ and $u \in N_+^D(t)$ it holds that $\alpha(u) \subseteq \alpha(t)$ and $\gamma(u) \subseteq \gamma(t)$.
(TL.4) For all $t \in V(D)$ and $u_1, u_2 \in N_+^D(t)$, either $u_1 \approx u_2$ or $\gamma(u_1) \cap \gamma(u_2) = \sigma(u_1) \cap \sigma(u_2)$.
(TL.5) For every connected component A of G there is a $t \in V(D)$ with $\sigma(t) = \emptyset$ and $\alpha(t) = V(A)$.

Directed acyclic graphs are "moderate" generalizations of (directed) trees, and they allow the flexibility we need to make the decompositions automorphism invariant, while at the same time allowing us to work with the decompositions in a bottom-up, "dynamic programming style", fashion. This explains Axiom (TL.1). Axiom (TL.2) makes sure that separators and components behave essentially as their name indicates (except that components may be unions of connected components of the graph, as for tree decompositions). Axiom (TL.3) is a basic compatibility condition between the components/cones and the edge relation of D. Axiom (TL.5) makes sure that the whole graph is covered by the decomposition. (TL.4) is the most interesting axiom. The corresponding axiom for a tree decomposition would require $\gamma(u_1) \cap \gamma(u_2) = \sigma(u_1) \cap \sigma(u_2)$ for all distinct $u_1, u_2 \in N_+^D(t)$. By allowing $u_1 \approx u_2$, we add a lot more flexibility. The idea is that, while building a decomposition top-down, at some node t we may not have a canonical way to decompose some induced subgraph H of $G[\gamma(t)]$. The relaxed axiom allows us to take several different decompositions of H and add a child u to t with $\gamma(u) = V(H)$ for each of these decompositions. The price we pay for this added flexibility is that we lose the "connectedness condition" of tree decomposition: in a treelike decomposition (D, σ, α) of a graph G it is *not* necessarily the case that for each $v \in V(G)$ the set of all t such that $v \in \beta(t)$ is connected in the undirected graph underlying D.

Example 3.2. For every cycle C, we define a decomposition $\Delta(C) = (D, \sigma, \alpha)$ as follows.

$$V(D) := \big\{(v_1, v_2, v_3) \in V(C)^3 \;\big|\; v_1 \neq v_2, v_3\big\},$$
$$E(D) := \big\{(v_1, v_2, v_3)(v_1', v_2', v_3') \in V(D)^2 \;\big|$$

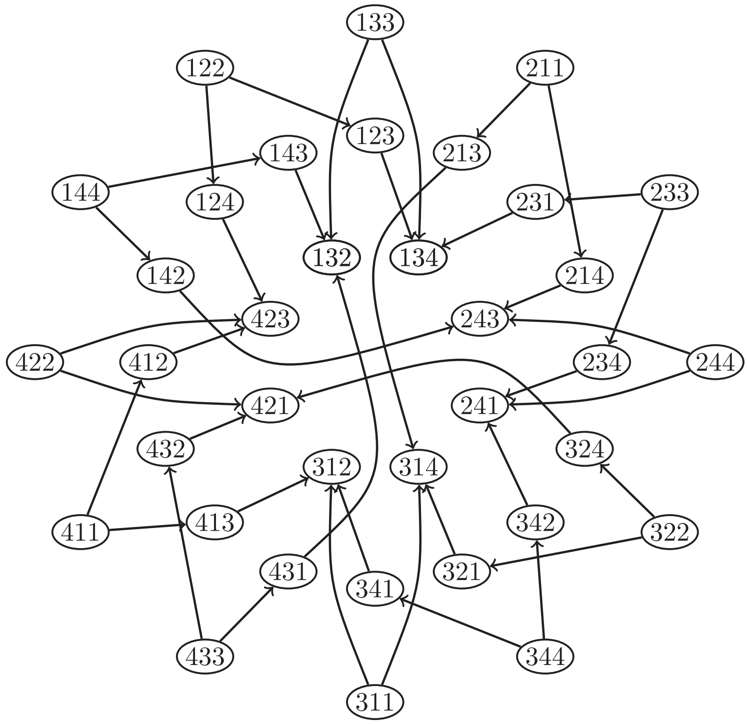


Fig. 3.3. The directed graph of the treelike decomposition of the cycle $C_4 := \big([4], \{12, 23, 34, 41\}\big)$ defined in Example 3.2.

$$
\begin{aligned}
& v_3 = v_2 \text{ and } v_1' = v_1 \text{ and } v_2' = v_2 \text{ and } v_3' \neq v_2' \\
\text{or} \quad & v_3 \neq v_2 \text{ and } v_1' = v_1 \text{ and } v_2' = v_3 \text{ and } v_3' \neq v_2' \\
& \quad \text{and } \{v_1', v_2'\} \text{ separates } v_3' \text{ from } v_2 \text{ in } C \\
\text{or} \quad & v_3 \neq v_2 \text{ and } v_1' = v_3 \text{ and } v_2' = v_2 \text{ and } v_3' \neq v_2' \\
& \quad \text{and } \{v_1', v_2'\} \text{ separates } v_3' \text{ from } v_1 \text{ in } C\Big\},
\end{aligned}
$$

$$
\sigma(v_1, v_2, v_3) := \begin{cases} \emptyset & \text{if } v_3 = v_2, \\ \{v_1, v_2\} & \text{otherwise,} \end{cases}
$$

$$
\begin{aligned}
\alpha(v_1, v_2, v_3) := V(A),\ & \text{where } A \text{ is the connected component of} \\
& C \setminus \sigma(v_1, v_2, v_3) \text{ that contains } v_3.
\end{aligned}
$$

Figure 3.3 shows the digraph D for the cycle $C := C_4$ of length 4. For the node $t := (1, 2, 3)$ (in Figure 3.3, the node appears as 123 in the twelve-o'clock position of the second layer), we have $\sigma(t) = \{1, 2\}$, $\alpha(t) = \{3, 4\}$, $\gamma(t) = \{1, 2, 3, 4\}$, $\beta(t) = \{1, 2, 3\}$, and $\tau(t) = K[\{1, 2, 3\}]$.

As another example, consider the cycle $C := C_8 := \big([8], \{12, 23, \dots, 78, 81\}\big)$ of length 8. Then for the node $t = (3, 6, 8) \in V(\Delta(C))$ we have

$$
\begin{aligned}
\sigma(t) &= \{3, 6\}, & N_+^D(t) &= \{(3, 8, 1), (3, 8, 2), (8, 6, 7)\}, \\
\alpha(t) &= \{7, 8, 1, 2\}, & \beta(t) &= \{3, 6, 8\}, \\
\gamma(t) &= \{6, 7, 8, 1, 2, 3\}, & \tau(t) &= \big(\{3, 6, 8\}, \{36, 68, 38\}\big).
\end{aligned}
$$

We leave it to the reader to verify that for every cycle C the decomposition $\Delta(C)$ is a treelike decomposition of C. Furthermore, for all $\bar{v} \in V(D)$ it holds that $\beta(t) = \widetilde{v}$ and $\tau(t) = K[\widetilde{v}]$.

The following lemma collects a few basic properties of treelike decompositions. Recall that for a directed graph D, $\trianglelefteq^D$ denotes the reflexive transitive closure of $E(D)$, which is a partial order if D is acyclic. We write $t \triangleleft^D u$ to denote that $t \trianglelefteq^D u$ and $u \ntrianglelefteq^D t$.

LEMMA 3.3 (β-γ-σ-LEMMA). *Let $\Delta = (D, \sigma, \alpha)$ be a treelike decomposition of a graph G, and let $t, u \in V(\Delta)$. Then:*

(1) *$\gamma(t) = \bigcup_u \beta(u)$, where the union ranges over all $u \in V(D)$ with $t \trianglelefteq^D u$.*
(2) *$\sigma(t) = \beta(t) \setminus \alpha(t)$ and thus, in particular, $\sigma(t) \subseteq \beta(t)$.*
(3) *If $tu \in E(D)$ then $\beta(t) \cap \beta(u) = \beta(t) \cap \gamma(u) = \sigma(u)$.*
(4) *If $t \triangleleft^D u$ then $\gamma(u) \subseteq \gamma(t)$ and $\beta(t) \cap \gamma(u) \subseteq \sigma(u)$.*

PROOF. Assertion (1) is proved by induction on $t \in V(D)$. If $N_+^D(t) = \emptyset$, then $\gamma(t) = \beta(t)$. Otherwise,

$$\gamma(t) \subseteq \beta(t) \cup \bigcup_{u \in N_+^D(t)} \alpha(u) \;\subseteq\; \beta(t) \cup \bigcup_{u \in N_+^D(t)} \gamma(u) \;\subseteq\; \gamma(t)$$

where the last inclusion holds by (TL.3). Hence

$$\gamma(t) = \beta(t) \cup \bigcup_{u \in N_+^D(t)} \gamma(u) = \beta(t) \cup \bigcup_{u \in V(D) \text{ with } t \triangleleft^D u} \beta(u)$$

by the induction hypothesis.

To prove (2), we first prove $\sigma(t) \subseteq \beta(t)$. Note that $\sigma(t) \cap \alpha(t) = \emptyset$ by (TL.2) and therefore $\sigma(t) \cap \alpha(u) = \emptyset$ for all $u \in N_+^D(t)$ by (TL.3). Hence $\sigma(t) \subseteq \beta(t)$ follows from the definition (3.2) of $\beta(t)$. Now $\sigma(t) = \beta(t) \setminus \alpha(t)$ follows from $\sigma(t) = \gamma(t) \setminus \alpha(t)$ and $\sigma(t) \subseteq \beta(t) \subseteq \gamma(t)$.

To prove (3), let $tu \in E(D)$. Observe that

$$\beta(t) \cap \beta(u) \subseteq \beta(t) \cap \gamma(u) \subseteq \sigma(u),$$

because $\beta(t) \cap \alpha(u) = \emptyset$ by the definition of β. For the converse inclusion $\sigma(u) \subseteq \beta(t) \cap \beta(u)$, we only have to prove $\sigma(u) \subseteq \beta(t)$, because $\sigma(u) \subseteq \beta(u)$ by (2). Let $v \in \sigma(u)$. Then $v \in \gamma(u) \subseteq \gamma(t)$ by (TL.3). Moreover, $v \notin \alpha(u)$ by (TL.2), and $v \notin \alpha(u')$ for any other $u' \in N_+^D(t)$ by (TL.4). Hence $v \in \beta(t)$.

To prove (4), let $t \triangleleft^D u$, and let $t = u_0, u_1, \ldots, u_m = u$ be a path from t to u in D. We prove $\gamma(u_i) \subseteq \gamma(t)$ and $\beta(t) \cap \gamma(u_i) \subseteq \sigma(u_i)$ by induction on $i \in [m]$. For $i = 1$, we have $\gamma(u_1) \subseteq \gamma(u_0) = \gamma(t)$ by (TL.3) and $\beta(t) \cap \gamma(u_1) = \sigma(u_1)$ by (3). For $i \geq 2$, we have $\gamma(u_i) \subseteq \gamma(u_{i-1}) \subseteq \gamma(t)$, where the first inclusion holds by (TL.3) and the second by the induction hypothesis. Furthermore, we have $\beta(t) \cap \gamma(u_i) \subseteq \sigma(u_{i-1}) \cap \gamma(u_i)$, because $\beta(t) \cap \gamma(u_{i-1}) \subseteq \sigma(u_{i-1})$ by the induction hypothesis and $\gamma(u_i) \subseteq \gamma(u_{i-1})$ by (TL.3). Hence

$$\begin{aligned}
\beta(t) \cap \gamma(u_i) &\subseteq \sigma(u_{i-1}) \cap \gamma(u_i) \\
&\subseteq \beta(u_{i-1}) \cap \gamma(u_i) && \text{(by (2))} \\
&= \sigma(u_i) && \text{(by (3)).}
\end{aligned}$$

□

Remark 3.4. Let us briefly discuss how treelike decompositions relate to tree decompositions. Let G be a connected graph. Then for every treelike decomposition (D, σ, α) of G there is a tree decomposition (T', β') of G and a homomorphism h from T' to D such that for all $t \in V(T')$ the nodes t and $h(t)$ have the same torsos in their respective decompositions.

Conversely, if (T, β) is a tree decomposition of G, and we define σ, α as described in the introduction to this section, then (T, σ, α) is a treelike decomposition of G.

There are minor technical complications in adapting these constructions to disconnected graphs G. A more detailed comparison between treelike decompositions and tree decompositions, which includes a treatment of disconnected graphs, and proofs of the claims made in this remark, can be found in Grohe [2012].

3.2. Decomposition Schemes and Definable Decompositions

In principle, we could use any logic to define treelike decomposition, but we will restrict our attention to inflationary fixed-point logic IFP here. We define decompositions by a form of transduction.

An IFP-*decomposition scheme* (for short: *d-scheme*) is a tuple

$$\Lambda = \big(\lambda_V(\bar{x}), \lambda_E(\bar{x}, \bar{x}'), \lambda_\sigma(\bar{x}, y), \lambda_\alpha(\bar{x}, y)\big)$$

of IFP-formulae in the vocabulary of graphs, where $\bar{x}, \bar{x}'$ are nonempty tuples of vertex variables of the same length and y is another vertex variable. The *dimension* of Λ is the length of the tuple $\bar{x}$.

In the following, let G be a graph and $\Lambda = \big(\lambda_V(\bar{x}), \lambda_E(\bar{x}, \bar{x}'), \lambda_\sigma(\bar{x}, y), \lambda_\alpha(\bar{x}, y)\big)$ an ℓ-dimensional d-scheme. We let

$$\begin{aligned} V\big(\Lambda[G]\big) &:= \lambda_V[G, \bar{x}], \\ E\big(\Lambda[G]\big) &:= \lambda_E[G, \bar{x}, \bar{x}'] \cap \big(V^{\Lambda[G]}\big)^2, \\ D^{\Lambda[G]} &:= (V(\Lambda[G]), E(\Lambda[G])), \end{aligned}$$

and for every $\bar{v} \in V\big(\Lambda[G]\big)$ we let

$$\begin{aligned} \sigma^{\Lambda[G]}(\bar{v}) &:= \lambda_\sigma[G, \bar{v}, y], \\ \alpha^{\Lambda[G]}(\bar{v}) &:= \lambda_\alpha[G, \bar{v}, y]. \end{aligned}$$

Then

$$\Lambda[G] := \big(D^{\Lambda[G]}, \sigma^{\Lambda[G]}, \alpha^{\Lambda[G]}\big)$$

is a decomposition of G, which we call the *decomposition defined by* Λ *on* G.

Example 3.5. It is easy to construct a d-scheme Λ such that for every cycle C it holds that $\Lambda[C] = \Delta(C)$, where $\Delta(C)$ is the decomposition introduced in Example 3.2.

Example 3.6. As a pathological example, let us decompose the empty graph $\emptyset$. Let $\Delta_\emptyset$ be the decomposition in which $D^{\Delta_\emptyset}$ is the empty digraph and $\sigma^{\Delta_\emptyset}, \alpha^{\Delta_\emptyset}$ are the empty mappings. Then $\Delta_\emptyset$ is a treelike decomposition of $\emptyset$. Curiously, for every d-scheme Λ it holds that $\Lambda[\emptyset] = \Delta_\emptyset$.

From now on, we shall ignore the empty graph when discussing decompositions.

The following example shows that the decomposition of a graph into its connected components is IFP-definable.

Example 3.7. There is a d-scheme Λ such that for all graphs G the decomposition $\Lambda[G]$ is treelike and its torsos are the connected components of G.

Indeed, let Λ be the d-scheme with $\lambda_V(x) := \mathsf{true}$, $\lambda_E(x, x') := \mathsf{false}$, $\lambda_\sigma(x, y) := \mathsf{false}$, and $\lambda_\alpha(x, y) := \mathsf{path}(x, y)$, where $\mathsf{path}(x, y)$ is an IFP-formula stating that there is a path from x to y.

Then for every graph G it holds that $V(\Lambda[G]) = V(G)$ and $E(\Lambda[G]) = \emptyset$. Furthermore, for every $v \in V(\Lambda[G])$, the set $\sigma(v)$ is empty, and $\alpha(v) = \gamma(v) = \beta(v)$ is the vertex set of the connected component of v. It is is easy to verify that the decomposition $\Lambda[G]$ is treelike.

A decomposition Δ of a graph G is *over* a class $\mathcal{A}$ if for all $t \in V(D^{\Delta})$ we have $\tau^{\Delta}(t) \in \mathcal{A}$.

LEMMA 3.8 (DEFINABILITY LIFTING LEMMA). *Let $\mathcal{A}$ be an* IFP*-definable class of graphs, and let Λ be a d-scheme. Then the class of all graphs G such that $\Lambda[G]$ is a treelike decomposition over $\mathcal{A}$ is* IFP*-definable.*

PROOF. Let φ be an IFP-sentence that defines the class $\mathcal{A}$, and let $\Lambda = (\lambda_V(\bar{x}), \lambda_E(\bar{x}, \bar{x}'), \lambda_\sigma(\bar{x}, y), \lambda_\alpha(\bar{x}, y))$.

We first prove that the cones, bags, and torsos of the decomposition defined by Λ are IFP-definable. We let

$$\begin{aligned}
\lambda_\gamma(\bar{x}, y) &:= \lambda_\sigma(\bar{x}, y) \vee \lambda_\alpha(\bar{x}, y),\\
\lambda_\beta(\bar{x}, y) &:= \lambda_\gamma(\bar{x}, y) \wedge \forall \bar{x}'\big(\lambda_E(\bar{x}, \bar{x}') \to \neg\lambda_\alpha(\bar{x}', y)\big),\\
\lambda_\tau(\bar{x}, y_1, y_2) &:= E(y_1, y_2) \vee \big(\lambda_\sigma(\bar{x}, y_1) \wedge \lambda_\sigma(\bar{x}, y_2)\big)\\
&\quad\vee \exists \bar{x}'\big(\lambda_E(\bar{x}, \bar{x}') \wedge \lambda_\sigma(\bar{x}', y_1) \wedge \lambda_\sigma(\bar{x}', y_2)\big).
\end{aligned}$$

Then for all graphs G and all $\bar{v} \in V(\Lambda[G])$ we have $\lambda_\gamma[G, \bar{v}, y] = \gamma^{\Lambda[G]}(\bar{v})$ and $\lambda_\beta[G, \bar{v}, y] = \beta^{\Lambda[G]}(\bar{v})$. Furthermore, for all $w_1, w_2 \in \beta^{\Lambda[G]}(\bar{v})$ we have $G \models \lambda_\tau[\bar{v}, w_1, w_2] \iff w_1w_2 \in E(\tau^{\Lambda[G]}(\bar{v}))$. Hence for the IFP-graph transduction $\Theta(\bar{x})$ with $\theta_{dom}(\bar{x}) := \lambda_V(\bar{x})$ and $\theta_V(\bar{x}, y) := \lambda_\beta(\bar{x}, y)$ and $\theta_E(\bar{x}, y_1, y_2) := \lambda_\tau(\bar{x}, y_1, y_2)$ we have $\Theta[G, \bar{v}] = \tau^{\Lambda[G]}(\bar{v})$. Then by the Transduction Lemma (Fact 2.9) there is a formula $\varphi^{-\Theta}(\bar{x})$ such that $G \models \varphi^{-\Theta}[\bar{v}] \iff \tau^{\Lambda[G]}(\bar{v}) \models \varphi$.

It is easy to write IFP-sentences $\psi_1, \ldots, \psi_5$ such that a graph G satisfies ψ_i if and only if the decomposition $\Lambda[G]$ satisfies (TL.i). For example, we let

$$\psi_3 := \forall x \forall \bar{x}'\Big(\lambda_E(\bar{x}, \bar{x}') \to \forall y\big((\lambda_\alpha(\bar{x}', y) \to \lambda_\alpha(\bar{x}, y)) \wedge (\lambda_\gamma(\bar{x}', y) \to \lambda_\gamma(\bar{x}, y))\big)\Big).$$

Then the formula

$$\psi_1 \wedge \cdots \wedge \psi_5 \wedge \forall \bar{x}\big(\lambda_V(\bar{x}) \to \varphi^{-\Theta}(\bar{x})\big)$$

defines the class of all graphs G such that $\Lambda[G]$ is a treelike decomposition of G over $\mathcal{A}$. $\square$

3.3. Decomposition into 3-Connected Components

In this subsection we prove a first nontrivial result about treelike decompositions. It says that the decompositions of graphs into their 3-connected components are IFP-definable. Recall that a graph G is *k-connected* if $|G| > k$ and for every set $S \subseteq V(G)$ of size $|S| < k$ the graph $G \setminus S$ is connected.

LEMMA 3.9 (3-CC DECOMPOSITION LEMMA). *There is a 3-dimensional d-scheme Λ such that for every graph G the decomposition $\Lambda[G]$ is treelike, and for all nodes $\bar{v} = (v_1, v_2, v_3)$ of $\Lambda[G]$ the following three conditions are satisfied.*

(i) *$\tau(\bar{v})$ is either 3-connected or a complete graph of order at most* 3.
(ii) *$\tau(\bar{v})$ is a minor of G.*
(iii) *$\sigma(\bar{v}) \subseteq \{v_1, v_2\}$, and $\sigma(\bar{v}) = \emptyset$ if and only if $\bar{v}$ is $\trianglelefteq$-minimal in $D^{\Lambda[G]}$.*

The proof requires some preparation. Let G be a graph and $W, X \subseteq V(G)$. A *proper W-X-separator* is a set $S \subseteq V(G)$ such that $W \not\subseteq S$ and $X \not\subseteq S$ and there is no path from a vertex in $W\backslash S$ to a vertex in $X\backslash S$ in $G\backslash S$. A *split extension* of W in G is a vertex $v \in V(G)$ such that for every connected component A of $G\backslash(W \cup \{v\})$ it holds that $N(A) \subset W \cup \{v\}$ (note the strict inclusion).

Example 3.10. Let C be a cycle of length at least 5 and $w_1w_2 \in E(C)$ an arbitrary edge. Let $W := \{w_1, w_2\}$. Then all vertices $v \in V(C)\backslash W$ are split extensions of W in C.

Now let v_i be the neighbor of w_i distinct from w_{3-i}, and let $S_i := \{v_i, w_{3-i}\}$. Then S_i is a separator of C, and the connected components of $C\backslash S_i$ are $C[\{w_i\}]$ and $A_i := C \setminus \{v_i, w_i, w_{3-i}\}$. Furthermore, there is no proper W-S_i-separator in C.

Note that we have $A_1 \cap A_2 = C \setminus \{v_1, v_2, w_1, w_2\} \neq \emptyset$ and $V(A_1) \cap S_2 = \{v_2\} \neq \emptyset$ and $V(A_2) \cap S_1 = \{v_1\} \neq \emptyset$.

LEMMA 3.11. *Let G be a graph and $W \subseteq V(G)$ a clique that has no split extension in G. Let S_1, S_2 be distinct subsets of $V(G)$ and $A_1, A_2 \subseteq G$ such that for $i = 1, 2$:*

(i) $W \not\subseteq S_i$*;*
(ii) $|S_i| \leq 2$*;*
(iii) *there is no proper W-S_i-separator of order at most 2;*
(iv) *A_i is a connected component of $G\backslash S_i$ with $W \cap V(A_i) = \emptyset$;*
(v) $N^G(A_i) = S_i$.

Then $V(A_1) \cap V(A_2) = \emptyset$ and $V(A_1) \cap S_2 = \emptyset$ and $S_1 \cap V(A_2) = \emptyset$.

We omit the proof. It can be found in Appendix B.

PROOF OF THE 3-CC DECOMPOSITION LEMMA 3.9 (SKETCH). To explain the definition of Λ, we fix a graph G. We shall describe a treelike decomposition $\Delta = (D, \sigma, \alpha)$ of G satisfying conditions (i)–(iii) of the lemma. It will be straightforward to construct a d-scheme Λ, which of course will not depend on G, such that $\Delta = \Lambda[G]$. The decomposition Δ will have three kinds of nodes: r-nodes (root nodes), s-nodes (split-extension nodes), and c-nodes (component-nodes). The torsos at the r-nodes and s-nodes will be complete graphs of order at most 3. The torsos at the c-nodes will either be complete graphs of order at most 3 or 3-connected graphs. Let $\bar{v} = (v_1, v_2, v_3) \in V(G)^3$.

(A) $\bar{v}$ is an *r-node* if $v_1 = v_2 = v_3$.

If $\bar{v}$ is an r-node, then we let $A_{\bar{v}}$ be the connected component of G that contains v_3, and we let $S_{\bar{v}} := \emptyset$. Now suppose that $v_3 \notin \{v_1, v_2\}$. In this case, we let $A_{\bar{v}}$ be the connected component of $G \setminus \{v_1, v_2\}$ that contains v_3. We let $S_{\bar{v}} := N^G(A_{\bar{v}})$ and let

$$H_{\bar{v}} := G\big[V(A_{\bar{v}} \cup S_{\bar{v}})\big] \cup K[S_{\bar{v}}].$$

A set $S \subseteq G$ is a *hinge* if it satisfies one of the following three conditions.

—$|S| = 1$.
—$|S| = 2$ and there is an edge between the two elements of S.
—$|S| = 2$ and there are distinct connected components A, A' of $G\backslash S$ such that $N^G(A) = N^G(A') = S$.

Observe that if $\{v_1, v_2\}$ is a hinge then $H_{\bar{v}}$ is a minor of G.

(B) $\bar{v}$ is an *s-node* if $v_3 \notin \{v_1, v_2\}$ and $S_{\bar{v}} = \{v_1, v_2\}$ and $S_{\bar{v}}$ is a hinge in G and v_3 is a split extension of $S_{\bar{v}}$ in $H_{\bar{v}}$.
(C) $\bar{v}$ is a *c-node* if $v_3 \notin \{v_1, v_2\}$ and $S_{\bar{v}} = \{v_1, v_2\}$ and $S_{\bar{v}}$ is a hinge in G and there is no split extension of $S_{\bar{v}}$ in $H_{\bar{v}}$.

We let V_r, V_s, V_c be the sets of r-nodes, s-nodes, and c-nodes, respectively, and $V(D) := V_r \cup V_s \cup V_c$. To define the edge relation $E(D)$, let $\bar{v} = (v_1, v_2, v_3)$, $\bar{w} = (w_1, w_2, w_3) \in V(D)$. Then $\bar{v}\bar{w} \in E(D)$ if one of the following conditions is satisfied.

(D) $\bar{v} \in V_r$ and $\bar{w} \in V_s \cup V_c$ and $S_{\bar{w}} = \widetilde{v} = \{v_1\}$.
(E) $\bar{v} \in V_s$ and $\bar{w} \in V_s \cup V_c$ and $S_{\bar{w}} \subseteq S_{\bar{v}} \cup \{v_3\}$ and $w_3 \in V(A_{\bar{v}}) \setminus \{v_3\}$.
(F) $\bar{v} \in V_c$ and $\bar{w} \in V_s \cup V_c$ and $S_{\bar{w}} \subseteq V(H_{\bar{v}})$ and $S_{\bar{v}} \not\subseteq S_{\bar{w}}$ and there is no proper $S_{\bar{v}}$-$S_{\bar{w}}$-separator of order at most 2 in $H_{\bar{v}}$ and $S_{\bar{w}}$ separates w_3 from $S_{\bar{v}}$ in $H_{\bar{v}}$.

We define $\sigma, \alpha : V(D) \to 2^{V(G)}$ as follows.

(G) For all $\bar{v} \in V(D)$ we let $\sigma(\bar{v}) := S_{\bar{v}}$ and $\alpha(\bar{v}) := V(A_{\bar{v}})$.

This completes the definition of the decomposition Δ. It is completely straightforward to construct a d-scheme Λ (not depending on the specific graph G) such that $\Delta = \Lambda[G]$.

It can be proved that Δ is a treelike decomposition. It is immediate from the definitions that Δ satisfies condition (iii). Furthermore, for every node $\bar{v} \in V(D)$ the separator $\sigma(\bar{v})$ is a hinge of G. This implies that for every node $\bar{v} \in V(D)$ the torso $\tau(\bar{v})$ is a minor of G. Hence Δ satisfies (ii) as well. To see that it satisfies (i), let $\bar{v} \in V(D)$. Depending on the type of node $\bar{v} = (v_1, v_2, v_3)$, the torso τ has the following form.

—If $\bar{v}$ is an r-node or an s-node then $\tau[\bar{v}] = K[\{v_1, v_2, v_3\}]$.
—If $\bar{v}$ is a c-node then $\tau[\bar{v}]$ is 3-connected or $K[\{v_1, v_2, v_3\}]$.

This implies (i). Details can be found in Appendix B. $\square$

3.4. Ordered Treelike Decompositions

An *o-decomposition* of a graph G is a quadruple $\Delta = (D^\Delta, \sigma^\Delta, \alpha^\Delta, \leqslant^\Delta)$ such that $(D^\Delta, \sigma^\Delta, \alpha^\Delta)$ is a decomposition of G and $\leqslant^\Delta$ is a mapping that associates a binary relation $\leqslant_t^\Delta \subseteq V(G)^2$ with every $t \in V(D^\Delta)$.

We call $(D^\Delta, \sigma^\Delta, \alpha^\Delta)$ the *underlying plain decomposition* of Δ. As usually, we omit the index $^\Delta$ if Δ is clear from the context.

Definition 3.12. An *ordered treelike decomposition* of a graph G is an o-decomposition $\Delta = (D, \sigma, \alpha, \leqslant)$ of G such that the underlying plain decomposition of Δ is treelike and the following axiom is satisfied.

(OTL) For every $t \in V(D)$ the relation $\leqslant_t$ is a linear order of $\beta(t)$.

An IFP-*o-decomposition scheme* (for short: *od-scheme*) is a tuple

$$\Lambda = \big(\lambda_V(\bar{x}), \lambda_E(\bar{x}, \bar{x}'), \lambda_\sigma(\bar{x}, y), \lambda_\alpha(\bar{x}, y), \lambda_\leqslant(\bar{x}, y_1, y_2)\big)$$

of IFP-formulae in the vocabulary of graphs, where $\bar{x}, \bar{x}'$ are tuples of vertex variables of the same length and y, y_1, y_2 are other vertex variables. The *dimension* of Λ is the length of the tuple $\bar{x}$.

Let G be a graph and $\Lambda = \big(\lambda_V(\bar{x}), \lambda_E(\bar{x}, \bar{x}'), \lambda_\sigma(\bar{x}, y), \lambda_\alpha(\bar{x}, y), \lambda_\leqslant(\bar{x}, y_1, y_2)\big)$ an od-scheme. For every $\bar{v} \in V\big(\Lambda[G]\big)$, we define $D^{\Lambda[G]}, \sigma^{\Lambda[G]}, \alpha^{\Lambda[G]}$ as before and let

$$\leqslant_{\bar{v}}^{\Lambda[G]} := \lambda_\leqslant[G, \bar{v}, y_1, y_2].$$

We call $\Lambda[G] := (D^{\Lambda[G]}, \sigma^{\Lambda[G]}, \alpha^{\Lambda[G]}, \leqslant^{\Lambda[G]})$ the *o-decomposition defined by* Λ *on* G.

$\mathcal{OT}_\Lambda$ denotes the class of all graphs G such that $\Lambda[G]$ is an ordered treelike decomposition of G. We say that a class $\mathcal{C}$ of graphs *admits* IFP-*definable ordered treelike decompositions* if there is an od-scheme Λ such that $\mathcal{C} \subseteq \mathcal{OT}_\Lambda$.

The following simple lemma may be viewed as a variant of the Definability Lifting Lemma 3.8. It is particularly useful in combination with the Canonization

Theorem 3.14, because it gives us a way to recognize whether a graph belongs to the domain of a canonization mapping definably in IFP.

LEMMA 3.13. *For every od-scheme Λ, the class $\mathcal{OT}_\Lambda$ is* IFP*-definable.*

PROOF. It is straightforward to formalize axioms (TL.1)–(TL.5) and (OTL) in IFP. □

The main reason for our interest in ordered definable treelike decompositions is the following theorem. It may be viewed as a far reaching generalization of the well-known result that trees admit IFP+C-definable canonization (going back to Immerman and Lander [1990]).

THEOREM 3.14 (CANONIZATION THEOREM). *Let $\mathcal{C}$ be a class of graphs that admits* IFP*-definable ordered treelike decompositions. Then $\mathcal{C}$ admits* IFP+C*-definable canonization.*

PROOF SKETCH. Suppose that we want to canonize a graph $G \in \mathcal{C}$. Let $\Delta = (D, \sigma, \alpha, \leqslant)$ be the ordered treelike decomposition of G that we have by assumption. By induction starting at the leaves, for every node $t \in V(D)$ we define a canonical ordered copy H_t of the graph $G_t := G[\gamma(t)]$. At a leaf t, there is nothing to do, because G_t is already ordered by $\leqslant_t$. For an inner node t, we take the graph $G[\beta(t)]$ and then add in lexicographical order copies of the graphs H_u for the children u of t. We need counting (and thus the logic IFP+C rather than just IFP) here, because some of the graphs H_u for children u of t may be isomorphic, and we need to determine how many isomorphic copies to include. To glue $G[\beta(t)]$ and the copies of H_u for the children u of t together, we need additional information on the location of the vertices in $\sigma(u)$ within H_u. But this is not a problem, as we may define H_u for every u in such a way that the vertices in $\sigma(u)$ are the first in the linear order.

The full proof is tedious, but follows this sketch. Details can be found in Appendix C. □

The following lemmas can be used to construct ordered treelike decompositions. The proofs of the first two lemmas are straightforward; for the following two we sketch proofs. Full proofs of all four lemmas can be found in Appendix D.

LEMMA 3.15. *Let $\mathcal{C}$ be a class of graphs that admits* IFP*-definable orders. Then $\mathcal{C}$ admits* IFP*-definable ordered treelike decompositions.*

LEMMA 3.16.

(1) *Let $\mathcal{B}, \mathcal{C}$ be classes of graphs that admit* IFP*-definable ordered treelike decompositions. Then $\mathcal{B} \cup \mathcal{C}$ admits* IFP*-definable ordered treelike decompositions.*
(2) *Every class of graphs that is finite up to isomorphism admits* IFP*-definable ordered treelike decompositions.*
(3) *Let $\mathcal{C}$ and $\mathcal{C}^*$ be classes of graphs such that the symmetric difference $\mathcal{C} \triangle \mathcal{C}^*$ is finite up to isomorphism. Then $\mathcal{C}$ admits* IFP*-definable ordered treelike decompositions if and only if $\mathcal{C}^*$ admits* IFP*-definable ordered treelike decompositions.*

It may seem surprising that in Lemma 3.16 (1) we need not impose any definability conditions on $\mathcal{B}$ and $\mathcal{C}$. The reason is that for an od-scheme Λ with $\mathcal{B} \subseteq \mathcal{OT}_\Lambda$ we can define $\mathcal{OT}_\Lambda$ in IFP (by Lemma 3.13), and this is sufficient for the proof. This nicely illustrates why definable ordered treelike decompositions are often easier to handle than definable canonizations. It is not necessarily the case that if classes $\mathcal{B}, \mathcal{C}$ admit IFP+C-definable canonization then their union admits IFP+C-definable canonization as well, simply because we cannot recognize (in an IFP-definable way) whether a transduction Θ that canonizes all graphs in $\mathcal{B}$ canonizes a given graph G (unless, of course, $\mathcal{B}$ is IFP-definable and $G \in \mathcal{B}$).

Like the Definability Lifting Lemma 3.8, the following lemma is a "lifting lemma", allowing it to "lift" properties from the torsos of a decomposition (in this case the existence of an ordered treelike decomposition) to the whole graph.

LEMMA 3.17 (3-CC LIFTING LEMMA). *Let $\mathcal{C}$ be a class of graphs that is closed under taking minors, and suppose that the class of all 3-connected graphs in $\mathcal{C}$ admits* IFP*-definable ordered treelike decompositions. Then $\mathcal{C}$ admits* IFP*-definable ordered treelike decompositions.*

PROOF SKETCH. By the 3-CC Decomposition Lemma 3.9, for all graphs G we have an IFP-definable treelike decomposition Δ into torsos that are either complete graphs of order at most 3 or 3-connected minors of G. If $G \in \mathcal{C}$ then by assumption, the torsos that are 3-connected minors of G are contained in $\mathcal{C}$, and hence they admit IFP-definable treelike decompositions. The torsos of order at most 3 come from a finite class of graphs, and hence by Lemma 3.16(2) they admit IFP-definable treelike decompositions as well. Thus by Lemma 3.16(1), all torsos of Δ admit IFP-definable treelike decompositions. We can lift the ordered treelike decompositions of the torsos to an ordered treelike decomposition of the whole graph by a product construction. $\square$

The following lemma and its corollary are "extension lemmas", which in certain situations allow us to extend ordered treelike decompositions from a subgraph of a graph to the whole graph.

LEMMA 3.18 (ORDERED EXTENSION LEMMA). *Let $\mathcal{A}, \mathcal{B}, \mathcal{C}$ be classes of graphs such that $\mathcal{A}$ admits* IFP*-definable orders and $\mathcal{B}$ admits* IFP*-definable ordered treelike decompositions. Let $\Theta(\bar{x})$ be a 1-dimensional* IFP*-graph transduction such that for all $G \in \mathcal{C}$ there is a tuple $\bar{p} \in G^{\bar{x}}$ with $(G, \bar{p}) \in \mathcal{D}_{\Theta(\bar{x})}$ and*

$$\Theta[G, \bar{p}] \in \mathcal{A} \quad \textit{and} \quad G \setminus \Theta[G, \bar{p}] \in \mathcal{B}.$$

Then $\mathcal{C}$ admits IFP*-definable ordered treelike decompositions.*

PROOF SKETCH. Let $\mathsf{ord}(\bar{y}, z_1, z_2)$ be an IFP-formula that defines an order on all graphs in $\mathcal{A}$, and let Λ be an od-scheme that defines an ordered treelike decomposition on all graphs in $\mathcal{B}$.

To explain the proof, we fix a graph $G \in \mathcal{C}$. For the sake of this proof sketch, let us assume that G is connected. Let $\bar{p} \in G^{\bar{x}}$ such that $(G, \bar{p}) \in \mathcal{D}_{\Theta(\bar{x})}$, and let $\bar{q} \in G^{\bar{y}}$. We let $H^{\bar{p}} := \Theta[G, \bar{p}]$ and $\leq_{\bar{p}\bar{q}} := \mathsf{ord}[H^{\bar{p}}, \bar{q}, y_1, y_2]$. Note that $\leq_{\bar{p}\bar{q}}$ is not necessarily a linear order of $H^{\bar{p}}$, because not all choices of $\bar{p}$ yield a structure $H^{\bar{p}} \in \mathcal{A}$, and even if $H^{\bar{p}} \in \mathcal{A}$ the binary relation $\leq_{\bar{p}\bar{q}}$ is a linear order only for some choices of the parameters $\bar{q}$. But there is an IFP-formula $\mathsf{check\text{-}ord}(\bar{x}, \bar{y})$ such that $G \models \mathsf{check\text{-}ord}[\bar{p}, \bar{q}]$ if any only if $\leq_{\bar{p}\bar{q}}$ is a linear order of $V(H^{\bar{p}})$. Let $G^{\bar{p}} := G \setminus H^{\bar{p}}$ and $\Delta^{\bar{p}} = (D^{\bar{p}}, \sigma^{\bar{p}}, \alpha^{\bar{p}}, \leqslant^{\bar{p}}) := \Lambda[G^{\bar{p}}]$. Note that $\Delta^{\bar{p}}$ is not necessarily an ordered treelike decomposition of $G^{\bar{p}}$, because $G^{\bar{p}}$ is not contained in $\mathcal{B}$ for all $\bar{p}$. But there is an IFP-formula $\mathsf{check\text{-}dec}(\bar{x})$ such that $G \models \mathsf{check\text{-}dec}[\bar{p}]$ if any only if $G^{\bar{p}} \in \mathcal{OT}_{\Lambda}$. Let P be the set of all $\bar{p}\bar{q} \in G^{\bar{x}\bar{y}}$ such that $(G, \bar{p}) \in \mathcal{D}_{\Theta(\bar{x})}$ and $\leq_{\bar{p}\bar{q}}$ is a linear order of $H^{\bar{p}}$ and $G^{\bar{p}} \in \mathcal{OT}_{\Lambda}$. By the assumptions of the lemma, P is nonempty. In the following we denote elements of P by p, p'. If $p = \bar{p}\bar{q}$ we let $H^p := H^{\bar{p}}$ and $\leq_p := \leq_{\bar{p},\bar{q}}$ and $G^p := G^{\bar{p}}$ and $\Delta^p = (D^p, \sigma^p, \alpha^p, \leqslant^p) := \Delta^{\bar{p}}$.

Then we define an ordered treelike decomposition $\Delta = (D, \sigma, \alpha, \leqslant)$ of G as follows. To construct the digraph D, we take the disjoint union of the digraphs D^p for all $p \in P$. Let us call the nodes $t \in V(D^p)$ for $p \in P$ the *internal nodes* of D. For each $p \in P$, we add a new *external node* x_p, and we add edges from x_p to all internal nodes $t \in V(D^p)$ such that $\sigma^p(t) = \emptyset$ and $\alpha^p(t)$ is the vertex set of a connected component of G^p. For the external nodes, we let $\sigma(x_p) := \emptyset$ and $\alpha(x_p) := V(G)$. For the internal nodes $t \in V(D^p)$, we let $\sigma(t) := \sigma^p(t) \cup V(H^p)$ and $\alpha(t) := \alpha^p(t)$.

It can be shown that (D, σ, α) is a treelike decomposition of G. Furthermore, for each external node x_p we have $\beta(x_p) = V(H^p)$ and for each internal node $t \in V(D^p)$ we have $\beta(t) = \beta^p(t) \cup V(H^p)$. Hence we can order the bags by concatenating the linear orders $\leq_p$ of $V(H^p)$ and $\leqslant_t^p$ of $\beta^p(t)$. □

A *k-extension* of a graph G is an extension of G by at most k vertices, that is, a graph H such that $G = H \backslash X$ for some $X \subseteq V(H)$ with $|X| \leq k$.

COROLLARY 3.19 (FINITE EXTENSION LEMMA). *Let $\mathcal{C}$ be a class of graphs that admits* IFP*-definable ordered treelike decompositions. Then for every $k \geq 1$, the class of all k-extensions of the graphs in $\mathcal{C}$ admits* IFP*-definable ordered treelike decompositions.*

3.5. Graphs of Bounded Tree Width

Let G be a graph. The *width* of a tree decomposition (T, β) of G is $\max\{|\beta(t)| \mid t \in V(T)\} - 1$. Similarly, the *width* of a treelike decomposition (D, σ, α) of G is defined to be $\max\{|\beta(t)| \mid t \in V(D)\} - 1$. It can be shown that G has a tree decomposition of width k if and only it has a treelike decomposition of width k (refer to Remark 3.4). The *tree width* of G is the minimum of the widths of all tree decompositions of G.

THEOREM 3.20. *For every $k \in \mathbb{N}$ there is a d-scheme Λ such that for every graph G of tree width at most k, the decomposition $\Lambda[G]$ is a treelike decomposition of G of width at most k. Furthermore, for all $\bar{v} \in V(\Lambda[G])$ it holds that $\beta(\bar{v}) \subseteq \widetilde{v}$.*

PROOF SKETCH. Let G be a graph of tree width at most k. For the sake of this proof sketch, we assume that G is connected. In the first step of the proof we define a treelike decomposition Δ' of G, and in the second step we "prune" Δ' to obtain a treelike decomposition Δ of width at most k. It will be easy to see that both Δ' and Δ are IFP-definable by d-schemes Λ', Λ, respectively, that do not depend on G.

The nodes of $\Lambda' = (D', \sigma', \alpha')$ will be tuples $\bar{v} = \bar{v}_I\bar{v}_{II} \in V(G)^{2(k+1)}$. There will be two kinds of nodes, r-nodes (root nodes) and c-nodes (child nodes). A tuple $\bar{v} \in V(G)^{2(k+1)}$ is an r-node if $\bar{v}_I = \bar{v}_{II}$. A tuple $\bar{v} \in V(G)^{2(k+1)}$ is a c-node if $\widetilde{v}_{II} \setminus \widetilde{v}_I \neq \emptyset$ and there is a connected component A of $G \setminus \widetilde{v}_I$ such that $\widetilde{v}_{II} \setminus \widetilde{v}_I \subseteq V(A)$ and $N^G(A) = \widetilde{v}_I \cap \widetilde{v}_{II}$.

For every r-node $\bar{v}$, we let $\alpha'(\bar{v}) := V(G)$ and $\sigma'(\bar{v}) := \emptyset$. For every c-node $\bar{v}$, we let $\alpha'(\bar{v})$ be the vertex set of the connected component of $G \setminus \widetilde{v}_I$ that contains $\widetilde{v}_{II} \setminus \widetilde{v}_I$, and we let $\sigma'(\bar{v}) := N^G(\alpha'(\bar{v})) = \widetilde{v}_I \cap \widetilde{v}_{II}$.

For nodes $\bar{v}, \bar{w} \in V(D')$ we let $\bar{v}\bar{w} \in E(D')$ if $\bar{w}$ is a c-node and $\bar{w}_I = \bar{v}_{II}$ and $\alpha'(\bar{w}) \subseteq \alpha'(\bar{v})$.

It is not hard to prove that the decomposition Δ' is treelike.

In the second step of the proof we define a decomposition $\Delta = (D, \sigma, \alpha)$, where D is an induced subgraph of D' and σ, α are the restrictions of σ', α' to $V(D)$. We inductively define a sequence of sets $U_h \subseteq V(D)$, for all $h \in \mathbb{N}^+$:

—U_1 consists of all $\bar{v} \in V(D)$ such that $\gamma'(\bar{v}) = \widetilde{v}_{II}$.
—U_{h+1} consists of all $\bar{v} \in V(D')$ such that for every connected component A of $G[\gamma'(\bar{v})] \setminus \widetilde{v}_{II}$ there is a node $\bar{w} \in N_+^{D'}(\bar{v}) \cap U_h$ with $\alpha'(\bar{w}) = V(A)$.

The *height* of a tree decomposition (T, β) of a graph H is the height of the tree T. The following claim is crucial.

CLAIM 1. *Let $\bar{v} \in V(D')$ such that there is a tree decomposition (T, β) of $G[\gamma'(\bar{v})]$ of height at most h and of width at most k with $\beta(r) = \widetilde{v}_{II}$ for the root r of T. Then $\bar{v} \in U_{h+1}$.*
The claim is proved by induction on h.

Now we let D be the induced subgraph of D' with vertex set $\bigcup_{h \in \mathbb{N}^+} U_h$. It follows from Claim 1 and the fact that there is a tree decomposition of G of width at most k that

$\Delta = (D, \sigma, \alpha)$ is a treelike decomposition of G. It follows from the definition of the sets U_h that for all nodes $\bar{v} \in V(D)$ we have $\beta(\bar{v}) \subseteq \widetilde{v}_{II}$ and thus $|\beta(\bar{v})| \leq k + 1$.

The full proof can be found in Appendix E. □

Once we have defined a treelike decomposition $\Lambda[G]$ with $\beta(\bar{v}) \subseteq \widetilde{v}$ for all nodes $\bar{v}$, we can directly turn it into an ordered treelike decomposition by ordering the vertices of $\beta(\bar{v})$ according to their first appearance in the tuple $\bar{v}$. Hence we obtain the following corollary.

COROLLARY 3.21. *For every $k \in \mathbb{N}$, the class of all graphs of tree width at most k admits* IFP*-definable ordered treelike decompositions.*

Via the Canonization Theorem 3.14, this implies the following result.

COROLLARY 3.22 (GROHE AND MARIÑO [1999]). *For every $k \in \mathbb{N}$,* IFP+C *captures* PTIME *on the class of all graphs of tree width at most k.*

4. BACKGROUND ON SURFACES AND EMBEDDINGS OF GRAPHS

In this section, we review the definitions and some of the most basic facts about graph embeddings. For more background, I refer the reader to Mohar and Thomassen [2001]. For planar graphs, Diestel [2005] is also a good reference. Appendix B of Diestel [2005] is a short introduction to surface topology that covers everything needed here (in slightly different terminology).

4.1. Topological Prerequisites

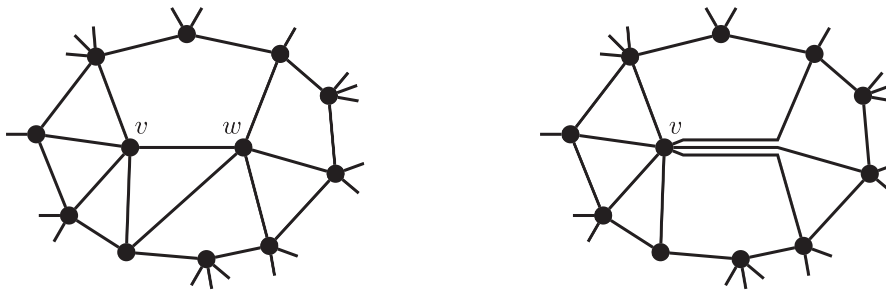
We denote topological spaces by bold-face letters. In particular, $\boldsymbol{R}^n$ denotes the Euclidean n-space, $\boldsymbol{S}^n$ denotes the n-sphere, and $\boldsymbol{B}^n$ denotes the closed unit ball in $\boldsymbol{R}^n$. In our notation, we do not distinguish between a topological space and its ground set, and we equip subsets of topological spaces with the usual induced topology and view them as subspaces. The boundary of a subspace (or subset) $\boldsymbol{Y}$ of a topological space $\boldsymbol{X}$ is denoted by $\boldsymbol{bd_X}(\boldsymbol{Y})$. The interior of $\boldsymbol{Y}$ is denoted by $\boldsymbol{int_X}(\boldsymbol{Y})$, and the closure $\boldsymbol{Y} \cup \boldsymbol{bd_X}(\boldsymbol{Y})$ of $\boldsymbol{Y}$ is denoted by $\boldsymbol{cl_X}(\boldsymbol{Y})$. We omit the subscript $_{\boldsymbol{X}}$ if $\boldsymbol{X}$ is clear from the context. A *closed disk* is a space homeomorphic to $\boldsymbol{B}^2$, and an *open disk* is a space homeomorphic to $\boldsymbol{int}(\boldsymbol{B}^2)$. A *simple curve* in a topological space $\boldsymbol{X}$ is a homeomorphic image of the unit interval $\boldsymbol{[0, 1]}$ (with the usual topology) in $\boldsymbol{X}$. The *endpoints* and *internal points* of a simple curve are defined in the obvious way, and two simple curves are *internally disjoint* if they share no internal points. A *simple closed curve* in $\boldsymbol{X}$ is a homeomorphic image of the 1-sphere $\boldsymbol{S}^1$ in $\boldsymbol{X}$. Let $\boldsymbol{g}$ be a simple closed curve in the 2-sphere $\boldsymbol{S}^2$. By the *Jordan Curve Theorem*, $\boldsymbol{S}^2 \setminus \boldsymbol{g}$ has precisely two arcwise connected components $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, and $\boldsymbol{bd}(\boldsymbol{A}_1) = \boldsymbol{bd}(\boldsymbol{A}_2) = \boldsymbol{g}$. A consequence of the Jordan Curve Theorem is the following useful fact (Lemma 4.1.2 of Diestel [2005]).

FACT 4.1. *Let $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3$ be three simple curves in the 2-sphere $\boldsymbol{S}^2$ that have the same endpoints v_1, v_2 and mutually disjoint interiors. Then $\boldsymbol{S}^2 \setminus (\boldsymbol{g}_1 \cup \boldsymbol{g}_2 \cup \boldsymbol{g}_2)$ has precisely three arcwise connected components $\boldsymbol{f}_{12}, \boldsymbol{f}_{13}, \boldsymbol{f}_{23}$ with boundaries $\boldsymbol{g}_1 \cup \boldsymbol{g}_2$, $\boldsymbol{g}_1 \cup \boldsymbol{g}_3$, $\boldsymbol{g}_2 \cup \boldsymbol{g}_3$, respectively.*

Furthermore, if $\boldsymbol{h}$ is another simple curve with endpoints x_1 in the interior of $\boldsymbol{g}_1$ and x_2 in the interior of $\boldsymbol{g}_2$ such that $\boldsymbol{h} \cap \boldsymbol{f}_{12} = \emptyset$ then $\boldsymbol{h} \cap \boldsymbol{g}_3 \neq \emptyset$.

4.2. Graph Embeddings

Let $\boldsymbol{X}$ be a topological space. An *embedded graph* in $\boldsymbol{X}$ is a pair $G = \big(V(G), E(G)\big)$ where $V(G)$ is a finite subset of $\boldsymbol{X}$ and $E(G)$ a set of simple curves in $\boldsymbol{X}$ such that for all $\boldsymbol{e} \in E(G)$, both endpoints and no interior points of $\boldsymbol{e}$ are in $V(G)$, and any two distinct $\boldsymbol{e}, \boldsymbol{e}' \in E(G)$ have at most one endpoint and no interior points in common. By $\boldsymbol{G}$ we



(a) (b)

Fig. 4.1. Edge contraction in an embedded graph.

denote both the set $V(G) \cup \bigcup_{\boldsymbol{e} \in E(G)} \boldsymbol{e}$ and the subspace of $\boldsymbol{X}$ it induces; recall that we do not distinguish between a topological space and its ground set in our notation. The *faces* of G are the arcwise connected components of $\boldsymbol{X} \setminus \boldsymbol{G}$. We denote the set of all faces of an embedded graph G by $F(G)$. A vertex $v \in V(G)$ is *incident* to a face $\boldsymbol{f} \in F(G)$, if $v \in \boldsymbol{bd}(\boldsymbol{f})$. An edge $\boldsymbol{e} \in E(G)$ is *incident* to $\boldsymbol{f}$ if $\boldsymbol{bd}(\boldsymbol{f})$ contains an interior point of $\boldsymbol{e}$. Subgraphs of embedded graphs are defined in the natural way. They are also viewed as embedded graphs.

The *underlying graph* of an embedded graph G is the graph with vertex set $V(G)$ and edge set $\{vw \in V(G)^2 \mid \exists \boldsymbol{e} \in E(\boldsymbol{G}) : \boldsymbol{e} \cap V(G) = \{v, w\}\}$. We make no notational distinction between an embedded graph and its underlying "abstract" graph; it will always be clear from the context whether we refer to the embedded graph or its underlying graph.

An *embedding* of a graph G in $\boldsymbol{X}$ is an isomorphism from G to a graph G' embedded in $\boldsymbol{X}$, and G is *embeddable* in $\boldsymbol{X}$ if it has an embedding in $\boldsymbol{X}$.

4.3. Plane and Planar Graphs

A *plane graph* is a graph embedded in the 2-sphere $\boldsymbol{S}^2$. Of course it would be more natural to call graphs embedded in $\boldsymbol{R}^2$ "plane". However, a graph is embeddable in $\boldsymbol{S}^2$ if any only if it is embeddable in $\boldsymbol{R}^2$, and it will be more convenient for us to work with embeddings in $\boldsymbol{S}^2$. A *planar graph* is a graph that is embeddable in $\boldsymbol{S}^2$.

In the following, let G be a plane graph. Then an edge $\boldsymbol{e} \in E(G)$ is incident with a face $\boldsymbol{f} \in F(G)$ if any only if $\boldsymbol{e} \subseteq \boldsymbol{bd}(\boldsymbol{f})$. Hence for every face $\boldsymbol{f} \in F(G)$ there is a subgraph $H \subseteq G$ such that $\boldsymbol{H} = \boldsymbol{bd}(\boldsymbol{f})$. We denote H by $Bd(\boldsymbol{f})$ and call it the *boundary subgraph* of $\boldsymbol{f}$. We call a subgraph $H \subseteq G$ a *facial subgraph* if it is the boundary subgraph of some face $\boldsymbol{f} \in F(G)$. Every edge $\boldsymbol{e} \in E(G)$ is incident to at least one and at most two faces. It can be shown that if G is connected, all its facial subgraphs are connected. If G is 2-connected, then all its facial subgraphs of G are cycles (we call them *facial cycles*), and every edge is contained in precisely two facial cycles. (See Section 4.2 of Diestel [2005] for proofs of the facts mentioned in this paragraph.)

The edges incident with a vertex v of a plane graph have a natural cyclic order in which for any two successive edges $\boldsymbol{e} = vw, \boldsymbol{e}' = vw'$ there is a face $\boldsymbol{f}$ that is incident with both $\boldsymbol{e}, \boldsymbol{e}'$. We will discuss this cyclic order in more detail in Section 4.5 for graphs embedded in arbitrary surfaces.

Clearly, every subgraph of a planar graph is planar. Figure 4.1 shows that contracting an edge in a planar graph preserves planarity. Hence the class of planar graphs is closed under taking minors.

3-connected planar graphs play a special role, because by a theorem of Whitney's [1932], any two embeddings of a 3-connected planar graph in the sphere $\boldsymbol{S}^2$ are

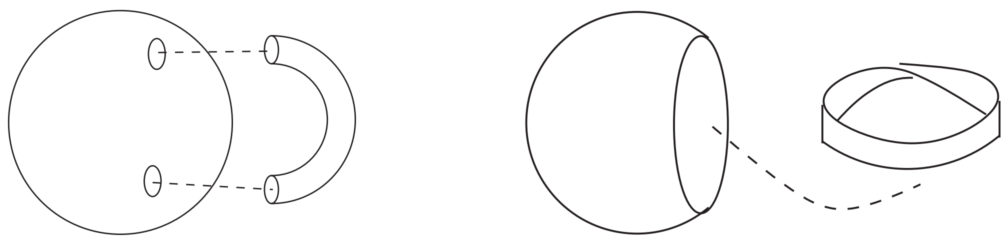


Fig. 4.2. Adding a handle and a crosscap.

homeomorphic, that is, topologically indistinguishable.[2] The "combinatorial core" of this result is the following fact, which describes the facial cycles of a 3-connected plane graph in purely graph-theoretic terms, not referring to the embedding. A cycle $C \subseteq G$ is *nonseparating* if $G \setminus C$ is either empty or connected. C is *chordless* if it is an induced subgraph of G.

FACT 4.2 [WHITNEY 1932]. *Let G be a 3-connected plane graph and $C \subseteq G$. Then C is a facial subgraph of G if and only if C is a chordless and nonseparating cycle.*

(See Diestel [2005], Proposition 4.2.7 for a proof.)

The following fact helps us to deal with planar graphs that are not 3-connected.

FACT 4.3. *Let G, H be two planar graphs such that either $|G \cap H| \leq 1$ or $G \cap H = K_2$, that is, $|G \cap H| = 2$ and the two vertices in $V(G) \cap V(H)$ are adjacent in both G and H. Then $G \cup H$ is planar.*

(See Gross and Tucker [1987] , Theorem 1.6.7] for a proof.)

The readers who want to proceed to our main results for planar graphs before delving into embeddings of graphs in arbitrary surfaces may skip the rest of this section.

4.4. Surfaces

A *surface* is an arcwise connected compact 2-manifold. Unless explicitly stated otherwise, surfaces in this article are without boundary. The classification theorem for surfaces (see, for example, Theorem 3.1.3 of Mohar and Thomassen [2001]) says that, up to homeomorphism, there are only two infinite families $(\boldsymbol{S}_g)_{g \in \mathbb{N}}$ and $(\boldsymbol{N}_h)_{h \in \mathbb{N}^+}$ of surfaces constructed in a fairly simple way. Every orientable surface $\boldsymbol{S}$ is homeomorphic to a surface $\boldsymbol{S}_g$, for some $g \geq 0$, obtained from the 2-sphere by adding g handles,[3] and every nonorientable surface $\boldsymbol{S}$ is homeomorphic to a surface $\boldsymbol{N}_h$, for some $h \geq 1$, obtained from the 2-sphere by adding h crosscaps (see Figure 4.2). The number g or h, respectively, is called the *(orientable or nonorientable) genus* of $\boldsymbol{S}$. It is convenient to define the *Euler genus* $\mathrm{eg}(\boldsymbol{S})$ of a surface $\boldsymbol{S}$ to be $2g$ if $\boldsymbol{S}$ is homeomorphic to $\boldsymbol{S}_g$ and h if $\boldsymbol{S}$ is homeomorphic to $\boldsymbol{N}_h$.

Example 4.4. $\boldsymbol{S}_1$ is a torus (and every torus is homeomorphic to $\boldsymbol{S}_1$). $\boldsymbol{S}_2$ is a double torus, $\boldsymbol{S}_3$ is a triple torus (also known as "pretzel"), et cetera. $\boldsymbol{N}_1$ is a projective plane, and $\boldsymbol{N}_2$ a Klein bottle. (Illustrations of all these surfaces can easily be found on the Web.)

[2]As we are not going to use this result here, we do not not have to worry about its exact meaning. See Section 4.3 of Diestel [2005] for a proof.

[3]For $g = 0$, this is just the 2-sphere $\boldsymbol{S}^2$. Thus $\boldsymbol{S}_0$ denotes the 2-sphere as well, but we usually denote it by $\boldsymbol{S}^2$.

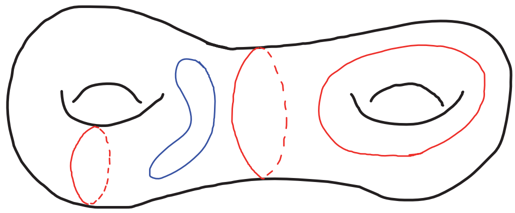


Fig. 4.3. Simple closed curves on a double torus.

A simple closed curve $\boldsymbol{g}$ in a surface $\boldsymbol{S}$ is *nonseparating* if $\boldsymbol{S} \setminus \boldsymbol{g}$ is arcwise connected, and *separating otherwise*. It is *contractible* if it is the boundary of a closed disk in $\boldsymbol{S}$, and *noncontractible* otherwise.[4]

Example 4.5. Figure 4.3 shows four simple closed curves on a double torus. Two of them are separating, and one is contractible.

FACT 4.6. *Every surface of positive Euler genus (that is, every surface except the 2-sphere) contains a nonseparating simple closed curve.*

(This is Lemma B.1 of Diestel [2005]).

The following fact summarizes the properties of (non)separating simple closed curves in surfaces most important for us (the fact combines Lemmas B.4 and B.5 of Diestel [2005]).

FACT 4.7. *Let $\boldsymbol{S}$ be a surface and $\boldsymbol{g}$ a simple closed curve in $\boldsymbol{S}$.*

(1) *If $\boldsymbol{g}$ is nonseparating then it is noncontractible, and there is a surface $\boldsymbol{S}'$ such that $\boldsymbol{S} \setminus \boldsymbol{g}$ is homeomorphic to a space obtained from $\boldsymbol{S}'$ by deleting one or two disjoint closed disks.*
Furthermore, $\operatorname{eg}(\boldsymbol{S}') < \operatorname{eg}(\boldsymbol{S})$.
(2) *If $\boldsymbol{g}$ is separating, then $\boldsymbol{S} \setminus \boldsymbol{g}$ has precisely two arcwise connected components $\boldsymbol{A}_1, \boldsymbol{A}_2$, and for $i = 1, 2$, there is a surface $\boldsymbol{S}_i$ such that $\boldsymbol{A}_i$ is homeomorphic to a space obtained from $\boldsymbol{S}_i$ by deleting a closed disk.*
Furthermore, $\operatorname{eg}(\boldsymbol{S}_1) + \operatorname{eg}(\boldsymbol{S}_2) = \operatorname{eg}(\boldsymbol{S})$.

The surfaces $\boldsymbol{S}'$ in (1) and $\boldsymbol{S}_1, \boldsymbol{S}_2$ in (2) are unique up to homeomorphism. We call them the surface(s) obtained from $\boldsymbol{S}$ by *cutting along $\boldsymbol{g}$ and capping the holes* (see Appendix B of Diestel [2005] for a description of the construction of $\boldsymbol{S}'$ or $\boldsymbol{S}_1, \boldsymbol{S}_2$ from $\boldsymbol{S}$ and $\boldsymbol{g}$).

Note that a contractible curve $\boldsymbol{g}$ in a surface $\boldsymbol{S}$ is separating, and one of the surfaces $\boldsymbol{S}_1, \boldsymbol{S}_2$ obtained from $\boldsymbol{S}$ by cutting along $\boldsymbol{g}$ and capping the holes is a sphere, because a sphere is the only surface $\boldsymbol{S}'$ such that deleting a closed disk from it yields an open disk. If $\boldsymbol{S}_1$ is a sphere, then $\operatorname{eg}(\boldsymbol{S}_1) = 0$ and thus $\operatorname{eg}(\boldsymbol{S}_2) = \operatorname{eg}(\boldsymbol{S})$. Facts 4.6 and 4.7 have the following consequences for noncontractible curves.

COROLLARY 4.8. *Let $\boldsymbol{S}$ be a surface of positive Euler genus.*

(1) *$\boldsymbol{S}$ has a noncontractible simple closed curve.*
(2) *Let $\boldsymbol{g}$ be a noncontractible simple closed curve in $\boldsymbol{S}$. Then either $\boldsymbol{g}$ is separating, and the two surfaces $\boldsymbol{S}_1, \boldsymbol{S}_2$ obtained from $\boldsymbol{S}$ by cutting along $\boldsymbol{g}$ and capping the holes*

[4]The standard definition of contractibility says that $\boldsymbol{g}$ is contractible if there is a continuous function $g : \boldsymbol{S}^1 \to \boldsymbol{S}$ that is a homeomorphism from $\boldsymbol{S}^1$ onto $\boldsymbol{g}$ and that is homotopic to a constant function. The two definitions are equivalent by a result of Levine [1963].

have smaller Euler genus than $\boldsymbol{S}$, or $\boldsymbol{g}$ is nonseparating, and the surface $\boldsymbol{S}'$ obtained from $\boldsymbol{S}$ by cutting along $\boldsymbol{g}$ and capping the holes has smaller Euler genus than $\boldsymbol{S}$.

4.5. Graphs in Surfaces

Some basic facts about embeddings directly generalize from plane graphs to graphs embedded in an arbitrary surface. Let G be a graph embedded in a surface $\boldsymbol{S}$. Then an edge $\boldsymbol{e} \in E(G)$ is incident with a face $\boldsymbol{f} \in F(G)$ if any only if $\boldsymbol{e} \subseteq \boldsymbol{bd}(\boldsymbol{f})$. Hence for every face $\boldsymbol{f} \in F(G)$ there is a *boundary subgraph* $Bd(\boldsymbol{f}) =: H$ such that $\boldsymbol{H} = \boldsymbol{bd}(\boldsymbol{f})$. Furthermore, every edge $\boldsymbol{e} \in E(G)$ is incident with at least one and at most two faces.

For every vertex $v \in V(G)$ there is a cyclic permutation π_v of the set $E(v)$ of all edges incident with v, such that in any "sufficiently small" closed disk around v the edges intersect the boundary of the disk in the cyclic order given by π_v. More precisely, if $\boldsymbol{D} \subseteq \boldsymbol{S}$ is a closed disk with $v \in \boldsymbol{D}$ and $\boldsymbol{e} \cap \boldsymbol{bd}(\boldsymbol{D}) \neq \emptyset$ for every $\boldsymbol{e} \in E(v)$, and if for every $\boldsymbol{e} \in E(v)$ we let $\boldsymbol{e}_0$ be the segment of $\boldsymbol{e}$ from v to $\boldsymbol{bd}(\boldsymbol{D})$ and $x_{\boldsymbol{e}} \in \boldsymbol{bd}(\boldsymbol{D})$ be the endpoint $\boldsymbol{e}_0$, then the vertices $x_{\boldsymbol{e}}$ appear on $\boldsymbol{bd}(\boldsymbol{D})$ in the cyclic order given by π_v (see Lemma 3.2.2 of Mohar and Thomassen [2001] for a proof that this order does not depend on the disk $\boldsymbol{D}$). Clearly, for all edges $\boldsymbol{e} = vw$ there is a face $\boldsymbol{f} \in F(G)$ such that both $\boldsymbol{e}$ and $\pi_v(\boldsymbol{e})$ are incident with $\boldsymbol{f}$; just take the face $\boldsymbol{f} \in F(G)$ that contains the segment of $\boldsymbol{bd}(\boldsymbol{D}) \setminus \{x_{\boldsymbol{e}'} \mid \boldsymbol{e}' \in E(v)\}$ from $x_{\boldsymbol{e}}$ to $x_{\pi_v(\boldsymbol{e})}$. (This face $\boldsymbol{f}$ is not unique if $E(v) = \{\boldsymbol{e}, \pi_v(\boldsymbol{e})\}$.) By a similar argument, for every face $\boldsymbol{f}$ incident with v and every edge $\boldsymbol{e} \in E(v)$ incident with $\boldsymbol{f}$, either $\pi_v(\boldsymbol{e})$ is incident with $\boldsymbol{f}$ or $\pi_v^{-1}(\boldsymbol{e})$ is incident with $\boldsymbol{f}$.

The cyclic orders around the vertices are often used to give a purely combinatorial definition of embeddings of graphs in surfaces (see Chapter 4 of Mohar and Thomassen [2001]). We can also use them to define *facial walks*. Let $\boldsymbol{f}$ be a face of a graph G embedded in a surface, B a connected component of $Bd(\boldsymbol{f})$, and $\boldsymbol{e}_1 = v_1v_2 \in E(B)$. We define a closed walk $v_1\boldsymbol{e}_1v_2\boldsymbol{e}_2v_3 \dots v_{m-1}\boldsymbol{e}_{m-1}v_m\boldsymbol{e}_mv_{m+1} = v_1$ by letting $e_{i+1} \in \{\{\pi_{v_{i+1}}(\boldsymbol{e}_i), \pi^{-1}_{v_{i+1}}(\boldsymbol{e}_i)\}$ such that $\boldsymbol{e}_{i+1}$ is incident with $\boldsymbol{f}$.[5] We stop the walk as soon as the next edge would be $\boldsymbol{e}_1 = v_1v_2$ again in the same direction, so that we have $(v_i, v_{i+1}) \neq (v_1, v_2)$ for all $i \in [2, m]$. Then $V(B) = \{v_1, \dots, v_m\}$ and $E(B) = \{e_1, \dots, e_m\}$. Every edge $\boldsymbol{e}$ may appear at most twice in this walk (once in each direction), and if it appears twice then $\boldsymbol{f}$ is the only face incident with $\boldsymbol{e}$. Of course, if $\boldsymbol{bd}(\boldsymbol{f})$ is connected, then modulo orientation and starting edge there is only one facial walk associated with $\boldsymbol{f}$.

Observe that, as the class of planar graphs, for every surface $\boldsymbol{S}$ the class of all graphs embeddable in $\boldsymbol{S}$ is closed under taking minors. Figure 4.1 illustrates why.

Let us now turn to the differences between plane graphs and graphs embedded in higher surfaces. Facial subgraphs can be much more complicated. They can be disconnected even if the graph is connected, and they are not necessarily cycles even if the graph is 3-connected. The faces can also be more complicated spaces than just open disks as for connected plane graphs. A graph G is *2-cell embedded* in a surface $\boldsymbol{S}$ if all faces $\boldsymbol{f} \in F(G)$ are open disks. An embedding of a graph in $\boldsymbol{S}$ is *2-cell* if its image is 2-cell embedded in $\boldsymbol{S}$. Every 2-cell embedded graph is connected.

Example 4.9. Figure 4.4(a) shows an embedding of K_4 in the torus that is not 2-cell, because it has a face homeomorphic to a cylinder.

Figure 4.4(b) shows an embedding of K_5 in the torus that is 2-cell, but has a facial subgraph that is not a cycle. This example also shows that even if a graph is 2-cell embedded in a surface, it is not necessarily the case that the closure of each face is a closed disk.

[5]There is a slight ambiguity here because it may happen that both $\pi_{v_{i+1}}(\boldsymbol{e}_i)$ and $\pi^{-1}_{v_{i+1}}(\boldsymbol{e}_i)$ are incident with $\boldsymbol{f}$. This can be resolved, but is not so important for this article.

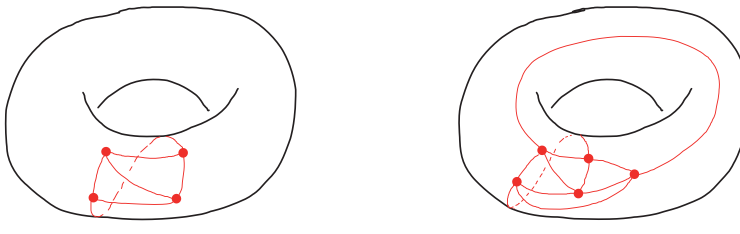


Fig. 4.4. Embeddings of K_4 and K_5 in the torus.

The following lemma says that every embedding of a connected graph in a surface of least possible Euler genus must be 2-cell.

FACT 4.10. *Let G be a connected graph that is embeddable in a surface $\boldsymbol{S}$, but not in a surface of Euler genus smaller than $\mathrm{eg}(\boldsymbol{S})$. Then every embedding of G in $\boldsymbol{S}$ is 2-cell.*

(For a proof, see Proposition 3.4.1 of Mohar and Thomassen [2001].)

FACT 4.11 (EULER'S FORMULA). *Let G be a graph embedded in a surface $\boldsymbol{S}$, and let $C(G)$ be the set of connected components of G. Then*

$$|V(G)| - |E(G)| + |F(G)| - |C(G)| + 1 \geq 2 - \mathrm{eg}(\boldsymbol{S}). \tag{4.1}$$

If G is 2-cell embedded in $\boldsymbol{S}$, then equality holds.

In the literature, Euler's formula is usually only stated for 2-cell embedded graphs. As 2-cell embedded graphs are connected, in this special case we have $|V(G)| - |E(G)| + |F(G)| = 2 - \mathrm{eg}(\boldsymbol{S})$. The integer $\chi(\boldsymbol{S}) := 2 - \mathrm{eg}(\boldsymbol{S})$ is known as the *Euler characteristic* of $\boldsymbol{S}$. We sketch a proof of the inequality (4.1) for arbitrary embedded graphs in Appendix F.

It is well-known that Euler's formula can be used to prove that sufficiently large surface graphs of minimum degree at least 3 have facial subgraphs with at most 6 vertices. We need a slight generalization of this fact to graphs of minimum degree 2. Recall that a *branch vertex* of a graph G is a vertex of degree at least 3 and that an *isolated path* path in G is a path $P \subseteq G$ that has no internal branch vertices.

LEMMA 4.12. *Let G be a graph of minimum degree at least 2 embedded in a surface $\boldsymbol{S}$. Then either G has at most $14(\mathrm{eg}(\boldsymbol{S}) - 2)$ branch vertices or there is a facial subgraph H of G that contains at most 6 branch vertices.*

PROOF. Suppose that every facial subgraph H of G contains more than 6 branch vertices. We shall prove that G has at most $14(\mathrm{eg}(\boldsymbol{S}) - 2)$ branch vertices.

Let $V_2 := \{v \in V(G) \mid \deg(v) = 2\}$ and $V_{\geq 3} := \{v \in V(G) \mid \deg(v) \geq 3\}$. Then $2|V_2| + 3|V_{\geq 3}| \leq \sum_{v \in V(G)} \deg(v) = 2 \cdot |E(G)|$ and thus

$$|V_{\geq 3}| \leq \frac{2}{3}\big(|E(G)| - |V_2|\big). \tag{4.2}$$

Let $P(G)$ be the set of all isolated paths in G of length at least 1 and with both endpoints in $V_{\geq 3}$. Observe that for every $p \in P(G)$,

(i) $|E(p)| - |V(p) \cap V_2| = 1$;
(ii) for all $\boldsymbol{f} \in F(G)$, if $E(p) \cap E(Bd(\boldsymbol{f})) \neq \emptyset$ then $p \subseteq Bd(\boldsymbol{f})$;
(iii) there are at most two faces $\boldsymbol{f} \in F(G)$ with $p \subseteq Bd(\boldsymbol{f})$.

Moreover, observe that (i) implies

$$|E(G)| - |V_2| = |P(G)|. \tag{4.3}$$

For every facial subgraph H of G, let $P(H) = \{p \in P(G) \mid p \subseteq H\}$. If H is a facial subgraph, then the minimum degree of H is at least 2. To see this, let $\boldsymbol{f}$ be the face such that $Bd(\boldsymbol{f}) = H$. Suppose that $v \in V(H)$ is a vertex with $\deg^H(v) \leq 1$. Then $\deg^G(v) \leq 1$ as well, because no edge of G has a nonempty intersection with the face $\boldsymbol{f}$ that "surrounds" v. This is a contradiction, because we assumed G to be of minimum degree 2.

It follows that every vertex in $V(H) \cap V_{\geq 3}$ is an endvertex of at least two paths in $P(H)$. This implies

$$|V(H) \cap V_{\geq 3}| \leq |P(H)|. \tag{4.4}$$

Recall that all facial subgraphs of G have more than 6 branch vertices. Then

$$\begin{aligned} 7|F(G)| &\leq \sum_{\boldsymbol{f} \in F(G)} \left|V(Bd(\boldsymbol{f})) \cap V_{\geq 3}\right| \\ &\leq \sum_{\boldsymbol{f} \in F(G)} \left|P(Bd(\boldsymbol{f}))\right| && \text{(by (4.4))} \\ &\leq 2|P(G)| && \text{(by (iii)).} \end{aligned}$$

Thus

$$|F(G)| \leq \frac{2}{7}|P(G)|. \tag{4.5}$$

By Euler's formula (4.1),

$$\begin{aligned} 2 - \mathrm{eg}(\boldsymbol{S}) + C(G) - 1 &\leq |V(G)| - |E(G)| + |F(G)| \\ &= |V_{\geq 3}| - |P(G)| + |F(G)| && \text{(by (4.3))} \\ &\leq \frac{2}{3}|P(G)| - |P(G)| + \frac{2}{7}|P(G)| && \text{(by (4.2), (4.3), and (4.5))} \\ &= -\frac{1}{21}|P(G)|. \end{aligned}$$

Then $|P(G)| \leq 21(\mathrm{eg}(\boldsymbol{S}) - C(G) - 1) \leq 21(\mathrm{eg}(\boldsymbol{S}) - 2)$, which implies

$$\begin{aligned} |V_{\geq 3}| &\leq \frac{2}{3}|P(G)| && \text{(by (4.2) and (4.3))} \\ &\leq 14(\mathrm{eg}(\boldsymbol{S}) - 2). \end{aligned}$$

□

We have already remarked that Fact 4.2 does not directly generalize from plane graphs to graphs embeddable in higher surfaces: even 3-connected graphs embedded in a torus may have facial subgraphs that are not cycles. However, we will see that there is a generalization of the fact. It is more complicated than the version for plane graphs, but still useful. We will look at both directions of the equivalence stated in Fact 4.2 separately, starting with the backward direction. Let G be a graph embedded in a surface $\boldsymbol{S}$, and let $C \subseteq G$ be a cycle. C is *contractible* if the simple closed curve $\boldsymbol{C}$ is contractible in $\boldsymbol{S}$ and *noncontractible* otherwise. It follows from Corollary 4.8 (2) that if C is noncontractible then every connected component of $G \setminus C$ is embeddable in a surface that has a smaller Euler genus than $\boldsymbol{S}$.

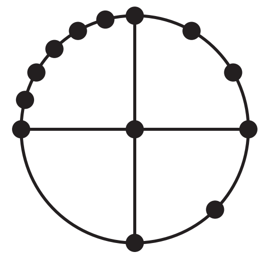


Fig. 4.5. A subdivided wheel with 4 spokes.

LEMMA 4.13. *Let G be a graph embedded in a surface $\boldsymbol{S}$ and $C \subseteq G$ a contractible, chordless, and nonseparating cycle of G. Then C is a facial cycle.*

PROOF. As $\boldsymbol{C}$ is contractible, $\boldsymbol{S} \setminus \boldsymbol{C}$ has two arcwise connected components $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ (one of which is an open disk). As C is nonseparating, either $V(G) \cap \boldsymbol{A}_1 = \emptyset$ or $V(G) \cap \boldsymbol{A}_2 = \emptyset$. Say, $V(G) \cap \boldsymbol{A}_2 = \emptyset$. Then $V(G) \subseteq \boldsymbol{cl}(\boldsymbol{A}_1)$. Let $\boldsymbol{e} \in E(G)$. If both endvertices of $\boldsymbol{e}$ are in $V(C)$, then $\boldsymbol{e} \in E(C)$ because C is chordless, and hence $\boldsymbol{e} \subseteq \boldsymbol{C} \subseteq \boldsymbol{cl}(\boldsymbol{A}_1)$. And if at least one endvertex of $\boldsymbol{e}$ is in $\boldsymbol{A}_1$, then clearly we also have $\boldsymbol{e} \subseteq \boldsymbol{cl}(\boldsymbol{A}_1)$. It follows that $\boldsymbol{G} \subseteq \boldsymbol{cl}(\boldsymbol{A}_1)$, and hence $\boldsymbol{A}_2$ is a face of G with boundary subgraph C. □

A subset $\boldsymbol{X} \subseteq \boldsymbol{S}$ is *G-normal* if $\boldsymbol{X} \cap \boldsymbol{G} \subseteq V(G)$, that is, $\boldsymbol{X}$ has an empty intersection with the interior of all edges of G. Suppose now that $\mathrm{eg}(\boldsymbol{S}) > 0$. Recall that by Corollary 4.8, there is a noncontractible simple closed curve in $\boldsymbol{S}$. The *representativity* $\rho(G)$ of G is the maximum $r \in \mathbb{N}$ such that every G-normal noncontractible simple closed curve $\boldsymbol{g}$ intersects $\boldsymbol{G}$ in at least r vertices. We extend the definition to plane graphs by letting $\rho(G) := \infty$ for every plane graph G. Intuitively, an embedded graph of high representativity locally looks like a plane graph. It is not hard to see that a connected embedded graph of representativity at least 1 is 2-cell embedded. Furthermore, if the representativity of an embedded graph G is at least 3 then for every vertex $v \in V(G)$ the subgraph $B(v) := \bigcup_{\boldsymbol{f}} Bd(\boldsymbol{f})$, where the union ranges over all faces $\boldsymbol{f} \in F(G)$ incident with v, is plane. (More precisely, there is an open disk in $\boldsymbol{S}$ that contains this subgraph.)

We say that G is *polyhedrally embedded* in $\boldsymbol{S}$ if G is 3-connected and embedded in $\boldsymbol{S}$ with representativity $\rho(G) \geq 3$. It can be shown [Robertson and Vitray 1990] that G is polyhedrally embedded if and only if for every vertex $v \in V(G)$ the graph $B(v)$ is plane and isomorphic to a subdivided wheel with at least 3 spokes (see Figure 4.5). Note that every 3-connected plane graph is polyhedrally embedded in the sphere.

Example 4.14. Figure 4.4(b) shows a 2-cell embedding of the 3-connected graph K_5 in the torus that is not polyhedral, because its representativity is 1.

The following fact due to Robertson and Vitray [1990] (to be precise, assertion (1) of the fact) is the promised generalization of the forward direction of Fact 4.2. For a proof, see Propositions 5.5.13 and 5.5.12 of Mohar and Thomassen [2001].

FACT 4.15. *Let G be polyhedrally embedded in a surface $\boldsymbol{S}$.*

(*1*) *All facial subgraphs of G are chordless and nonseparating cycles.*
(*2*) *For any two facial cycles C, C' of G it holds that* $|E(C) \cap E(C')| \leq 1$.
(*3*) *All edges of G are contained in exactly two facial cycles.*

We close this section with a few remarks on the algorithmic problem of embedding graphs in surfaces. It is NP-hard to decide whether a given graph can be embedded in a given surface (where the surface is given say, by specifying its genus and whether it is orientable or not) [Thomassen 1988]. This changes if we fix the surface.

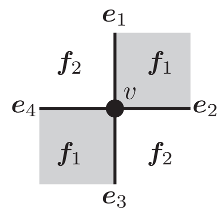


Fig. 5.1. Unexpected angles.

FACT 4.16 [MOHAR 1999]. *For every fixed surface* $\boldsymbol{S}$*, it is decidable in linear time whether a given graph is embeddable in* $\boldsymbol{S}$*.*

5. ANGLES

Let G be a graph embedded in a surface $\boldsymbol{S}$. An *angle* of a face $\boldsymbol{f} \in F(G)$ is a triple $(v_1, v_2, v_3) \in V(G)^3$ such that $v_1 \neq v_3$ and $v_1v_2, v_2v_3 \in E(Bd(\boldsymbol{f}))$. The set of all angles of a face $\boldsymbol{f}$ is denoted by $\angle(\boldsymbol{f})$. We let $\angle(G) := \bigcup_{\boldsymbol{f} \in F(G)} \angle(\boldsymbol{f})$. An angle $\bar{w} := (w_1, w_2, w_3) \in \angle(G)$ is aligned with an angle $\bar{v} := (v_1, v_2, v_3) \in \angle(G)$ (we write $\bar{v} \curvearrowright \bar{w}$) if $w_1 = v_2$ and $w_2 = v_3$ and there is a face $\boldsymbol{f} \in F(G)$ such that $(v_1, v_2, v_3), (w_1, w_2, w_3) \in \angle(\boldsymbol{f})$. Note that the angle relation is symmetric with respect to its first and third argument, that is, $(v_1, v_2, v_3) \in \angle(G) \iff (v_3, v_2, v_1) \in \angle(G)$. Moreover, $(v_1, v_2, v_3) \curvearrowright (w_1, w_2, w_3) \iff (w_3, w_2, w_1) \curvearrowright (v_3, v_2, v_1)$.

Angles are tailored towards 3-connected embedded graphs in which all facial subgraphs are cycles, because then they have the nice properties stated in the following lemmas. Recall that the facial cycles of 3-connected plane graphs and graphs polyhedrally embedded in a surface are cycles (by Facts 4.2 and 4.15). Thus the lemma applies to these.

LEMMA 5.1. *Let G be a 3-connected graph embedded in a surface such that all facial subgraphs of G are cycles.*

(1) *For every every angle $\bar{v} \in \angle(G)$ there is a unique face $\boldsymbol{f} \in F(G)$ such that $\bar{v} \in \angle(\boldsymbol{f})$.*
(2) *For every angle $\bar{v} \in V(G)$ there is a unique angle $\bar{w} \in \angle(G)$ such that $\bar{v} \curvearrowright \bar{w}$.*

It is interesting to note that even (1) does not necessarily hold in 3-connected embedded graphs in which not all facial subgraphs are cycles, because it may happen that there is a vertex v incident with four edges $\boldsymbol{e}_1 = vw_1, \dots, \boldsymbol{e}_4 = vw_4$ and two faces $\boldsymbol{f}_1, \boldsymbol{f}_2$ as in Figure 5.1. Then not only $(w_1, v, w_2), (w_3, v, w_4) \in \angle(\boldsymbol{f}_1), (w_2, v, w_3), (w_4, v, w_1) \in \angle(\boldsymbol{f}_2)$ are angles, as they should be, but actually all triples (w_i, v, w_j) for $i \neq j$ are angles in both $\angle(\boldsymbol{f}_1)$ and $\angle(\boldsymbol{f}_2)$.

PROOF OF LEMMA 5.1. To prove (1), suppose for contradiction that $\bar{v} = (v_1, v, v_2) \in \angle(\boldsymbol{f}_1) \cap \angle(\boldsymbol{f}_2)$ for distinct faces $\boldsymbol{f}_1, \boldsymbol{f}_2 \in F(G)$. Let $\boldsymbol{e}_1 := vv_1$ and $\boldsymbol{e}_2 := vv_2$ be the two edges of this angle. Both of these edges are incident with the faces $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ and thus with no other face. Let π_v be the cyclic permutation of the edges incident with v, and suppose that $\boldsymbol{e} := \pi_v(\boldsymbol{e}_1) \neq \boldsymbol{e}_2$. Then $\boldsymbol{e}$ would be incident with either $\boldsymbol{f}_1$ or $\boldsymbol{f}_2$, contradicting the assumption that $Bd(\boldsymbol{f}_1)$ and $Bd(\boldsymbol{f}_2)$ be cycles. Hence $\pi_v(\boldsymbol{e}_1) = \boldsymbol{e}_2$ and similarly $\pi_v(\boldsymbol{e}_2) = \boldsymbol{e}_1$. It follows that $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are the only edges incident with v. Hence $\{v_1, v_2\}$ separates v from $V(G) \setminus \{v, v_1, v_2\}$, which contradicts G being 3-connected.

To prove (2), let $\bar{v} = (v_1, v_2, v_3) \in \angle(\boldsymbol{f})$, and let $C := Bd(\boldsymbol{f})$. Then there is a unique vertex $v_4 \in N^C(v_3) \setminus \{v_2\}$, and thus (v_2, v_3, v_4) is the unique angle aligned with $\bar{v}$. $\square$

LEMMA 5.2. *Let G be a graph of minimum degree* 3 *embedded in a surface such that all facial subgraphs of G are cycles. Then for every* $v \in V(G)$ *the graph*

$$C_v := \big(N(v), \{ww' \mid (w, v, w' \in \angle(G)\}\big)$$

is a cycle.

PROOF. Let $v \in V(G)$, and let π_v by the cyclic permutation of the edges incident with v induced by the embedding. Suppose that the edges incident with v are $\boldsymbol{e}_1, \dots, \boldsymbol{e}_m, \boldsymbol{e}_{m+1} = \boldsymbol{e}_1$, where $\pi_v(\boldsymbol{e}_i) = \boldsymbol{e}_{i+1}$ for all $i \in [m]$. Then for all $i \in [m]$, there is a face $\boldsymbol{f}_i$ incident with $\boldsymbol{e}_i$ and $\boldsymbol{e}_{i+1}$, and every face $\boldsymbol{f}$ incident with v is among $\boldsymbol{f}_1, \dots, \boldsymbol{f}_m$. Furthermore, the faces $\boldsymbol{f}_i$ are mutually distinct, because if $\boldsymbol{f}_i = \boldsymbol{f}_j$ for some $i \neq j$, then $\boldsymbol{e}_i, \boldsymbol{e}_{i+1}, \boldsymbol{e}_j, \boldsymbol{e}_{j+1} \in E(Bd(\boldsymbol{f}_i))$. This contradicts $Bd(\boldsymbol{f}_i)$ being a cycle.

For every $i \in [m]$, let $\boldsymbol{e}_i = vw_i$. Then $(w_i, v, w_{i+1}) \in \angle(G)$, and all angles of the form (w, v, w') are among $(w_1, v, w_2), \dots, (w_m, v, w_{m+1})$. It follows that $E(C_v) = \{w_i w_{i+1} \mid i \in [m]\}$, and as $m = |N(v)| \geq 3$ and $w_{m+1} = w_1$, it follows that C_v is a cycle. $\square$

The next two lemmas are the main results of this section. They show that a linear order of an embedded graph or at least parts of the graph can be defined from the angles and the alignment relation between the angles. Readers mainly interested in our results for planar graphs may safely ignore the claims about polyhedrally embedded graphs in the following lemma.

LEMMA 5.3 (ANGLE LEMMA). *For all* IFP*-formulae* $\mathsf{ang}(y_1, y_2, y_3)$ *and* $\mathsf{align}(y_1, y_2, y_3, y_4)$ *there is an* IFP*-formula* $\mathsf{ord}(\bar{x}, z_1, z_2)$*, where* $|\bar{x}| = 3$*, such that the following holds. Suppose that G is a 3-connected plane graph or a graph polyhedrally embedded in some surface such that for all* $v_1, v_2, v_3, v_4 \in V(G)$:

(i) $G \models \mathsf{ang}[v_1, v_2, v_3] \iff (v_1, v_2, v_3) \in \angle(G)$;
(ii) $G \models \mathsf{align}[v_1, \dots, v_4] \iff (v_1, v_2, v_3) \curvearrowright (v_2, v_3, v_4)$.

Then there is a triple $\bar{u} \in V(G)^3$ *such that* $\mathsf{ord}[G, \bar{u}, z_1, z_2]$ *is a linear order of* $V(G)$*.*

PROOF. Let G be a 3-connected plane graph or a graph that is polyhedrally embedded in a surface. Suppose that G satisfies (i) and (ii). By Fact 4.2 or Fact 4.15, all facial subgraphs of G are cycles. Hence G satisfies the assumptions of Lemmas 5.1 and 5.2. By Lemma 5.1, for every angle $(v_1, v_2, v_3) \in \angle(G)$ there is a unique vertex $f(\bar{v}) \in V(G)$ such that $(v_1, v_2, v_3) \curvearrowright (v_2, v_3, f(\bar{v}))$. Let C_v be the cycle with vertex set $N(v)$ obtained from Lemma 5.2.

Let $\bar{u} = (u_1, u_2, u_3) \in \angle(G)$ be arbitrary. Depending on $\bar{u}$, we shall define an increasing sequence $V_1 \subset V_2 \subset \cdots \subset V_n = V(G)$ of nonempty subsets of $V(G)$ and an increasing sequence $\leq_1 \subseteq \leq_2 \subseteq \cdots \subseteq \leq_n \subseteq V(G)^2$ of binary relations such that for each $i \in [n]$ the following two conditions are satisfied.

(A) $\leq_i$ is a linear order of V_i.
(B) For every $v \in V_i$, there is an angle $\bar{w} \in V_i^3 \cap \angle(G)$ such that $v \in \widetilde{w}$.

We let $V_1 := \{u_1, u_2, u_3\}$ and define $\leq_1$ by $u_1 \leq_1 u_2 \leq_1 u_3$. For the inductive step, suppose that V_i and $\leq_i$ are defined and that $V_i \neq V(G)$. Let $v \in V_i$ be minimal with respect to the linear order $\leq_i$ such that $N(v) \not\subseteq V_i$. Such a v exists, because G is connected and thus the subset $V_i \subset V(G)$ has at least one outgoing edge.

Case 1: There are $w, w' \in N(v) \cap V_i$ such that $(w, v, w') \in \angle(G)$.
Let $(w, w') \in V_i^2$ be lexicographically minimal with respect to $\leq_i$ such that $(w, v, w') \in \angle(G)$. Let w'' be the first vertex in $N(v) \setminus V_i$ on the cyclic walk around C_v starting with w followed by w'. We let $V_{i+1} := V_i \cup \{w''\}$, and we let $\leq_{i+1}$ be the extension of $\leq_i$ with $v' \leq_{i+1} w''$ for all $v' \in V_i$. Then (A) is obviously

satisfied. Let w''' be the predecessor of w'' on the cyclic walk around C_v. Then $(w''', v, w'') \in V_{i+1}^3 \cap \angle(G)$. Hence (B) is also satisfied.

Case 2: There are no $w, w' \in N(v) \cap V_i$ such that $(w, v, w') \in \angle(G)$.
Then by (B) and by the symmetry of the angle relation, there are $w, v' \in V_i$ such that $(w, v', v) \in \angle(G)$. We choose such (w, v') lexicographically minimal with respect to $\leq_i$. Let $w' := f(w, v', v)$. Then $(v', v, w') \in \angle(G)$, and thus $w' \notin V_i$ by the assumption of Case 2. We let $V_{i+1} := V_i \cup \{w'\}$, and we let $\leq_{i+1}$ be the extension of $\leq_i$ with $w'' \leq_{i+1} w'$ for all $w'' \in V_i$. Then (A) and (B) are obviously satisfied.

Let $n \in \mathbb{N}$ such that $V_n = V(G)$. (Actually, $n = |V(G)| - 2$.) Then $\leq_n$ is a linear order of $V(G)$. It is easy to see that there is an IFP-formula $\mathsf{ord}(x_1, x_2, x_3, y_1, y_2)$, which uses the formulae $\mathsf{ang}(y_1, y_2, y_3)$ and $\mathsf{align}(y_1, y_2, y_3, y_4)$ as building blocks, and which does not depend on the specific graph G, such that

$$\leq_n = \mathsf{ord}[G, \bar{u}, y_1, y_2]. \quad \square$$

The readers mainly interested in our main results for planar graphs may safely skip the rest of this section.

For the results about graphs embeddable in higher surfaces, we also need a variant of the Angle Lemma that applies in a situation where we cannot define all angles of our graph by an IFP-formula. If G is a graph embedded in a surface and $\mathcal{F}$ a family of facial cycles of G, then we let $\angle(\mathcal{F})$ be the set of all angles of all faces $\boldsymbol{f}$ with $Bd(\boldsymbol{f}) \in \mathcal{F}$.

LEMMA 5.4 (PARTIAL ANGLE LEMMA). *Let* $\mathsf{ang}(y_1, y_2, y_3)$*,* $\mathsf{align}(y_1, y_2, y_3, y_4)$*, and* $\mathsf{rest\text{-}ord}(\bar{x}, y_1, y_2)$ *be* IFP*-formulae. Then there is an* IFP*-formula* $\mathsf{ord}(\bar{x}, z_1, z_2)$ *such that the following holds. Let G be a graph polyhedrally embedded in some surface,* $\bar{u} \in G^{\bar{x}}$ *and* $\mathcal{F} \subset F(G)$ *a family of faces of G such that:*

(i) *for all* $v_1, v_2, v_3 \in V(G)$*,*

$$G \models \mathsf{ang}[v_1, v_2, v_3] \iff (v_1, v_2, v_3) \in \angle(\mathcal{F});$$

(ii) *for all* $v_1, v_2, v_3, v_4 \in V(G)$*,*

$$G \models \mathsf{align}[v_1, \ldots, v_4] \iff (v_1, v_2, v_3), (v_2, v_3, v_4) \in \angle(\mathcal{F}) \textit{ and } (v_1, v_2, v_3) \curvearrowright (v_2, v_3, v_4);$$

(iii) $\mathsf{rest\text{-}ord}[G, \bar{u}, y_1, y_2]$ *is a linear order of the set*

$$W := \big\{w \in V(G) \bigm| \textit{there is an edge } \boldsymbol{e} \in E(G) \textit{ incident with } w \textit{ such that there is at most one } \boldsymbol{f} \in \mathcal{F} \textit{ incident with } \boldsymbol{e}\big\}.$$

Then $\mathsf{ord}[G, \bar{u}, z_1, z_2]$ *is a linear order of* $V(G)$*.*

Note that this lemma only applies if $\mathcal{F}$ is a strict subset of $F(G)$. If $\mathcal{F} = F(G)$, we can apply the Angle Lemma 5.3 to define a linear order.

PROOF. Let G be a graph polyhedrally embedded in some surface, $\bar{u} \in G^{\bar{x}}$, and $\mathcal{F} \subset F(G)$ such that (i)–(iii) are satisfied. Let W be defined as in (iii) and $\leq_W := \mathsf{rest\text{-}ord}[G, \bar{u}, y_1, y_2]$. By Fact 4.15, all facial subgraphs of G are cycles. Hence G satisfies the assumptions of Lemmas 5.1 and 5.2. For every angle $\bar{v} := (v_1, v_2, v_3) \in \angle(G)$ we let $f(\bar{v})$ be the unique vertex such that $\bar{v} \curvearrowright (v_2, v_3, f(\bar{v}))$. Note that if $\bar{v} \in \angle(\mathcal{F})$ then $(v_2, v_3, f(\bar{v})) \in \angle(\mathcal{F})$. For every $v \in V(G)$, let C_v the cycle with vertex set $N(v)$ defined in Lemma 5.2.

It follows from by Fact 4.15(3) that W is the union of the vertex sets of all facial cycles of G that are not contained in $\mathcal{F}$. Hence $W \neq \emptyset$, because $\mathcal{F} \subset F(G)$. For all $v \in V(G)$, we let

$$Q_v := \big(N(v), \{ww' \mid (w, v, w') \in \angle(\mathcal{F})\}\big).$$

Then $Q_v \subseteq C_v$. As C_v is a cycle, if $Q_v \neq C_v$ then Q_v is a disjoint union of paths, possibly of length 0, with both endvertices in W. Furthermore, we have

$$Q_v = C_v \iff v \in V(G)\setminus W.$$

We define an increasing sequence $V_1 \subset V_2 \subset V_3 \subset \cdots \subset V_n = V(G)$ of nonempty subsets of $V(G)$ and an increasing sequence $\leq_1 \subseteq \leq_2 \subseteq \cdots \subseteq \leq_n \subseteq V(G)^2$ of binary relations such that for each $i \in [n]$:

(A) $\leq_i$ is a linear order of V_i;
(B) for every $v \in V_i\setminus W$, there is an angle $\bar{w} \in V_i^3 \cap \angle(\mathcal{F})$ such that $v \in \widetilde{w}$.

We let $V_1 := W$ and $\leq_1 := \leq_W$. Note that $V_1 \neq \emptyset$, because $\mathcal{F} \neq F(G)$. For the inductive step, suppose that V_i and $\leq_i$ are defined and that $V_i \neq V(G)$. Let $v \in V_i$ be minimal with respect to the linear order $\leq_i$ such that $N(v) \not\subseteq V_i$.

Case 1: $v \notin W$ and there are $w, w' \in N(v) \cap V_i$ such that $(w, v, w') \in \angle(\mathcal{F})$.
Then $Q_v = C_v$ is a cycle, and we can argue exactly as in Case 1 of the proof of the Angle Lemma 5.3.
Case 2: $v \notin W$ and there are no $w, w' \in N(v) \cap V_i$ such that $(w, v, w') \in \angle(\mathcal{F})$.
In this case, we can argue as in Case 2 of the proof of the Angle Lemma 5.3.
Case 3: $v \in W$.
Then $Q_v \neq C_v$ is a disjoint union of paths with both endvertices in W. We can use the linear order of W to define a linear order on $N(v) = V(Q_v)$. We first order the paths lexicographically by their endvertices, and then we order each path of positive length linearly from the smaller to the larger endvertex. Let w be the first vertex in $N(v)\setminus V_i$ with respect to this linear order. We let $V_{i+1} := V_i \cup \{w\}$, and we let $\leq_{i+1}$ be the extension of $\leq_i$ with $w' \leq_{i+1} w$ for all $w' \in V_i$.

It is easy to see that the order $\leq_n$ is IFP-definable. □

6. PLANAR GRAPHS

In this section, we shall prove the following theorem.

THEOREM 6.1. *The class of 3-connected planar graphs admits* IFP*-definable orders.*

Combined with Lemma 2.13, the theorem has the following corollary.

COROLLARY 6.2. *The logic* IFP *captures* PTIME *on the class of 3-connected planar graphs.*

The following corollary follows from Theorem 6.1 by Lemma 3.15 and the 3CC-Lifting Lemma 3.17, for the latter recalling that the class planar graphs is closed under taking minors.

COROLLARY 6.3 (DEFINABLE STRUCTURE THEOREM FOR PLANAR GRAPHS). *The class of all planar graphs admits* IFP*-definable ordered treelike decompositions.*

Theorem 1.2 for planar graphs follows via definable canonization (Theorem 3.14).

COROLLARY 6.4. *The logic* IFP+C *captures* PTIME *on the class of planar graphs.*

With slightly more effort, we can also use Theorem 6.1 to prove that planarity is IFP-definable.

THEOREM 6.5. *The class of planar graphs is* IFP*-definable.*

PROOF. Let $\mathcal{P}$ be the class of all planar graphs and $\mathcal{P}'$ the class of all 3-connected planar graphs together with all complete graphs of order at most 3. By the 3CC-Decomposition Lemma, there is a d-scheme Λ such that for every $G \in \mathcal{P}$ the decomposition $\Lambda[G]$ is a treelike decomposition of G over $\mathcal{P}'$, and all separators $\sigma(t)$ of this decomposition are of size at most 2.

It follows from Theorem 6.1 that there is an IFP-formula $\mathsf{ord}(\bar{x}, y_1, y_2)$ that defines an order on every graph $G \in \mathcal{P}'$. Let $\mathcal{O} \supseteq \mathcal{P}'$ be the class of all graphs G such that ord defines an order on G. By Lemma 2.13, IFP captures PTIME on $\mathcal{O}$. Hence the polynomial-time decidable class $\mathcal{P}' \subseteq \mathcal{O}$ is IFP-definable.

Let $\mathcal{Q} \supseteq \mathcal{P}$ be the class of all graphs G such that $\Lambda[G]$ is a treelike decomposition of G of adhesion at most 2 over $\mathcal{P}'$. If follows from the Definability Lifting Lemma 3.8 that $\mathcal{Q}$ is IFP-definable.

I claim that $\mathcal{P} = \mathcal{Q}$. We already know that $\mathcal{P} \subseteq \mathcal{Q}$. To prove the converse inclusion, let $G \in \mathcal{Q}$. Let $\Delta = (D, \sigma, \alpha) := \Lambda[G]$. By induction on D, starting from the leaves, we shall prove that for all $t \in V(D)$ the graph

$$H_t := G[\gamma(t)] \cup K[\sigma(t)]$$

is planar. As by (TL.5), for every connected component A of G there is a $t \in V(D)$ such $G[\gamma(t)] = A$, this will show that all connected components of G are planar. Hence G is planar as well.

For the base step, note that for all leaves t we have $H_t = \tau(t) \in \mathcal{P}' \subseteq \mathcal{P}$. For the inductive step, let $t \in V(D)$, and let $u_1, \ldots, u_m \in N_+^D(t)$ by a system of representatives of the $\approx^\Delta$ classes in $N_+^D(t)$. That is, for each $u \in N_+^D(t)$ there is exactly one $i \in [m]$ such that $u \approx^\Delta u_i$. Let $H^0 := \tau(t)$ and $H^i := H^{i-1} \cup H_{u_i}$ for all $i \in [m]$. Note that for all $i \in [m]$,

$$H^{i-1} \cap H_{u_i} = K[\sigma(u_i)],$$

because $\sigma(u_i)$ is a clique in both $\tau(t)$ and H_{u_i} and $\alpha(u_i) \cap \alpha(u_j) = \emptyset$ for $j \neq i$ and $\alpha(u_i) \cap V(\tau(t)) = \alpha(u_i) \cap \beta(t) = \emptyset$. Recall that $|\sigma(u_i)| \leq 2$. As H_0 is planar and H_{u_i} is planar for all $i \in [m]$ by the induction hypothesis, a straightforward induction based on Fact 4.3 shows that H^i is planar for all $i \in [m]$. In particular, $H^m \supseteq H_t$ is planar. □

We shall give two different proofs of Theorem 6.1. The theorem follows from the proof of Theorem 7.1 given in the next section as a special case. In this section, we give a direct and arguably simpler proof that I find much nicer. The idea of this proof is also important in the proof of Theorem 1.3, or more specifically Lemma 8.9. The key to the proof is the following lemma.

LEMMA 6.6. *There are* IFP-*formulae* planar-angle(x_1, x_2, x_3) *and* planar-aligned(x_1, x_2, x_3, x_4) *such that for every 3-connected plane graph G and all $v_1, v_2, v_3, v_4 \in V(G)$ we have*

$$G \models \text{planar-angle}[v_1, v_2, v_3] \iff (v_1, v_2, v_3) \text{ is an angle of } G,$$
$$G \models \text{planar-aligned}[v_1, v_2, v_3, v_4] \iff (v_1, v_2, v_3) \curvearrowright (v_2, v_3, v_4).$$

The lemma will be proved in Section 6.1 next.

PROOF OF THEOREM 6.1. The theorem follows from Lemma 6.6 by means of the Angle Lemma 5.3. □

6.1. Proof of Lemma 6.6

The proof requires some preparation. In this subsection, we make the following assumption.

ASSUMPTION 6.7. *G is a 3-connected plane graph and $v_1, v_2, v_3 \in V(G)$ such that $v_1, v_3 \in N(v_2)$ and $v_1 \neq v_3$.*

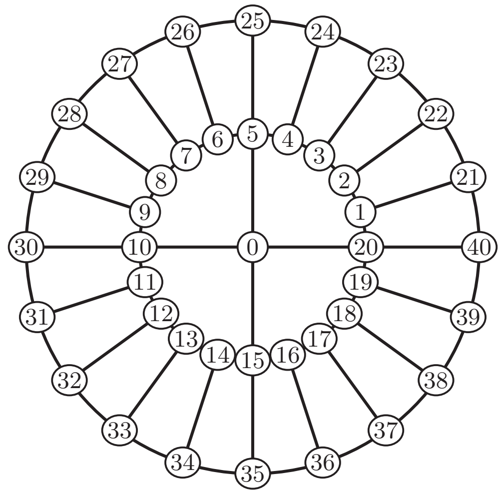


Fig. 6.1. The graph from Examples 6.8 and 6.12.

We let $\mathcal{C}(v_1, v_2, v_3)$ be the set of all cycles $C \subseteq G$ with $v_1v_2, v_2v_3 \in E(C)$. Observe that $\mathcal{C}(v_1, v_2, v_3)$ contains a facial cycle of G if and only if (v_1, v_2, v_3) is an angle of G. For each cycle $C \in \mathcal{C}(v_1, v_2, v_3)$, we define the *inside* of C to be the set

$$I(C) := \{v_2\} \cup \left\{ v \in V(G) \;\middle|\; \begin{array}{l}\text{each path from } v \text{ to } v_2 \text{ has a nonempty}\\ \text{intersection with } C \setminus \{v_2\}\end{array} \right\}.$$

Note that $V(C) \subseteq I(C)$. Inclusion of the sets $I(C)$ defines a quasiorder on the set $\mathcal{C}(v_1, v_2, v_3)$. We shall see that if $\mathcal{C}(v_1, v_2, v_3)$ contains a facial cycle then this facial cycle is the unique minimal element of this quasiorder.

Example 6.8. Let G be the 3-connected planar graph shown in Figure 6.1. Consider the following cycles from $\mathcal{C}(5, 0, 10)$ (we specify the cycles as sequences of vertices).

$$\begin{aligned}
C_1 &:= 5, 0, 10, 9, 8, 7, 6, 5,\\
C_2 &:= 5, 0, 10, 30, 29, 28, 27, 7, 6, 5,\\
C_3 &:= 5, 0, 10, 9, 8, 28, 27, 26, 25, 5,\\
C_4 &:= 5, 0, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 1, 2, 3, 4, 5
\end{aligned}$$

We have

$$\begin{aligned}
I(C_1) &= \{0, 10, 9, 8, 7, 6, 5\},\\
I(C_2) &= \{0, 10, 9, 8, 7, 6, 5, 30, 29, 28, 27\},\\
I(C_3) &= \{0, 10, 9, 8, 7, 6, 5, 28, 27, 26, 25\},\\
I(C_4) &= V(G).
\end{aligned}$$

Thus $I(C_1) \subseteq I(C_2) \subseteq I(C_4)$ and $I(C_1) \subseteq I(C_3) \subseteq I(C_4)$. The sets $I(C_2)$ and $I(C_3)$ are incomparable.

LEMMA 6.9. *For every facial cycle $C \in \mathcal{C}(v_1, v_2, v_3)$ it holds that $I(C) = V(C)$.*

PROOF. Let $C \in \mathcal{C}(v_1, v_2, v_3)$ be a facial cycle. Suppose for contradiction that there is a vertex $w \in I(C)\setminus V(C)$. Then $V(C) \setminus \{v_2\}$ separates w from v_2. As G is 3-connected, there is a path P from w to v_2 in $G \setminus \{v_1, v_3\}$. Let P be such a path, and let u be the

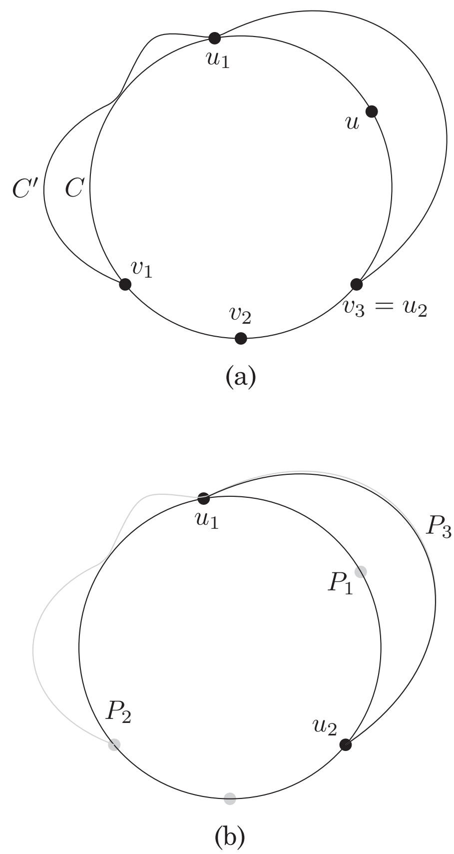


Fig. 6.2. Proof of Lemma 6.11.

last vertex on P in $V(C) \setminus \{v_2\}$, and let w' be the successor of u on P. Then the segment $P' := w'Pv_2$ has an empty intersection with $C \setminus \{v_2\}$. By Fact 4.2, C is chordless, and hence $w' \neq v_2$. Furthermore, C is nonseparating, and hence there is a path from w to w' in $G \setminus C$. Let Q be such a path. Then $Q \cup P'$ contains a path from w to v_2 that has an empty intersection with $V(C) \setminus \{v_2\}$, which is impossible because $w \in I(C)$. □

LEMMA 6.10. *Let $C, C' \in \mathcal{C}(v_1, v_2, v_3)$ such that $V(C) \subseteq I(C')$. Then $I(C) \subseteq I(C')$.*

PROOF. Let P be a path from a vertex $v \in I(C)$ to v_2. Then P has a nonempty intersection with $V(C) \setminus \{v_2\}$. Let $w \in \big(V(C) \setminus \{v_2\}\big) \cap V(P)$. Then $w \in I(C')$, and hence the path $wPv_2 \subseteq P$ has a nonempty intersection with $V(C') \setminus \{v_2\}$. □

LEMMA 6.11. *Let $C \in \mathcal{C}(v_1, v_2, v_3)$ be a facial cycle. Then for all $C' \in \mathcal{C}(v_1, v_2, v_3) \setminus \{C\}$ it holds that*

$$I(C) \subseteq I(C').$$

PROOF. Let $C' \in \mathcal{C}(v_1, v_2, v_3) \setminus \{C\}$. Let $P := C \setminus \{v_2\}$ and $P' := C' \setminus \{v_2\}$. By Lemma 6.9, it suffices to prove that $V(C) \subseteq I(C')$. Suppose that $u \in V(C) \setminus I(C')$. Then $u \notin V(C')$ and thus $u \notin V(P')$. Let u_1 be the last vertex of P' contained in v_1Pu (possibly $u_1 = v_1$), and let u_2 be the first vertex of $u_1P'v_3$ contained in uPv_3 (possibly $u_2 = v_3$); see Figure 6.2. Let $P_1 := u_1Pu_2$, P_2 the segment of C from u_1 to u_2 that does not contain u, and

$P_3 := u_1 P' u_2$. Note that $\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3$ are three simple curves in the sphere $\boldsymbol{S}^2$ with endpoints u_1, u_2 and mutually disjoint interiors and that $\boldsymbol{P}_1 \cup \boldsymbol{P}_2 = \boldsymbol{C}$ is the boundary of a face. To prove that $u \in I(C')$, we must prove that every path from u to v_2 has a nonempty intersection with $V(P')$. Let Q be a path from u to v_2. Then $\boldsymbol{Q}$ is a simple curve from an interior point on $\boldsymbol{P}_1$ to an interior point on $\boldsymbol{P}_2$. It has an empty intersection with the face bounded by $\boldsymbol{C} = \boldsymbol{P}_1 \cup \boldsymbol{P}_2$. Hence by Fact 4.1, it has a nonempty intersection with $\boldsymbol{P}_3 \subseteq \boldsymbol{C}'$. This shows that $u \in I(C')$ and hence that $I(C) \subseteq I(C')$. $\square$

We define a sequence $(W_n)_{n \in \mathbb{N} \cup \{\infty\}}$ of subsets of $V(G)$ by

$$W_0 := \emptyset,$$

$$W_{n+1} := W_n \cup \big\{ v \in N(W_n \cup \{v_2\}) \;\big|\; \exists C \in \mathcal{C}(v_1, v_2, v_3):\ V(C) \cap (W_n \cup \{v\}) = \emptyset \big\} \qquad (\text{for } n \in \mathbb{N}).$$

$$W_\infty := \bigcup_{n \in \mathbb{N}} W_n.$$

Observe that for every $n \in \mathbb{N} \cup \{\infty\}$ and every $w \in W_n$ there is a path from v_2 to w in $G[W_n \cup \{v_2\}]$.

Example 6.12. Let G be the 3-connected planar graph shown in Figure 6.1. We first compute the sets W_i for the angle $(v_1, v_2, v_3) := (5, 0, 10)$.

$$\begin{aligned}
W_1 &= \{15, 20\},\\
W_2 &= \{15, 20, 14, 16, 35, 19, 1, 40\},\\
W_3 &= \{15, 20, 14, 16, 35, 19, 1, 40, 13, 34, 17, 36, 18, 39, 2, 21\},\\
&\ \vdots\\
W_6 &= \{11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 1, 2, 3, 4,\\
&\qquad 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 21, 22, 23, 24\}\\
W_7 &= W_6 \cup \{25, 30\}\\
&\ \vdots\\
W_9 &= V(G) \setminus \{0, 5, 6, 7, 8, 9, 10\}\\
W_i &= W_9 \qquad \text{for all } i \geq 9.
\end{aligned}$$

Thus $W_\infty \backslash V(G)$ is the vertex set of the facial cycle determined by the angle (5, 0, 10).

Let us also compute the sets W_i for the triple $(v_1, v_2, v_3) := (5, 0, 15)$, which is not an angle. We have

$$\begin{aligned}
W_1 &= \{10, 20\}\\
W_2 &= \{10, 20, 9, 11, 30, 1, 19, 40\}\\
W_i &= W_2 \qquad \text{for all } i \geq 2.
\end{aligned}$$

LEMMA 6.13. *Let $C \in \mathcal{C}(v_1, v_2, v_3)$ be a facial cycle. Then*

$$V(C) = V(G) \backslash W_\infty.$$

PROOF. To see that $V(C) \subseteq V(G) \backslash W_\infty$, we prove that $V(C) \cap W_n = \emptyset$ by induction on n. This is trivial for $n = 0$. For the induction step, suppose that $V(C) \cap W_n = \emptyset$ and let $v \in W_{n+1} \backslash W_n$. Then there is a cycle $C' \in \mathcal{C}(v_1, v_2, v_3)$ such that $V(C') \cap (W_n \cup \{v\}) = \emptyset$. Furthermore, there is a path from v_2 to v in $G[W_i \cup \{v_2, v\}]$. Hence $v \notin I(C')$. By

Lemma 6.11, we have $V(C) \subseteq I(C) \subseteq I(C')$ and thus $v \notin V(C)$. This shows that $W_{i+1} \cap V(C) = W_i \cap V(C) = \emptyset$.

To see that $V(C) \supseteq V(G) \setminus W_\infty$, let $w \in V(G) \setminus W_\infty$. Suppose for contradiction that $w \notin V(C)$. As G is 3-connected, it holds that $\deg(v_2) \geq 3$. Hence there is a $w' \in N(v_2) \setminus \{v_1, v_3\}$. As C is nonseparating (by Fact 4.2), there is a path P from w' to w in $G \setminus C$. A straightforward induction on the length of P shows that $V(P) \subseteq W_\infty$. Thus $w \in W_\infty$, which is a contradiction. □

Proof of Lemma 6.6. It is not hard to formalize the inductive definition of the set W_∞ in the logic IFP. For once, let us carry out the details; the reader may safely skip them. Let

$$\begin{aligned}&\mathsf{possible}(X, x_1, x_2, x_3, x) := \neg\big(X(x_2) \vee x = x_2\big) \wedge\\ &\qquad \mathrm{ifp}\Big(Y(y) \leftarrow \neg X(y) \wedge y \neq x \wedge y \neq x_2 \wedge \big(y = x_1 \vee \exists y'\big(Y(y') \wedge E(y', y)\big)\big)\Big)(x_3).\end{aligned}$$

Then for every set $W \subseteq V(G)$ and every vertex $v \in V(G)$ we have

$$\begin{aligned}&G \models \mathsf{possible}[W, v_1, v_2, v_3, v]\\ \iff\ & v_2 \notin W \cup \{v\} \text{ and there is a path from } v_1 \text{ to } v_3 \text{ in } G \setminus \big(W \cup \{v_2, v\}\big)\\ \iff\ & \text{there is a } C \in \mathcal{C}(v_1, v_2, v_3) \text{ with } V(C) \cap (W \cup \{v\}) = \emptyset.\end{aligned}$$

Let

$$\mathsf{next}(X, x_1, x_2, x_3, x) := \mathsf{possible}(X, x_1, x_2, x_3, x) \wedge \exists x'((X(x') \vee x' = x_2) \wedge E(x', x)).$$

Then $\mathsf{next}[G, W_i, v_1, v_2, v_3, x] = W_{i+1}$ for all $i \in \mathbb{N}^+$. Now we let

$$\begin{aligned}\mathsf{w\text{-}infty}(x_1, x_2, x_3, x) &:= \mathrm{ifp}\big(X(x) \leftarrow \mathsf{next}(X, x_1, x_2, x_3, x)\big)(x),\\ \mathsf{candidate}(x_1, x_2, x_3, x) &:= \neg\mathsf{w\text{-}infty}(x_1, x_2, x_3, x).\end{aligned}$$

Then $\mathsf{w\text{-}infty}[G, v_1, v_2, v_3, x] = W_\infty$. By Lemma 6.11, if $\mathcal{C}(v_1, v_2, v_3)$ contains a facial cycle C then

$$V(C) = \mathsf{candidate}[G, v_1, v_2, v_3, x].$$

Next, we want to state that a set of vertices is the vertex set of a chordless and nonseparating cycle. We define a few auxiliary formulae.

—$\mathsf{deg}_2(X) := \forall x(X(x) \to \exists y_1 \exists y_2(y_1 \neq y_2 \wedge X(y_1) \wedge X(y_2) \wedge E(x, y_1) \wedge E(x, y_2) \wedge \forall y'((X(y') \wedge E(x, y')) \to (y' = y_1 \vee y' = y_2))))$ states that all vertices in X have exactly two neighbors in X.
—$\mathsf{conn}(X) := \forall x \forall x'((X(x) \wedge X(x')) \to \mathrm{ifp}(Z(z) \leftarrow z = x \vee \exists z'(Z(z') \wedge E(z', z) \wedge X(z)))(x'))$ states that X is connected.
—Hence $\mathsf{cl\text{-}cycle}(X) := \mathsf{deg}_2(X) \wedge \mathsf{conn}(X)$ states that X is the vertex set of a chordless cycle.
—$\mathsf{non\text{-}sep}(X) := \mathsf{conn}(\neg X)$, the formula obtained from $\mathsf{conn}(X)$ by replacing X by $\neg X$ everywhere, states that the complement of X is connected and hence that X is nonseparating.
—Hence $\mathsf{cl\text{-}ns\text{-}cycle}(X) := \mathsf{cl\text{-}cycle}(X) \wedge \mathsf{non\text{-}sep}(X)$ states that X is the vertex set of a chordless and nonseparating cycle.

By Fact 4.2, for every set $W \subseteq V(G)$ we have

$$G \models \mathsf{cl\text{-}ns\text{-}cycle}[W] \iff G[W] \text{ is a facial cycle.}$$

Now let planar-angle(x_1, x_2, x_3) be the formula obtained from cl-ns-cycle(X) by replacing every atom of the form $X(x'')$, for any variable x'', by the formula candidate(x_1, x_2, x_3, x''). (Note that none of the variables x_1, x_2, x_3 appears in the formula cl-ns-cycle(X). Hence there are no name clashes.) Then $G \models$ planar-angle$[v_1, v_2, v_3]$ if any only if $\mathcal{C}(v_1, v_2, v_3)$ contains a facial cycle. Recall that $\mathcal{C}(v_1, v_2, v_3)$ contains a facial cycle if and only if (v_1, v_2, v_3) is an angle.

Two angles (v_1, v_2, v_3) and (v_2, v_3, v_4) are aligned if and only if they determine the same facial cycle. We let

$$\begin{aligned}\mathsf{planar\text{-}aligned}(x_1, x_2, x_3, x_4) := {}& \mathsf{planar\text{-}angle}(x_1, x_2, x_3) \wedge \mathsf{planar\text{-}angle}(x_2, x_3, x_4)\\ &\wedge \forall x(\mathsf{candidate}(x_1, x_2, x_3, x) \leftrightarrow \mathsf{candidate}(x_2, x_3, x_4, x)). \quad \square\end{aligned}$$

7. GRAPHS EMBEDDABLE IN A SURFACE

In this section, we shall prove Theorem 1.2. It is an easy consequence of the following structure theorem.

THEOREM 7.1 (DEFINABLE STRUCTURE THEOREM FOR EMBEDDABLE GRAPHS). *For every surface **S**, the class of all graphs embeddable in **S** admits* IFP*-definable ordered treelike decompositions.*

Theorem 1.2 follows by means of definable canonization (the Canonization Theorem 3.14 and Lemma 2.16).

Recall that we proved the class of planar graphs to be IFP-definable. For all surfaces ***S*** except the sphere, it is an open question whether the class of all graphs embeddable in ***S*** is IFP-definable. As a consequence of the Definable Structure Theorem 7.1, at least we can prove IFP+C-definability.

COROLLARY 7.2. *For every surface **S**, the class of all graphs embeddable in **S** is* IFP+C*-definable.*

PROOF. Let $\mathcal{E}$ be the class of all graphs embeddable in ***S***. Let Λ be a d-scheme such that $\mathcal{E} \subseteq \mathcal{OT}_\Lambda$. Then $\mathcal{OT}_\Lambda$ admits IFP-definable ordered treelike decompositions, and thus IFP+C captures PTIME on $\mathcal{OT}_\Lambda$. As $\mathcal{E}$ is decidable in polynomial time by Fact 4.16 and $\mathcal{OT}_\Lambda$ is IFP+C-definable by Lemma 3.13, this implies that $\mathcal{E}$ is definable in IFP+C. $\square$

Our proof of Theorem 7.1 is by induction on the Euler genus. We could take Theorem 6.1 for planar graphs as the induction basis, but actually we do not have to, because for genus 0 our construction will never use the induction hypothesis. We will see that this gives an alternative proof of Theorem 6.1 (see Remark 7.17). For every $g \in \mathbb{N}$, we let $\mathcal{E}_g$ be the class of all disjoint unions of graphs of Euler genus at most g. We let $\mathcal{E}_{-1} := \emptyset$. The following lemma takes care of the inductive step of our proof.

LEMMA 7.3. *Let $g \in \mathbb{N}$, and suppose that there is an od-scheme Λ^{g-1} that defines ordered treelike decompositions on $\mathcal{E}_{g-1}$. Then for every surface **S** of Euler genus g there is an od-scheme Λ such that for all 3-connected graphs G embeddable in **S**, the o-decomposition $\Lambda[G]$ is an ordered treelike decomposition of G.*

PROOF OF THEOREM 7.1. By induction on $g \geq -1$, we prove that $\mathcal{E}_g$ admits IFP-definable ordered treelike decompositions. The induction basis $g = -1$ is trivial. For the inductive step, let $g \geq 0$. By Lemma 7.3, for every surface ***S*** of Euler genus g the class of all 3-connected graphs embeddable in ***S*** admits IFP-definable ordered treelike decompositions. Since up to homeomorphism, there are only two surfaces of Euler genus g, it follows from Lemma 3.16(1) that the class of all 3-connected graphs of Euler genus

g admits IFP-definable ordered treelike decompositions. Now it follows from the 3CC-Lifting Lemma 3.17 that $\mathcal{E}_g$ admits IFP-definable ordered treelike decompositions. $\square$

The rest of section is devoted to a proof of Lemma 7.3. Until the end of the section, we make the following assumption.

ASSUMPTION 7.4. *$\boldsymbol{S}$ is a surface of Euler genus $g \in \mathbb{N}$. Furthermore, Λ^{g-1} is an od-scheme such that $\mathcal{D}_{g-1} := \mathcal{OT}_{\Lambda^{g-1}} \supseteq \mathcal{E}_{g-1}$.*

7.1. Defining the Faces

In addition to Assumption 7.4, throughout Section 7.1 we assume the following.

ASSUMPTION 7.5. *G is a 3-connected graph in $\mathcal{E}_g \setminus \mathcal{D}_{g-1}$ that is polyhedrally embedded in $\boldsymbol{S}$.*

It is our goal to define an od-scheme Λ^g that defines an ordered treelike decomposition of G. The od-scheme Λ^g may depend on g and Λ^{g-1}, but of course not on the specific graph G. The proof strategy is as follows. We will iteratively define sets F_i of faces of G. Either we succeed to define all (or almost all) faces of G, then we can define an ordering on G using the Partial Angle Lemma 5.4. Or on the way we find a cycle C such that $G\backslash C \in \mathcal{D}_{g-1}$. We call such a cycle a *reducing cycle*. Then we delete the cycle C and apply the od-scheme Λ^{g-1} to the resulting graph $G\backslash C$. We apply the Ordered Extension Lemma 3.18 and combine a definable order on C with the ordered treelike decomposition of $G\backslash C$ to obtain an ordered treelike decomposition of G.

By induction on $i \in \mathbb{N}$, we define sets $F_i \subseteq F(G)$. In addition, we define sets $\boldsymbol{IS}_i, \boldsymbol{BS}_i, \boldsymbol{XS}_i \subseteq \boldsymbol{S}$, IV_i, BV_i, $XV_i \subseteq V(G)$, IE_i, BE_i, $XE_i \subseteq E(G)$ and a graphs H_i. Once we have defined F_i, to define $\boldsymbol{IS}_i, \dots, XE_i, H_i$, we look at the closed subset

$$\bigcup_{\boldsymbol{f} \in F_i} \boldsymbol{cl}(\boldsymbol{f})$$

of $\boldsymbol{S}$. We let $\boldsymbol{IS}_i, \boldsymbol{BS}_i, \boldsymbol{XS}_i$ be the interior, boundary, and "exterior" (that is, complement in $\boldsymbol{S}$) of this set. We let IV_i, BV_i, XV_i and IE_i, BE_i, XE_i be the vertices and edges of G in the respective sets $\boldsymbol{IS}_i, \boldsymbol{BS}_i, \boldsymbol{XS}_i$, and we let H_i be the subgraph in $\boldsymbol{BS}_i \cup \boldsymbol{XS}_i$. The idea is that at stage i of our induction we know the faces of G in $\boldsymbol{IS}_i$, and thus it remains to determine the faces of the graph H_i. The formal definitions are as follows.

(A) We let $F_0 := \emptyset$.

In the following, suppose that F_i is defined for some $i \geq 0$.

(B) We let $\boldsymbol{IS}_i := \boldsymbol{int}\big(\bigcup_{\boldsymbol{f} \in F_i} \boldsymbol{cl}(\boldsymbol{f})\big)$, $\boldsymbol{BS}_i := \boldsymbol{bd}(\boldsymbol{IS}_i)$, and $\boldsymbol{XS}_i := \boldsymbol{S} \setminus (\boldsymbol{IS}_i \cup \boldsymbol{BS}_i)$.
(C) We let $IV_i := V(G) \cap \boldsymbol{IS}_i$, $BV_i := V(G) \cap \boldsymbol{BS}_i$, and $XV_i := V(G) \cap \boldsymbol{XS}_i$.
(D) We let $BE_i := \{\boldsymbol{e} \in E(G) \mid \boldsymbol{e} \subseteq \boldsymbol{BS}_i\}$, $IE_i := \{\boldsymbol{e} \in E(G) \mid \boldsymbol{e} \subseteq \boldsymbol{IS}_i \cup \boldsymbol{BS}_i\} \setminus BE_i$, and $XE_i := E(G) \setminus (IE_i \cup BE_i)$.

Note that IE_i is the set of all edges of G only incident to faces in F_i, BE_i is the set of all edges of G both incident to a face in F_i and a face in $F(G)\backslash F_i$, and XE_i is the set of all edges not incident to any faces in F_i.

(E) We let $H_i := \big(V(G)\backslash IV_i, E(G) \setminus IE_i\big)$.

Observe that $H_0 = G$. As a subgraph of G, the graph H_i is embedded in $\boldsymbol{S}$. The faces of H_i are the faces of G not in F_i and the arcwise connected components of $\boldsymbol{IS}_i$.

A cycle C of H_i is *suitable (in H_i)* if C is chordless and nonseparating in G and $V(C)$ contains at most 36 branch vertices of H_i. (It will become apparent in the proof of Lemma 7.7 where the number 36 comes from.) Recall that a cycle C of G is reducing if $G\setminus C \in \mathcal{D}_{g-1}$; otherwise it is *nonreducing*. Note that every noncontractible cycle is reducing, because $\mathcal{E}_{g-1} \subseteq \mathcal{D}_{g-1}$. We can now complete the inductive definition of the sets F_i.

(F) We let

$$F_{i+1} := F_i \cup \{ \boldsymbol{f} \in F(G) \mid Bd(\boldsymbol{f}) \subseteq H_i \text{ is suitable and nonreducing} \}.$$

Note that a face whose boundary is a suitable facial cycle does not necessarily end up in F_{i+1}, because the boundary may still be a reducing cycle.

As G is finite and $F_0 \subseteq F_1 \subseteq F_2 \subseteq \cdots$, there is an $n_\infty \in \mathbb{N}$ such that $F_i = F_{n_\infty}$ for all $i \geq n_\infty$. We let $F_\infty := F_{n_\infty}$. Furthermore, we let $\boldsymbol{IS}_\infty := \boldsymbol{IS}_{n_\infty}$ and define $\boldsymbol{BS}_\infty$, $\boldsymbol{XS}_\infty$, $IV_\infty, BV_\infty, XV_\infty, IE_\infty, BE_\infty, XE_\infty$, and H_∞ similarly.

LEMMA 7.6. *For all $i \in \mathbb{N}$ the graph H_i has minimum degree at least* 2.

PROOF. Let $v \in V(H_i) = BV_i \cup XV_i$. Then there is at least one face $\boldsymbol{f} \in F(G)\setminus F_i$ incident with v. The two edges on the boundary cycle of $\boldsymbol{f}$ that are incident with v are both contained in $BE_i \cup XE_i = E(H_i)$. Hence $\deg^{H_i}(v) \geq 2$. □

LEMMA 7.7. *Let $i \in \mathbb{N}$. If H_i has more than* $\max\{2, 84g\}$ *branch vertices, then H_i contains a suitable facial cycle.*

PROOF. Suppose that H_i has more than $\max\{2, 84g\}$ branch vertices. As $84g > 14g-2$, by Lemma 4.12 the graph H_i has a facial cycle with at most 6 branch vertices. However, the face of H_i bounded by this cycle is not necessarily a face of G, but may also be an arcwise connected component of $\boldsymbol{IS}_i$. Let $\mathfrak{A}$ be the set of all arcwise connected components of $\boldsymbol{IS}_i$. Let us call $\boldsymbol{A} \in \mathfrak{A}$ *small* if its boundary contains at most 6 branch vertices of H_i, and let $\mathfrak{S} \subseteq \mathfrak{A}$ be the set of all small arcwise connected components of $\boldsymbol{IS}_i$. We define a new graph J as follows.

—$V(J)$ consists of all vertices in XV_i, all vertices in BV_i that are contained in the boundary of an $\boldsymbol{A} \in \mathfrak{A}\setminus\mathfrak{S}$, all vertices in BV_i that are branch vertices of H_i and contained in the boundary of an $\boldsymbol{A} \in \mathfrak{S}$, and a new vertex $v_{\boldsymbol{A}}$ for every $\boldsymbol{A} \in \mathfrak{S}$.

—The edge set $E(J)$ consists of all edges in XE_i, all edges in BE_i that are contained in the boundary of an $\boldsymbol{A} \in \mathfrak{A}\setminus\mathfrak{S}$, and for every $\boldsymbol{A} \in \mathfrak{S}$ new edges between $v_{\boldsymbol{A}}$ and all branch vertices of H_i contained in the boundary of $\boldsymbol{A}$.

We can embed J in $\boldsymbol{S}$ by placing the new vertices $v_{\boldsymbol{A}}$ and the interior of the edges incident with $v_{\boldsymbol{A}}$ in $\boldsymbol{A}$. This embedding is not unique, possibly not even up to homeomorphism. Nevertheless, we fix one such embedding and henceforth view J as a graph embedded in $\boldsymbol{S}$.

CLAIM 1. *For every $\boldsymbol{A} \in \mathfrak{S}$, the vertex $v_{\boldsymbol{A}}$ is a branch vertex of J.*

PROOF. Let $\boldsymbol{A} \in \mathfrak{S}$. Suppose for contradiction that $\boldsymbol{bd}(\boldsymbol{A})$ contains at most two branch vertices of H_i. As H_i has at least 3 branch vertices (here we use the "2" in $\max\{2, 84g\}$), there is some vertex $w \in V(H_i)$ such that $w \notin \boldsymbol{bd}(\boldsymbol{A})$. As $V(H_i) \cap \boldsymbol{IS}_i = \emptyset$ and $\boldsymbol{A} \subseteq \boldsymbol{IS}_i$, we have $w \notin \boldsymbol{cl}(\boldsymbol{A})$. Since every face of G is incident to at least 3 vertices and $\boldsymbol{A}$ contains at least one face of G, there must be a vertex $w' \in V(G) \cap \boldsymbol{cl}(\boldsymbol{A})$ that is not a branch vertex of H_i. As G is 3-connected, there are three internally disjoint paths from w to w' in G. All these paths must enter $\boldsymbol{cl}(\boldsymbol{A})$ in a branch vertex of H_i on $\boldsymbol{bd}(\boldsymbol{A})$. Hence $\boldsymbol{bd}(\boldsymbol{A})$ contains at least three branch vertices of H_i, which is a contradiction. □

CLAIM 2. *J has more than $14g$ branch vertices.*

Proof. Note first that all branch vertices of H_i that are not in the boundary of an $\boldsymbol{A} \in \mathfrak{S}$ are also branch vertices of J. Let $\boldsymbol{A} \in \mathfrak{S}$. Then $v_{\boldsymbol{A}}$ is a branch vertex of J by Claim 1, and $\boldsymbol{bd}(\boldsymbol{A})$ contains at most 6 branch vertices of H_i. Thus the number of branch vertices of H_i is at most 6 times the number of branch vertices of J, and as H_i has more than $84g$ branch vertices, the claim follows. $\square$

Claim 3. *For every face $\boldsymbol{f} \in F(J)$, every arcwise connected component of $\boldsymbol{f} \cap \boldsymbol{XS}_i$ is in $F(G) \cap F(H_i)$.*

Proof. Let $\boldsymbol{f} \in F(J)$, and let $\boldsymbol{f}'$ be an arcwise connected component of $\boldsymbol{f} \cap \boldsymbol{XS}_i$. Then $V(G) \cap \boldsymbol{f}' = \emptyset$, because $V(G) \cap \boldsymbol{XS}_i = XV_i = V(J) \cap \boldsymbol{XS}_i$. For every edge $\boldsymbol{e} \in E(G)$ we have $\boldsymbol{e} \cap \boldsymbol{f}' = \emptyset$, because if $\boldsymbol{e} \cap \boldsymbol{XS}_i \neq \emptyset$ then $\boldsymbol{e} \in XE_i \subseteq E(J)$. Furthermore, $\boldsymbol{bd}(\boldsymbol{f}') \subseteq \boldsymbol{bd}(\boldsymbol{XS}_i) \cup \big(\boldsymbol{bd}(\boldsymbol{f}) \cap \boldsymbol{XS}_i\big) \subseteq \boldsymbol{H}_i$. Hence $\boldsymbol{f}'$ is an arcwise connected component of $\boldsymbol{S} \setminus \boldsymbol{H}_i$ with $\boldsymbol{f}' \cap \boldsymbol{G} = \emptyset$ and hence a face of both H_i and G. $\square$

By Lemma 4.12, J has a face whose boundary contains at most 6 branch vertices of J. Let $\boldsymbol{f}$ be such a face, and let $\boldsymbol{f}'$ be an arcwise connected component of $\boldsymbol{f} \cap \boldsymbol{XS}_i$. Such a component exists, because if $\boldsymbol{f} \subseteq \boldsymbol{IS}_i \cup \boldsymbol{BS}_i$, then $\boldsymbol{f} \in \mathfrak{A} \setminus \mathfrak{S}$. But this is a contradiction, because $\boldsymbol{bd}(\boldsymbol{f})$ contains only 6 branch vertices. By Claim 3, $\boldsymbol{f}'$ is a face of both H_i and G. Let $v \in \boldsymbol{bd}(\boldsymbol{f}')$ be a branch vertex of H_i. Then $v \in \boldsymbol{bd}(\boldsymbol{f})$, because all branch vertices of H_i are vertices of J. If v is not a branch vertex of J, then $v \in \boldsymbol{bd}(\boldsymbol{A})$ for some $\boldsymbol{A} \in \mathfrak{S}$ with $v_{\boldsymbol{A}} \in \boldsymbol{bd}(\boldsymbol{f})$. As $v_{\boldsymbol{A}}$ is a branch vertex of J, there are at most 6 such $\boldsymbol{A} \in \mathfrak{S}$ with $v_{\boldsymbol{A}} \in \boldsymbol{bd}(\boldsymbol{f})$. For each of them, there are at most 6 branch vertices of H_i in $\boldsymbol{bd}(\boldsymbol{A}) \cap \boldsymbol{bd}(\boldsymbol{f}')$. Altogether, it follows that $\boldsymbol{bd}(\boldsymbol{f}')$ contains at most 36 branch vertices.

By Fact 4.15(1), $\boldsymbol{bd}(\boldsymbol{f}')$ is a chordless and nonseparating cycle of G. Hence it is suitable. $\square$

Corollary 7.8. *If the graph H_∞ contains no reducing cycle with at most 36 branch vertices, then H_∞ has at most* $\max\{2, 84g\}$ *branch vertices.*

Proof. Recall that $H_\infty = H_{n_\infty}$ and suppose that H_∞ contains no reducing cycle with at most 36 branch vertices, but has more than $\max\{2, 84g\}$ branch vertices. Then by Lemma 7.7, H_∞ has a suitable facial cycle C. This cycle has at most 36 branch vertices and is nonreducing. Hence the face bounded by $\boldsymbol{C}$ is in $F_{n_\infty+1} \setminus F_\infty$, which is a contradiction. $\square$

For every $i \in \mathbb{N} \cup \{\infty\}$, let $\mathcal{F}_i$ be the set of all boundary cycles of faces in F_i. Observe that the sets $IV_i, BV_i, XV_i, IE_i, BE_i, XE_i$ can be defined in terms of $\mathcal{F}_i$ alone with no reference to the embedding of G.

(G) IE_i is the set of all edges of G that appear in precisely two cycles in $\mathcal{F}_i$.
(H) BE_i is the set of all edges of G that appear in precisely one cycle in $\mathcal{F}_i$.
(I) XE_i is the set of all edges of G that appear in no cycle in $\mathcal{F}_i$.
(J) BV_i is the set of all vertices incident to an edge in BE_i.
(K) IV_i is the set of all vertices in $V(G) \setminus BV_i$ that are incident to an edge in IE_i.
(L) $XV_i = V(G) \setminus (IV_i \cup BV_i)$.

Hence H_i can also be defined in terms of $\mathcal{F}_i$ alone.

Lemma 7.9. *For every $i \in \mathbb{N}$, the set $\mathcal{F}_{i+1}$ is the set of all cycles in H_i that are suitable and nonreducing.*

Proof. By Lemma 4.13, all suitable and nonreducing cycles are facial cycles of G. $\square$

Lemma 7.9 is of crucial importance. The sets $\mathcal{F}_i$ where originally defined as sets of facial cycles of the embedded graph G, and also $IV_i, \ldots, XE_i$ and H_i were defined "topologically" by considering the region of $\boldsymbol{S}$ covered by these faces. Lemma 7.9 gives us a definition of the set $\mathcal{F}_i$ and hence of $IV_i, \ldots, XE_i$ and H_i only in terms of the abstract graph G; this definition does not depend on the embedding of G in $\boldsymbol{S}$ in any way. Note that it is important here that we work with nonreducing cycles, which are defined in terms of the abstract graph G, instead of noncontractible cycles, which are defined in terms of the embedding of G.

LEMMA 7.10. *Every nonreducing cycle $C \subseteq H_\infty$ contains at least 3 branch vertices of H_∞.*

PROOF. Suppose for contradiction that H_∞ contains a nonreducing cycle with less than 3 branch vertices of H_∞. Let C be such a cycle of minimum length.

CLAIM 1. *Let $\boldsymbol{e} \in E(H_\infty)$ be an edge incident to at least one vertex that is not a branch vertex of H_∞. Then $\boldsymbol{e}$ is incident with one face in F_∞ and one face in $F(G)\backslash F_\infty$.*

PROOF. Let v, v' be the endvertices of $\boldsymbol{e}$ and suppose that $\deg^{H_\infty}(v) \leq 2$. Let $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ be the two faces of G incident with $\boldsymbol{e}$. If both $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ were contained in F_∞, then $\boldsymbol{e}$ would not be an edge of H_∞. Hence at least one of the faces, say $\boldsymbol{f}_1$, is in $F(G) \setminus F_\infty$. For $i = 1, 2$, let C_i be the boundary cycle of $\boldsymbol{f}_i$ and $\boldsymbol{e}_i \in E(C_i) \setminus \{\boldsymbol{e}\}$ such that $\boldsymbol{e}_i$ is incident with v. By Fact 4.15(2) and because $\boldsymbol{e} \in E(C_1) \cap E(C_2)$, we have $\boldsymbol{e}_1 \neq \boldsymbol{e}_2$. As v is not a branch vertex of H_∞, at least one of the edges $\boldsymbol{e}_1, \boldsymbol{e}_2$ is not an edge of H_∞. We have $\boldsymbol{e}_1 \in E(H_\infty)$, because $\boldsymbol{f}_1 \in F(G)\backslash F_\infty$. Hence $\boldsymbol{e}_2 \notin E(H_\infty)$ and thus $\boldsymbol{f}_2 \in F_\infty$. □

CLAIM 2. *C contains at least two branch vertices of H_∞.*

PROOF. Suppose for contradiction that C contains at most one branch vertex of H_∞. If C contains exactly one branch vertex, let v_0 be this branch vertex, and otherwise let $v_0 \in V(C)$ be arbitrary. Let $v_0\boldsymbol{e}_1v_1\boldsymbol{e}_2\ldots\boldsymbol{e}_nv_n = v_0$ be a simple closed walk around C starting and ending in v_0. The edge $\boldsymbol{e}_1$ is contained in two faces $\boldsymbol{f}, \boldsymbol{f}' \in F(G)$. By Claim 1, one of them, say $\boldsymbol{f}$, is contained in F_∞ and the other one, $\boldsymbol{f}'$, is not contained in F_∞. Let C' be the boundary cycle of $\boldsymbol{f}'$ and $\boldsymbol{e}_2'$ the edge on C' after $\boldsymbol{e}_1$, that is, the unique edge in $E(C')\setminus\{\boldsymbol{e}_1\}$ incident with v_1. Then $\boldsymbol{e}_2' \in E(H_\infty)$, because it is incident with a face not in F_∞. As v_1 is not a branch vertex of H_∞, it follows that $\boldsymbol{e}_2 = \boldsymbol{e}_2'$. Applying the same argument inductively, we see that $C = C'$.

Hence C is a facial cycle and therefore chordless and nonseparating by Fact 4.15(1). We assumed that C contains only 1, hence at most 36, branch vertices. Thus C is a suitable cycle. Furthermore, C is nonreducing. Thus $C' = C \in \mathcal{F}_\infty$ and $\boldsymbol{f}' \in F_\infty$. This is a contradiction. □

Thus C contains exactly two branch vertices v and v'. Our idea to derive a contradiction is as follows. $\boldsymbol{C}$ must be a separating simple closed curve of $\boldsymbol{S}$, because it C nonreducing and hence contractible. However, the only connection from one side of C to the other is through the branch vertices v, v'. This contradicts the 3-connectedness of G. We will carry out the argument in detail now.

Let P_1, P_2 be the two segments of C connecting v and v'.

CLAIM 3. *For $i = 1, 2$ there is a face $\boldsymbol{f}_i \in F(G)\backslash F_\infty$ such that $P_i \subseteq Bd(\boldsymbol{f}_i)$.*

PROOF. Let $i \in [2]$. Let $v_0\boldsymbol{e}_1v_1\boldsymbol{e}_2\ldots\boldsymbol{e}_nv_n$, where $v_0 = v$ and $v_n = v'$, be a simple walk along P_i. If $n = 1$, then $\boldsymbol{P}_i = \boldsymbol{e}_1$, and as an edge of H_∞, the edge $\boldsymbol{e}_1$ is contained in the boundary of some face $\boldsymbol{f} \in F(G)\backslash F_\infty$. In the following, suppose that $n \geq 2$. Then $v_1, \ldots, v_{n-1}$ are vertices of degree 2 in H_∞. Let $\boldsymbol{f}, \boldsymbol{f}' \in F(G)$ be the two faces incident with $\boldsymbol{e}_1$. By Claim 1, one of them, say $\boldsymbol{f}$, is contained in F_∞ and the other one, $\boldsymbol{f}'$, is

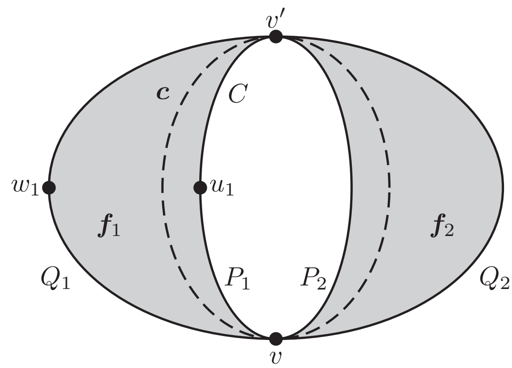


Fig. 7.1. Proof of Lemma 7.10.

not contained in F_∞. Let C' be the boundary cycle of $\boldsymbol{f}'$ and $\boldsymbol{e}_2'$ the edge on C' after $\boldsymbol{e}_1$, that is, the unique edge in $E(C') \setminus \{\boldsymbol{e}_1\}$ incident with v_1. Then $\boldsymbol{e}_2' \in E(H_\infty)$. As v_1 is not a branch vertex of H_∞, it follows that $\boldsymbol{e}_2 = \boldsymbol{e}_2'$. Applying the same argument inductively, we see that $P_i \subseteq C'$, and we let $\boldsymbol{f}_i := \boldsymbol{f}'$. $\square$

The following argument is illustrated by Figure 7.1. As G is a simple graph, at least one of the paths P_1, P_2, say P_1, is of length at least 2. Let $u_1 \in V(P_1) \setminus \{v, v'\}$. Choose faces $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ according to Claim 3. For $i = 1, 2$, let C_i be the boundary cycle of $\boldsymbol{f}_i$, and let $Q_i = C_i \setminus (V(P_i) \setminus \{v, v'\})$ be the segment of C_i that connects v and v' and is different from P_i.

Claim 4. *The path Q_1 has length at least* 2.

Proof. Suppose for contradiction that Q_1 has length 1, that is, consists of a single edge. Then $|Q_1| < |P_1|$. Let $C' := Q_1 \cup P_2$. Then $V(C') \subseteq V(C)$, and thus C' is nonreducing. Furthermore, $|C'| < |C|$, which contradicts the minimality of C. $\square$

Let $w_1 \in V(Q_1) \setminus \{v, v'\}$. We can find a simple closed curve $\boldsymbol{c} \in \boldsymbol{S}$ that is homotopic to $\boldsymbol{C}$ and contained in $\boldsymbol{f}_1 \cup \boldsymbol{f}_2 \cup \{v, v'\}$. (We have not explained the term homotopic. For readers not familiar with it, it will be sufficient to assume that $\boldsymbol{c}$ is the boundary of a disk in $\boldsymbol{S}$ if and only if $\boldsymbol{C}$ is. We can find such a curve by following $\boldsymbol{C}$ very closely, but in the interior of the faces $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$.) As C is a nonreducing cycle, the simple closed curve $\boldsymbol{C}$ is contractible (that is, the boundary of a disk in $\boldsymbol{S}$), and hence $\boldsymbol{c}$ is contractible as well.

Claim 5. $\{v, v'\}$ *separates u_1 from w_1.*

Proof. As G is 2-cell embedded, $\boldsymbol{f}_1$ is an open disk, and hence $\boldsymbol{d} := \boldsymbol{cl}(\boldsymbol{f}_1)$ is a closed disk. Let $\boldsymbol{p}$ be the segment of $\boldsymbol{c}$ from v to v' that is contained in $\boldsymbol{d}$.

We first show that $\boldsymbol{d} \setminus \boldsymbol{c} = \boldsymbol{d} \setminus \boldsymbol{p}$ has two arcwise connected components $\boldsymbol{a}, \boldsymbol{b}$ such that $w_1 \in \boldsymbol{a}$ and $u_1 \in \boldsymbol{b}$. We can "glue a disk on the boundary of the disk $\boldsymbol{d}$" and obtain a sphere $\boldsymbol{s} \supseteq \boldsymbol{d}$. (This sphere $\boldsymbol{s}$ has nothing to do with the surface $\boldsymbol{S}$, it is just a "virtual" surface we create to be able to apply Fact 4.1.) Note that now we have three mutually disjoint simple curves $\boldsymbol{P}_1, \boldsymbol{Q}_1, \boldsymbol{p} \subseteq \boldsymbol{s}$ with the same endpoints v, v'. Hence we can apply Fact 4.1 (with $\boldsymbol{g}_1 := \boldsymbol{P}_1$, $\boldsymbol{g}_2 := \boldsymbol{Q}_1$, $\boldsymbol{g}_3 := \boldsymbol{p}$ and $x_1 := u_1$, $x_2 := w_1$). Note that the arcwise connected component of $\boldsymbol{s} \setminus (\boldsymbol{P}_1 \cup \boldsymbol{Q}_1 \cup \boldsymbol{p})$ with boundary $\boldsymbol{P}_1 \cup \boldsymbol{Q}_1$ is precisely $\boldsymbol{s} \setminus \boldsymbol{d}$. Thus there is no simple curve $\boldsymbol{h} \subseteq \boldsymbol{d}$ from u_1 to w_1 that has an empty intersection with $\boldsymbol{p}$.

As $\boldsymbol{c}$ is contractible in $\boldsymbol{S}$, it is separating, and thus $\boldsymbol{S} \setminus \boldsymbol{c}$ has two arcwise connected components $\boldsymbol{A}, \boldsymbol{B}$ with $\boldsymbol{bd}(\boldsymbol{A}) = \boldsymbol{bd}(\boldsymbol{B}) = \boldsymbol{c}$. The components $\boldsymbol{a}, \boldsymbol{b}$ of $\boldsymbol{d} \setminus \boldsymbol{c}$ cannot be contained in the same component, because both $\boldsymbol{A}$ and $\boldsymbol{B}$ must contain points arbitrarily close to $\boldsymbol{c}$ in $\boldsymbol{d}$. Say, $\boldsymbol{a} \subseteq \boldsymbol{A}$ and $\boldsymbol{b} \subseteq \boldsymbol{B}$. Thus $u_1 \in \boldsymbol{A}$ and $w_1 \in \boldsymbol{B}$.

Now $P \subseteq G$ be a path from u_1 to w_1. Then $\boldsymbol{P}$ is a simple curve in $\boldsymbol{S}$ from u_1 to w_1, and we must have $\boldsymbol{P} \cap \boldsymbol{c} \neq \emptyset$. Since we have $\boldsymbol{c} \cap \boldsymbol{G} = \{v, v'\}$, it follows that $V(P) \cap \{v, v'\} \neq \emptyset$. Thus $\{v, v'\}$ separates u_1 from w_1. □

Claim 5 contradicts G being 3-connected. □

The following consequence of Lemma 7.10 is what we need later.

COROLLARY 7.11. *Suppose that H_∞ contains no reducing cycle with less than 3 branch vertices. Then for all $v, w \in V(H_\infty)$ there is at most one isolated path from v to w in H_∞.*

PROOF. Suppose for contradiction that there are vertices $v, w \in V(H_\infty)$ and two distinct isolated paths $P, Q \subseteq V(H_\infty)$ from v to w. Then P and Q are internally disjoint, because otherwise they would not be isolated. Hence $C := P \cup Q$ is a cycle, and this cycle contains at most 2 branch vertices (v and w). By the assumption of the corollary, C is nonreducing. This contradicts Lemma 7.10. □

LEMMA 7.12. *There are* IFP*-formulae* $\mathsf{angle}_g(x_1, x_2, x_3)$ *and* $\mathsf{aligned}_g(x_1, x_2, x_3, x_4)$ *such that for all vertices $v_1, v_2, v_3, v_4 \in V(G)$ we have*

$$\begin{aligned} G \models \mathsf{angle}_g[v_1, v_2, v_3] &\iff (v_1, v_2, v_3) \in \angle(F_\infty), \\ G \models \mathsf{aligned}_g[v_1, v_2, v_3, v_4] &\iff (v_1, v_2, v_3), (v_2, v_3, v_4) \in \angle(F_\infty) \\ &\qquad \textit{and } (v_1, v_2, v_3) \curvearrowright (v_2, v_3, v_4). \end{aligned}$$

Of course the formulae $\mathsf{angle}_g(x_1, x_2, x_3)$ and $\mathsf{aligned}_g(x_1, x_2, x_3, x_4)$ of the lemma do not depend on the specific graph G, but just on g and the od-scheme Λ^{g-1}.

PROOF. In the first step of the proof, we formalize the inductive step of an inductive definition of the set $\angle(F_{i+1})$ from $\angle(F_i)$.

CLAIM 1. *Let X be a ternary relation variable. There are* IFP*-formulae* $\mathsf{iv}(X, x)$, $\mathsf{bv}(X, y)$, $\mathsf{xv}(X, x)$, $\mathsf{ie}(X, x_1, x_2)$, $\mathsf{be}(X, x_1, x_2)$, $\mathsf{xe}(X, x_1, x_2)$, $\mathsf{vertH}(X, x)$, $\mathsf{edgeH}(X, x_1, x_2)$, $\mathsf{ang}(X, x_1, x_2, x_3)$, *and* $\mathsf{aln}(X, x_1, x_2, x_3, x_4)$ *such that for all $i \in \mathbb{N}$ we have*

$$\begin{aligned} \mathsf{iv}[G, \angle(F_i), x] &= IV_i, & \mathsf{ie}[G, \angle(F_i), x_1, x_2] &= IE_i, \\ \mathsf{bv}[G, \angle(F_i), x] &= BV_i, & \mathsf{be}[G, \angle(F_i), x_1, x_2] &= BE_i, \\ \mathsf{xv}[G, \angle(F_i), x] &= XV_i, & \mathsf{xe}[G, \angle(F_i), x_1, x_2] &= XE_i, \\ \mathsf{vertH}[G, \angle(F_i), x] &= V(H_i), & \mathsf{edgeH}[G, \angle(F_i), x_1, x_2] &= E(H_i), \end{aligned}$$

and

$$\begin{aligned} \mathsf{ang}[G, \angle(F_i), x_1, x_2, x_3] &= \angle(F_{i+1}), \\ \mathsf{aln}[G, \angle(F_i), x_1, x_2, x_3, x_4] &= \big\{(v_1, v_2, v_3, v_4) \bigm| (v_1, v_2, v_3), (v_2, v_3, v_4) \in \angle(F_{i+1}) \\ &\qquad\qquad \textit{and } (v_1, v_2, v_3) \curvearrowright (v_2, v_3, v_4)\big\}. \end{aligned}$$

PROOF. By Lemma 5.1, for every angle $\bar{v}$ of G there is exactly one face $\boldsymbol{f} \in F(G)$ such that $\bar{v}$ is incident to $\boldsymbol{f}$. Hence the angles in $\angle(F_i)$ determine the cycles in $\mathcal{F}_i$, and we can use (G)–(L) to define the formulae $\mathsf{iv}, \mathsf{bv}, \dots, \mathsf{xe}$. For example, we let

$$\begin{aligned} \mathsf{be}(X, x_1, x_2) &:= \exists x_3 (X(x_1, x_2, x_3) \wedge \forall x_3' (X(x_1, x_2, x_3') \rightarrow x_3 = x_3')); \\ \mathsf{bv}(X, x) &:= \exists x' \, \mathsf{be}(X, x, x'). \end{aligned}$$

Using these formulae, we can define the vertex set and the edge relation of the graph H_i by

$$\begin{aligned}\mathsf{vertH}(X, x) &:= \mathsf{bv}(X, x) \vee \mathsf{xv}(X, x),\\ \mathsf{edgeH}(X, x_1, x_2) &:= \mathsf{be}(X, x_1, x_2) \vee \mathsf{xe}(X, x_1, x_2).\end{aligned}$$

To define the suitable cycles, we need further auxiliary formulae:

—a formula $\mathsf{branch}(X, x)$ such that $\mathsf{branch}[G, \angle(F_i), x]$ is the set of all branch vertices of H_i,
—a formula $\mathsf{isopath}(X, x, x', y, y')$ such that for all $v, v', w, w' \in V(G)$ we have $G \models \mathsf{isopath}[\angle(F_i), v, v', w, w']$ if an only if there is an isolated path P in H_i from v to v' such that $ww' \in E(P)$,
—a formula $\mathsf{isopath}'(X, x, x', y, y', z, z')$ such that for all $v, v', w, w', u, u' \in V(G)$ we have $G \models \mathsf{isopath}'[\angle(F_i), v, v', w, w', u, u']$ if an only if there is an isolated path P in H_i from v to v' such that $ww', uu' \in E(P)$.

Using the Transduction Lemma (Fact 2.9) and the formulae vertH and edgeH, it is easy to define such a formulae in IFP.

We can specify a cycle $C \subseteq H_i$ with $\ell \geq 1$ branch vertices by fixing its ℓ branch vertices, say, $v_1, v_3, v_5, \ldots, v_{2\ell-1}$, and for every $i \in [1, \ell]$ fixing the first vertex v_{2i} on the isolated path from v_{2i-1} to v_{2i+1}, where we let $v_{2\ell+1} := v_1$. If the path from v_{2i-1} to v_{2i+1} has length 1, we simply let $v_{2i} = v_{2i+1}$. We call C the cycle *specified* by $\bar{v} = (v_1, \ldots, v_{2\ell})$. We define an IFP-formula $\mathsf{cycle}_\ell(X, \bar{x}, y, y')$ such that for all $\bar{v} \in V(G)^{2\ell}$, the binary relation $\mathsf{cycle}_\ell[G, \angle(F_i), \bar{v}, y, y']$ is the edge relation of the cycle specified by $\bar{v}$, if $\bar{v}$ specifies a cycle in H_i, and $\mathsf{cycle}_\ell[G, \angle(F_i), \bar{v}, y, y'] = \emptyset$ otherwise. We let

$$\begin{aligned}&\mathsf{cycle}_\ell(X, x_1, x_2, \ldots, x_{2\ell}, y, y')\\ &\quad := \bigwedge_{i=1}^{\ell} \mathsf{branch}(X, x_{2i-1})\\ &\quad\wedge \bigwedge_{i=1}^{\ell-1} \mathsf{isopath}(X, x_{2i-1}, x_{2i+1}, x_{2i-1}, x_{2i}) \wedge \mathsf{isopath}(X, x_{2\ell-1}, x_1, x_{2\ell-1}, x_{2\ell})\\ &\quad\wedge\Big(\bigvee_{i=1}^{\ell-1} \mathsf{isopath}'(X, x_{2i-1}, x_{2i+1}, x_{2i-1}, x_{2i}, y, y')\\ &\qquad\quad \vee \mathsf{isopath}'(X, x_{2\ell-1}, x_1, x_{2\ell-1}, x_{2\ell}, y, y')\Big).\end{aligned}$$

Next, we define formulae $\mathsf{chordless}_\ell(X, \bar{x})$ and $\mathsf{non\text{-}sep}_\ell(X, \bar{x})$ such that for all $\bar{v} \in V(G)^{2\ell}$ specifying a cycle C in H_i we have:

—$G \models \mathsf{chordless}_\ell[\angle(F_i), \bar{v}]$ if and only if the cycle C specified by $\bar{v}$ is chordless in G;
—$G \models \mathsf{non\text{-}sep}_\ell[\angle(F_i), \bar{v}]$ if and only if the cycle C specified by $\bar{v}$ is nonseparating in G.

It is easy to define such IFP-formulae (refer to the proof of Lemma 6.6). Note that the properties of the cycle refer to the graph G and not to H_i.

Let φ_{g-1} be an IFP-sentence that defines the class $\mathcal{D}_{g-1}$. Such a sentence exists by Lemma 3.13, because $\mathcal{D}_{g-1} = \mathcal{OT}_{\Lambda^{g-1}}$ (see Assumption 7.4). Using this sentence, we can define an IFP-formula $\mathsf{non\text{-}red}_\ell(X, \bar{x})$ such that for all $\bar{v} \in V(G)^{2\ell}$ specifying a cycle C in H_i we have $G \models \mathsf{non\text{-}red}_\ell[\angle(F_i), \bar{v}]$ if and only if the cycle C specified by $\bar{v}$ is nonreducing in G. We let

$$\mathsf{good\text{-}cycle}_\ell(X, \bar{x}) := \mathsf{chordless}_\ell(X, \bar{x}) \wedge \mathsf{non\text{-}sep}_\ell(X, \bar{z}x) \wedge \mathsf{non\text{-}red}_\ell(X, \bar{x}).$$

Then the cycles specified by the 2ℓ-tuples $\bar{v} \in \mathsf{good\text{-}cycle}_\ell[\angle(F_i), \bar{x}]$ are precisely the suitable and nonreducing cycles with ℓ branch vertices. Recall that $\mathcal{F}_{i+1}$ consists of theses cycles for $\ell = 1, \dots, 36$. Now we let

$$\mathsf{ang}(X, x_1, x_2, x_3) := \bigvee_{\ell=1}^{36} \exists \bar{z}\big(\mathsf{good\text{-}cycle}_\ell(X, \bar{z}) \wedge \mathsf{cycle}_\ell(X, \bar{z}, x_1, x_2) \\ \wedge \mathsf{cycle}_\ell(X, \bar{z}, x_2, x_3) \wedge x_1 \neq x_3\big).$$

We define the formula $\mathsf{aln}(X, x_1, x_2, x_3, x_4)$ similarly.

It follows from Claim 1 that the formula

$$\mathsf{angle}_g(x_1, x_2, x_3) := \mathrm{ifp}(X(x_1, x_2, x_3) \leftarrow \mathsf{ang}(X, x_1, x_2, x_3))(x_1, x_2, x_3)$$

defines the set $\angle(F_\infty)$. To complete the proof of the lemma, we let $\mathsf{aligned}_g(x_1, x_2, x_3, x_4)$ be the formula obtained from the formula $\mathsf{aln}(X, x_1, x_2, x_3, x_4)$ by replacing each atom of the form $X(x_1, x_2, x_3)$ by the formula $\mathsf{angle}_g(x_1, x_2, x_3)$. $\square$

Note that the proof of Lemma 7.12 also yields the following corollary.

COROLLARY 7.13. *There are* IFP*-formulae* $\mathsf{vertH}(x)$ *and* $\mathsf{edgeH}(x_1, x_2)$ *such that*

$$H_\infty = \big(\mathsf{vertH}[G, x], \mathsf{edgeH}[G, x_1, x_2]\big).$$

PROOF. In the proof of Lemma 7.12 we constructed IFP-formulae $\mathsf{vertH}(X, x)$ and $\mathsf{edgeH}(X, x_1, x_2)$ with a free ternary relation variable X such that $\mathsf{vertH}[G, \angle(F_i), x] = V(H_i)$ and $\mathsf{edgeH}[G, \angle(F_i), x_1, x_2] = E(H_i)$. Then $\mathsf{vertH}[G, \angle(F_\infty), x] = V(H_\infty)$ and $\mathsf{edgeH}[G, \angle(F_\infty), x_1, x_2] = E(H_\infty)$. Thus if we replace each subformula of $\mathsf{vertH}(X, x)$ and $\mathsf{edgeH}(X, x_1, x_2)$ of the form $X(z_1, z_2, z_3)$ by $\mathsf{angle}_g(z_1, z_2, z_3)$, we obtain the desired formulae. $\square$

7.2. Proof of Lemma 7.3

In this subsection, we still make Assumption 7.4 (but no longer Assumption 7.5). Let $\mathcal{E}$ be the class of 3-connected graphs that are embeddable in $\boldsymbol{S}$. We classify the graphs in $\mathcal{E}$ into finitely many types and prove the existence of definable ordered treelike decompositions for each type. Then we apply Lemma 3.16(1) to combine the decompositions.

A graph $G \in \mathcal{E}$ is of *type I* if there is a set $W \subseteq V(G)$ such that $|W| \leq 3$ and $G \setminus W \in \mathcal{D}_{g-1}$. Let $\mathcal{Y}_I$ be the class of all graphs G of type I.

LEMMA 7.14. *The class* $\mathcal{Y}_I$ *admits* IFP*-definable ordered treelike decompositions.*

PROOF. This follows directly from Assumption 7.4 and the Finite Extension Lemma (Corollary 3.19). $\square$

Let $G \in \mathcal{E} \setminus \mathcal{Y}_I$. As $G \in \mathcal{E}$, we may assume that that G is actually embedded in $\boldsymbol{S}$. Observe that the embedding is polyhedral. To see this, note that the representativity of the embedding is at least 3, because $G \notin \mathcal{Y}_I$. As G is 3-connected, this means that the embedding is polyhedral.

Thus G satisfies Assumption 7.5, and we can define the graph H_∞ as in the previous subsection. The graph G is of *type II* if there is a cycle $C \subseteq H_\infty$ that contains at most 36 branch vertices of H_∞ and is a reducing cycle of G. Let $\mathcal{Y}_{II}$ be the class of all graphs of type II.

LEMMA 7.15. *The class* $\mathcal{Y}_{II}$ *admits* IFP*-definable ordered treelike decompositions.*

PROOF. It follows from Corollary 7.13 and the definition of type II that there are formulae $\mathsf{cycleV}(x)$ and $\mathsf{cycleE}(x, y)$ such that for every graph $G \in \mathcal{Y}_{II}$, the cycle

$$\big(\mathsf{cycleV}[G, x], \mathsf{cycleE}[G, x, y]\big)$$

is a reducing cycle with at most 36 branch vertices of H_∞. Since the class of all cycles admits IFP-definable orders, it follows from the Ordered Extension Lemma 3.18 and Assumption 7.4 that $\mathcal{Y}_{II}$ admits IFP-definable ordered tree decompositions. □

A graph $G \in \mathcal{E}$ is of *type III* if it is not of type I or II. We let $\mathcal{Y}_{III}$ be the class of all graphs of type III.

LEMMA 7.16. *The class $\mathcal{Y}_{III}$ admits* IFP*-definable orders.*

PROOF. Let $G \in \mathcal{Y}_{III}$. Without loss of generality we may assume that G is polyhedrally embedded in $\boldsymbol{S}$. We define H_∞ as usual. We want to apply the Angle Lemma 5.3 or the Partial Angle Lemma 5.4 to the formulae $\mathsf{angle}_g(\bar{x}_1, y_1, y_2, y_3)$, $\mathsf{align}_g(\bar{x}_2, y_1, y_2, y_3, y_4)$ and the set F_∞ of faces (as $\mathcal{F}$ in the Partial Angle Lemma). If $F_\infty = F(G)$, we can simply apply the Angle Lemma 5.3 to define a linear order on $V(G)$. Suppose that $F_\infty \subset F(G)$. Then the set W of the Partial Angle Lemma is $V(H_\infty)$. Thus we have to define a formula $\mathsf{rest\text{-}ord}(\bar{x}, y_1, y_2)$ such that for some tuple $\bar{u} \in G^{\bar{x}}$, the binary relation $\mathsf{rest\text{-}ord}[G, \bar{u}, y_1, y_2]$ is a linear order of $V(H_\infty)$. By Corollary 7.8, the graph H_∞ contains at most $k := \max\{2, 84g\}$ branch vertices, and by Corollary 7.11, there is at most one isolated path between any two branch vertices of H_∞. Let $u_1, \ldots, u_{k'}$ for some $k' \le k$ be an enumeration of all branch vertices of H_∞, and let

$$\bar{u} := (u_1, u_2, \ldots, u_{k'}, \underbrace{u_{k'}, \ldots, u_{k'}}_{k-k' \text{ times}}).$$

Let $U := \widetilde{u}$, and let $\le_U$ be the linear order on U defined by $u_i \le_U u_j :\Leftrightarrow i \le j$. Note that all vertices in $v \in W \setminus U$ appear on an isolated path with endvertices in U. Let $p(v) \le_U q(v)$ be the two endvertices of this isolated path. Let $\le$ be the linear order on W defined as follows. For all $v, v' \in W$,

$$v \le v' \;:\Leftrightarrow \begin{cases} & v, v' \in U \text{ and } v \le_U v', \\ \text{or} & v \in U \text{ and } v' \notin U \\ \text{or} & v, v' \notin U \text{ and } p(v) <_U p(v'), \\ \text{or} & v, v' \notin U \text{ and } p(v) = p(v') \text{ and } q(v) <_U q(v'), \\ \text{or} & v, v' \notin U \text{ and } p(v) = p(v') \text{ and } q(v) = q(v'), \\ & \text{and } v \text{ appears before } v' \text{ on the isolated path from } p(v) \text{ to } q(v), \\ \text{or} & v = v'. \end{cases}$$

This defines a linear order on W, because there is at most one isolated path between any two branch vertices of H_∞. It is straightforward to define this linear order in IFP. Hence an application of the Partial Angle Lemma 5.4 completes the proof. □

PROOF OF LEMMA 7.3. As $\mathcal{E} = \mathcal{Y}_I \cup \mathcal{Y}_{II} \cup \mathcal{Y}_{III}$, the lemma follows immediately from the previous three lemmas and Lemma 3.16(1) . □

Remark 7.17. For $g = 0$, the classes $\mathcal{Y}_I$ and $\mathcal{Y}_{II}$ are empty. Hence Lemma 7.16 implies Theorem 6.1.

8. GRAPHS WITH EXCLUDED MINORS

In this section, we give a high-level outline of the proof of Theorem 1.3. The full proof can be found in Grohe [2012]. It should be no surprise at this point that the proof is based on ordered treelike decompositions. The main technical result is the following structure theorem.

THEOREM 8.1 (DEFINABLE STRUCTURE THEOREM). *Let $\mathcal{C}$ be a class of graphs with excluded minors. Then $\mathcal{C}$ admits* IFP*-definable ordered treelike decompositions.*

Theorem 1.3 follows.

Another corollary of the Definable Structure Theorem is the IFP+C-definability of classes closed under taking minors.

COROLLARY 8.2. *Every class $\mathcal{C}$ of graphs that is closed under taking minors is* IFP+C-*definable.*

This is proved analogously to Corollary 7.2, using the fact due to Robertson and Seymour [1995] that all classes of graphs closed under taking minors are recognizable in cubic time. I conjecture that all classes closed under taking minors are actually definable in IFP, but have to leave this open.

The proof of the Definable Structure Theorem 8.1 is long, in fact it spans the whole second part of the monograph [Grohe 2012]. Here we can only highlight the key steps of the proof. It builds on Robertson and Seymour's structure theory for graphs with excluded minors, and we will start by describing this theory.

8.1. The Structure of Graphs with Excluded Minors

In this subsection, we will give the precise statement of Robertson and Seymour's [1999] structure theorem for graphs with excluded minors, which intuitively says that such graphs can be decomposed into pieces that are almost embeddable in a surface. This structure theorem plays a central role in Robertson and Seymour's proof of their celebrated Graph Minor Theorem [Robertson and Seymour 2004], stating that every class of graphs that is closed under taking minors can be characterized by finitely many excluded minors. The structure theorem has also found various algorithmic applications (e.g., Robertson and Seymour [1995] and Demaine et al. [2005]).

To explain what it means for a graph to be almost embeddable in a surface, we need to introduce the fairly technical notion of a *p-ring*. A *path decomposition* of a graph G is a tree decomposition (P, β) of G where P is a path. Let P_n denote the path with vertex set $[n] := \{1, \dots, n\}$ and edges from i to $(i+1)$ for all $i < n$. A *p-ring* is a pair $(R, \bar{r})$, where R is a graph and $\bar{r} = (r_1, \dots, r_n) \in V(R)^n$ is a tuple of pairwise distinct vertices of R such that there exists a path decomposition (P_n, β) of R of width at most $(p-1)$ with $r_i \in \beta(i)$ for all $i \in [n]$.

Example 8.3. Let R be a cycle of length n with chords between all pairs of vertices of distance 2 on the cycle. To be precise, say that $V(R) := [n]$ and $E(R) := \{i(i+1), i(i+2) \mid i \in [n]\}$, where addition is taken modulo n. Let $\bar{r} = (r_1, \dots, r_n)$ with $r_i := i$. Then $(R, \bar{r})$ is a 5-ring. Indeed, (P_n, β) with $\beta(1) := \{1\}$, $\beta(2) := \{1, 2\}$, and $\beta(i) := \{i, i-1, i-2, 2, 1\}$ for all $i \in [3, n]$ is a path decomposition of R of width 4 with $r_i \in \beta(i)$ for all $i \in [n]$.

The reason we need rings to describe the structure of graphs with excluded minors are graphs like the one in Figure 8.2 (also see Example 8.6), which should be thought of as the union of a plane graph with a ring of small width glued to the boundary of a face (the exterior face in Figure 8.2). The family of graphs constructed this way by gluing longer and longer rings to the boundary of a face of a plane graph (we call such graphs *almost planar*) has unbounded genus, and if the plane part is sufficiently connected, it cannot reasonably be cut into smaller pieces. Yet it does not contain any large complete graphs as minors.

The graph in Figure 8.2 may explain how rings got their name, but it also suggests that we should base rings on decompositions over cycles instead of paths. We could do

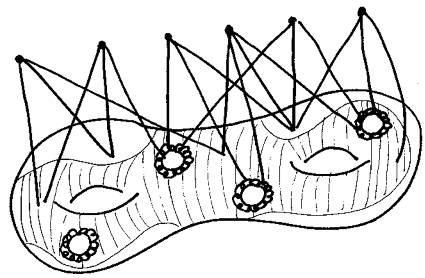


Fig. 8.1. A graph almost embedded in a double torus with four rings and six apices.

that, but it would not make a difference for the structure theory and only complicate things.

It will be convenient in the following to work with surfaces with boundary. It is a basic fact that each surface $\boldsymbol{S}$ with boundary can be obtained from a surface $\overline{\boldsymbol{S}}$ without boundary by deleting the interior of finitely many mutually disjoint closed disks, say, $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_q$. Hence the boundary of $\boldsymbol{S}$ consists of the q mutually disjoint simple closed curves $\boldsymbol{bd}(\boldsymbol{D}_i)$, which are called the *cuffs* of $\boldsymbol{S}$. It turns out that it does not matter which disks are deleted, in the sense that every other surface with boundary that is obtained from $\overline{\boldsymbol{S}}$ by deleting the interior of q mutually disjoint closed disks is homeomorphic to $\boldsymbol{S}$. Thus the classification of surfaces without boundary into the two families $(\boldsymbol{S}_g)_{g\in\mathbb{N}}$ and $(\boldsymbol{N}_h)_{h\in\mathbb{N}^+}$ yields the following classification of surfaces with boundary. Every orientable surface with boundary is homeomorphic to a surface $\boldsymbol{S}_{g,q}$ obtained from $\boldsymbol{S}_g$ by deleting the interior of q mutually disjoint closed disks, and every nonorientable surface with boundary is homeomorphic to a surface $\boldsymbol{N}_{h,q}$ obtained from $\boldsymbol{N}_h$ by deleting the interior of q mutually disjoint closed disks. If $\boldsymbol{S} \simeq \boldsymbol{S}_{g,q}$, then we let $\mathrm{eg}(\boldsymbol{S}) := 2g$ and $\mathrm{cf}(\boldsymbol{S}) := q$, and if $\boldsymbol{S} \simeq \boldsymbol{N}_{h,q}$, then we let $\mathrm{eg}(\boldsymbol{S}) := h$ and $\mathrm{cf}(\boldsymbol{S}) := q$. Even without explicitly referring to an ambient surface $\overline{\boldsymbol{S}}$, we denote the boundary (that is, the union of all cuffs) of a surface $\boldsymbol{S}$ with boundary by $\boldsymbol{bd}(\boldsymbol{S})$. Note that for every surface $\boldsymbol{S}$ with boundary obtained from a surface $\overline{\boldsymbol{S}}$ by deleting the interior of finitely many mutually disjoint closed disks, a graph is embeddable in $\boldsymbol{S}$ if and only if it is embeddable in $\overline{\boldsymbol{S}}$, because we can always place the deleted disks in the interior of faces of G. (This is why we only considered surfaces without boundary in the previous section.)

Intuitively, a graph is almost embeddable in a surface $\boldsymbol{S}$ with boundary if after removing a bounded number of vertices (called *apices*) the graph can be embedded in $\boldsymbol{S}$ except for a bounded number of nonembeddable regions, and each of these regions is a ring of bounded width attached to a cuff of the surface. The next definition makes this precise. Figure 8.1 illustrates the definition.

Definition 8.4. Let G be a graph.

(1) Let $\boldsymbol{S}$ be a surface with boundary, and let $\boldsymbol{c}^1, \ldots, \boldsymbol{c}^q$ be the cuffs of $\boldsymbol{S}$. An *arrangement*[6] of a graph G in $\boldsymbol{S}$ is a tuple

$$(G_0, \pi, R^1, \bar{r}^1, \ldots, R^q, \bar{r}^q),$$

[6]The definition of arrangements given here is from Grohe [2012]. (They are called *injective arrangements* there.) Robertson and Seymour [1999] define a different, but closely related notion of arrangement (refer to Section 17.1.2 of Grohe [2012]).

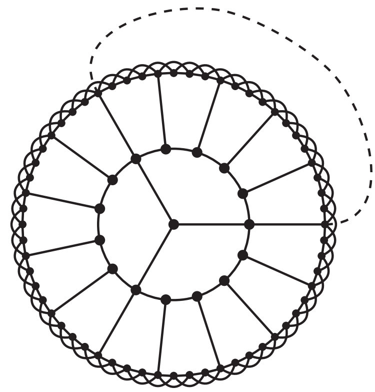


Fig. 8.2. A 5-almost planar graph (the outer edges connect vertices of distance two on the cycle).

where G_0 is a graph embedded in $\boldsymbol{S}$ and $\pi : V(G_0) \to V(G)$ and for all $i \in [q]$ we have $R^i \subseteq G$, and $\bar{r}^i = (r^i_1, \dots, r^i_{n_i}) \in V(G_0)^{n_i}$, and the following conditions are satisfied.

(A.1) π is injective.
(A.2) $G = \pi(G_0) \cup R^1 \cup \dots \cup R^q$.
(A.3) For all $i \in [q]$ it holds that $E(\pi(G_0)) \cap E(R^i) = \emptyset$.
(A.4) For all $i \in [q]$ it holds that $V(\pi(G_0)) \cap V(R^i) = \pi(\widetilde{r}^i)$.
(A.5) For all $i \in [q]$ it holds that $\boldsymbol{G}_0 \cap \boldsymbol{c}^i = \widetilde{r}^i$, and the vertices $r^i_1, \dots, r^i_{n_i}$ appear on $\boldsymbol{c}^i$ in cyclic order.
(A.6) For all $i, j \in [q]$ with $i \neq j$ it holds that $R^i \cap R^j = \emptyset$.

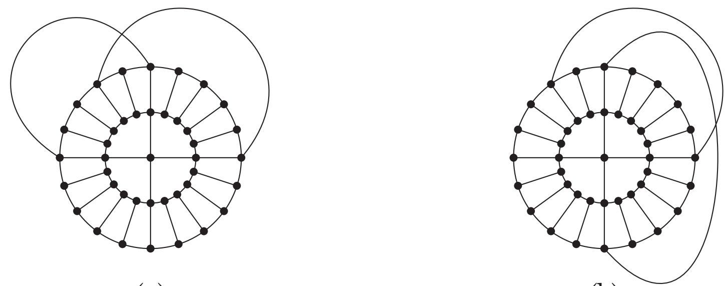
(2) Let $p \geq 1$. An arrangement $(G_0, \pi, R^1, \bar{r}^1, \dots, R^q, \bar{r}^q)$ of G in a surface $\boldsymbol{S}$ is a *p-arrangement of G* if for all $i \in [q]$ the pair $(R^i, \pi(\bar{r}^i))$ is a p-ring.
(3) Let $p, q, r, s \in \mathbb{N}$. The graph G is *(p, q, r, s)-almost embeddable* if there is a set $X \subseteq V(G)$ such that $|X| \leq s$ and $G \setminus X$ has a p-arrangement in a surface $\boldsymbol{S}$ of Euler genus r with q cuffs.
We denote the class of all (p, q, r, s)-almost embeddable graphs by $\mathcal{AE}_{p,q,r,s}$.

FACT 8.5 (STRUCTURE THEOREM [ROBERTSON AND SEYMOUR 1999]). *Let $\mathcal{C}$ be a class of graphs with excluded minors. Then there exist $p, q, r, s \in \mathbb{N}^+$ such that every graph in $\mathcal{C}$ has a tree decomposition over $\mathcal{AE}_{p,q,r,s}$.*

Actually, we use a stronger variant of the theorem (Theorem 3.1 of Robertson and Seymour [1999]). Intuitively it says that for every class $\mathcal{C}$ excluding a minor there are suitable parameters such the for every graph $G \in \mathcal{C}$, every "highly connected region" in G is almost embeddable in some surface. The formal definition of highly connected regions is technically difficult; it involves so-called *tangles* [Robertson and Seymour 1991]. Let us call this version of the theorem the *Local Structure Theorem*. It can be shown by fairly generic techniques [Robertson and Seymour 1991, 1999] that the Local Structure Theorem implies the "Global" Structure Theorem (Fact 8.5).

8.2. Almost Planar Graphs

Let us call a graph *p-almost planar* if it has a p-arrangement in a disk, and let $\mathcal{AP}_p$ denote the class of all p-almost planar graphs. Note that $\mathcal{AP}_p = \mathcal{AE}_{p,1,0,0}$.



(a) (b)

Fig. 8.3.

Example 8.6. Figure 8.2 shows a nasty example of a 5-almost planar graph (the graph without the dashed edge). Note that the graph is 3-connected and "highly" nonplanar. Also note that the facial cycles of the plane part of the graph all have length 7. Furthermore, the facial cycles of the graph are not chordless. If we add the dashed edge to the graph in Figure 8.2, the graph remains 5-almost planar, but now has a cycle of length 5 that is chordless and nonseparating, but not facial.

The first, and maybe most difficult step of the proof of the Definable Structure Theorem 8.1 is a proof of the following lemma.

LEMMA 8.7. *For all $p \in \mathbb{N}^+$ the class $\mathcal{AP}_p$ admits* IFP*-definable ordered treelike decompositions.*

As usual, we can restrict our attention to 3-connected graphs. To understand the idea of the proof, it may be best to first see why our two proofs that 3-connected planar graphs admit IFP-definable orders cannot be generalized to almost planar graphs. Example 8.6 shows that the proof given in Section 7 (for arbitrary surfaces) utterly fails, as it is based on the existence of short facial cycles and the fact that chordless and nonseparating cycles are facial. The following example shows that the proof given in Section 6 fails as well.

Example 8.8. Recall Example 6.12, referring to the 3-connected planar graph G shown in Figure 6.1. In that example, we computed the set W_∞ with respect to the angle $(5, 0, 10)$. We found that $V(G)\setminus W_\infty = \{0, 5, 6, 7, 8, 9, 10\}$ is the vertex set of the facial cycle determined by $(5, 0, 10)$. We also computed W_∞ with respect to the triple $(5, 0, 15)$, which is not angle, and found that $V(G)\setminus W_\infty$ is not the vertex set of a chordless and nonseparating cycle.

Now consider the 4-almost planar graph G' obtained from G by adding an edges between 25 and 30 and between 27 and 40 (Figure 8.3(a) shows G' without the vertex numbering). If we compute W_∞ with respect to the angle $(5, 0, 10)$ we find that $V(G)\setminus W_\infty = \{0, 5, 25, 30, 10\}$, which is the vertex set of a chordless and nonseparating cycle, but not the facial cycle determined by the angle $(5, 0, 10)$.

Let G'' be the 4-almost planar graph obtained from G by adding edges between 25 an 40 and between 27 and 40 (see Figure 8.3(b)). If we compute W_∞ with respect to the triple $(5, 0, 15)$, which is not an angle, we find that $V(G)\setminus W_\infty = \{0, 5, 25, 40, 15\}$, which again is the vertex set of a chordless and nonseparating cycle, but of course not a facial cycle.

It is not only the case that our previous proofs based on defining the angles (and hence the facial cycles) of the unique planar embedding of a 3-connected planar graph in IFP cannot be generalized to almost planar graphs. Even worse, the angles of a 3-connected almost planar graph cannot be defined at all, because different arrangements of the same graph in a disk may have different facial cycles. However, we can prove that facial cycles that are sufficiently far from the boundary of the disk are the same in all p-arrangements in a disk. "Sufficiently far" is not measured in terms of the graph metric. Instead, we say that a vertex v is *k-central* in some p-arrangement $(G_0, \pi, R, \bar{r})$ if $v \in \pi(V(G_0))$ and there are k disjoint concentric cycles around $\pi^{-1}(v)$ in the embedded graph G_0. We prove the following lemma.

LEMMA 8.9. *For all p there are fixed-point formulae $\mathsf{angle}^p(x_1, x_2, x_3)$ and $\mathsf{aligned}^p(x_1, x_2, x_3, x_4)$ with the following properties. Let G be a 3-connected graph and $(G_0, \pi, R, \bar{r})$ a p-arrangement of G in a disk. Then for all vertices $v_1, v_2, v_3, v_4 \in V(G)$ that are $(5p + 15)$-central,*

$$G \models \mathsf{angle}^p[v_1, v_2, v_3] \iff (v_1, v_2, v_3) \in \angle(G_0),$$
$$G \models \mathsf{aligned}^p[v_1, v_2, v_3, v_4] \iff (\pi^{-1}(v_1), \pi^{-1}(v_2), \pi^{-1}(v_3)) \curvearrowright (\pi^{-1}(v_2), \pi^{-1}(v_3), \pi^{-1}(v_4)).$$

The proof of the lemma is difficult. The basic idea is similar to the proof for planar graphs in Section 6.1, but instead of looking for just one path from v_1 to v_3 in $G \setminus \{v_2\}$, we look for $(p + 1)$ parallel paths. At most p of these parallel paths can go through the ring R, hence the "innermost" of the paths must go through G_0. This lets us recover the planar embedding, at least in sufficiently central regions.

Once we have the formulae angle^p and $\mathsf{aligned}^p$, we can define a linear order on all connected components of sufficiently central vertices. If we contract all these components to single vertices, we obtain a graph that we call the *skeleton* of the original graph. We prove that the skeleton has bounded tree width. Then we apply Corollary 3.21 to obtain an ordered treelike decomposition of the skeleton and a variant of the Ordered Extension Lemma 3.18 to add the linearly ordered central vertices to each bag, and this yields the desired ordered treelike decomposition.

8.3. Almost Embeddable Graphs

The next step is to prove the following lemma.

LEMMA 8.10. *For all $p, q, r, s \in \mathbb{N}^+$ the class $\mathcal{AE}_{p,q,r,s}$ of all (p, q, r, s)-embeddable graphs admits IFP-definable ordered treelike decompositions.*

For the proof, we first observe that by the Finite Extension Lemma (Corollary 3.19) we may assume $s = 0$ without loss of generality.

We prove that $\mathcal{AE}_{p,q,r,0}$ admits IFP-definable ordered treelike decompositions by induction. We have already dealt with the cases $q = 0$ (graphs embeddable in a surface of Euler genus r) and $r = 0, q = 1$ (almost planar graphs). Let $p, q, r \geq 0$ with $q + r \geq 2$, and suppose that for all $p', q', r' \geq 0$ such that either $r' < r$ or $r' = r$ and $q' < q$, the class $\mathcal{AE}_{p',q',r',0}$ admits definable ordered treelike decompositions. Let $G \in \mathcal{AE}_{p,q,r,0}$, and let $(G_0, \pi, R^1, \bar{r}^1, \ldots, R^q, \bar{r}^q)$ be a p-arrangement of G in a surface $\boldsymbol{S}$ of Euler genus r with q cuffs. The idea of the inductive step is very simple. We try to define a cycle $C \subseteq G_0$ such that $\Pi(C)$ is a noncontractible simple closed curve in $\boldsymbol{S}$ or a path $P \subseteq G_0$ that connects two of the rings, that is, goes from a vertex in $\bar{r}^i$ to a vertex in $\bar{r}^j$ for some $i \neq j$ (refer to Figure 8.4). If we delete C from G, we obtain a graph G' in $\mathcal{AE}_{p,q,r',0}$ for some $r' < r$, and if we delete P we obtain a graph G' in $\mathcal{AE}_{2p,q-1,r,0}$. In both cases we can define an ordered treelike decomposition of G' by the induction hypothesis and then apply the Ordered Extension Lemma 3.18.

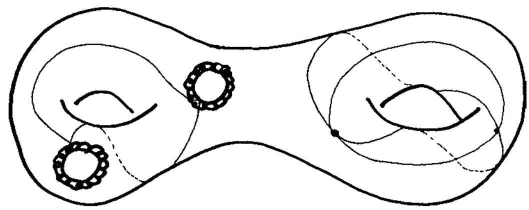


Fig. 8.4. Two paths connecting different rings (on the left) and two noncontractible cycles (on the right).

Unfortunately, there is no way to define a cycle C or path P as described in the last paragraph. Instead, we "guess" a few vertices of such a path or cycle that is shortest among all such path or cycles, and then consider all shortest paths between these vertices. We call the subgraphs we obtain this way *noncontractible belts*. These belts can be fairly complicated, and several things may happen, which lead to different applications of the induction hypothesis. The main case is that a belt is a planar graph. In this case, we delete the belt, apply the induction hypothesis to the resulting graph, and extend the decomposition to the original graph b the Ordered Extension Lemma 3.18, exploiting the fact that we can define linear orders on 3-connected planar graphs.

8.4. Proof of the Definable Structure Theorem 8.1

Let $\mathcal{C}$ be a class of graphs with excluded minors. If we had a d-scheme Λ such that for every graph $G \in \mathcal{C}$ the decomposition $\Lambda[G]$ would be a treelike decomposition over $\mathcal{AE}_{p,q,r,s}$ (for suitable parameters p, q, r, s depending on $\mathcal{C}$), then we could prove the Definable Structure Theorem by combining Lemma 8.10 with the Ordered Decomposition Lifting Lemma (Lemma 5.1.6 of Grohe [2012]), which states that if we have a definable treelike decomposition of a graph into torsos that admit definable ordered treelike decompositions, then we can "lift" the decompositions from the torsos to obtain an ordered treelike decomposition of the whole graph. Unfortunately, we do not have such a d-scheme Λ.

Instead, we need to prove a generalization of Lemma 8.10 stating that if we have some definable "pre-decomposition" Φ of a graph G, cutting off certain pieces of the graph, and the remaining part of the graph has an almost embeddable structure, then we can define an o-decomposition Δ of the graph that "completes" the pre-decomposition Φ in the sense that the pieces cut off by Φ appear at certain leaves of Δ, and the bags of all nodes that are not leaves corresponding to pieces cut off by Φ are linearly ordered by $\leqslant^{\Delta}$. This "local version" of Lemma 8.10 fits Robertson and Seymour's Local Structure Theorem (see the discussion following the Structure Theorem, Fact 8.5) very nicely. We can use the interplay between these two results to inductively define ordered treelike decomposition of the graphs in $\mathcal{C}$ by techniques similar to those developed in Robertson and Seymour [1991].

9. ISOMORPHISM TESTING AND THE WEISFEILER-LEHMAN ALGORITHM

It is a long-standing open problem whether there is a polynomial-time algorithm deciding if two graphs are isomorphic. Polynomial-time isomorphism tests are known for many natural classes of graphs, for example, planar graphs [Hopcroft and Tarjan 1972; Hopcroft and Wong 1974] and graphs embeddable in a fixed surface [Filotti and Mayer 1980; Miller 1980], graphs of bounded tree width [Bodlaender 1990], graphs with excluded minors [Ponomarenko 1988], and graphs of bounded degree [Luks 1982]. The isomorphism test for graphs of bounded degree due to Luks [1982] involves some nontrivial group theory, and many later isomorphism algorithms build on the group-theoretic techniques developed by Babai, Luks, and others (e.g., Babai [1981] and Babai and Luks [1983]) in the early 1980s. In particular, Ponomarenko's [1988] isomorphism algorithm for graphs with excluded minors builds on a generalization of Luks's bounded degree isomorphism test due to Miller [1983] and a canonization algorithm for graphs of bounded degree due to Babai and Luks [1983].

One of the simplest approaches to the graph isomorphism problem is the *color refinement* algorithm, which is also known as *vertex classification*. It computes a coloring of the vertices of a graph by the following iterative procedure. Initially, all vertices have the same color. Then in each round, the coloring is refined by assigning different colors to vertices that have a different number of neighbors of at least one color assigned in the previous round. Thus after the first round, two vertices have the same color if and only if they have the same degree. After the second round, two vertices have the same color if and only if they have the same degree and for each d the same number of neighbors of degree d. The algorithm stops if no further refinement is achieved; this happens after at most n rounds, where n is the number of vertices of the input graph. To use color refinement as an isomorphism test, it is applied to the disjoint union of two graphs. If after the refinement process the colorings of the two graphs differ, that is, for some color c the graphs have a different number of vertices of color c, then we know that they are nonisomorphic. Unfortunately, if both graphs have the same coloring, they may still be nonisomorphic. As an example, consider two nonisomorphic regular graphs with the same number of vertices, such as a cycle of length 6 and the disjoint union of two cycles of length 3. They will get the same coloring even though they are nonisomorphic. We say that color refinement *distinguishes* two graphs if they get different colorings. Babai, Erdös, and Selkow [1980] proved that color refinement distinguishes almost all nonisomorphic graphs. But as we have seen, there are very simple nonisomorphic graphs that color refinement does not distinguish.

The *k-dimensional Weisfeiler-Lehman algorithm* (for short: *k-WL*) is a straightforward generalization of the color refinement algorithm (see Cai et al. [1992] for a history of the algorithm). Instead of vertices, it colors k-tuples of vertices. Let G be the input graph. Initially, two tuples $(v_1, \dots, v_k), (w_1, \dots, w_k) \in V(G)^k$ get the same color if the mapping $v_i \mapsto w_i$ is an isomorphism from the induced subgraph $G[\{v_1, \dots, v_k\}]$ to the induced subgraph $G[\{w_1, \dots, w_k\}]$. For $i \leq k$, we say that a tuple $(v_1, \dots, v_k)$ is an *i-neighbor* of a tuple $(w_1, \dots, w_k)$ if $v_j = w_j$ for all $j \neq i$. In each round of the algorithm, the coloring is refined by assigning different colors to tuples that for some $i \in [k]$ and some color c have different numbers of i-neighbors of color c. The algorithm stops if no further refinement is achieved; this happens after at most n^k rounds. Again, isomorphic graphs obviously get the same coloring, and we say that k-WL *distinguishes* two nonisomorphic graphs if they get different colorings. It is not so obvious that there are nonisomorphic graphs that are not distinguished by k-WL for $k \geq 3$, let alone $k = \log n$. By a beautiful construction, which has found several other applications in finite model theory, Cai, Fürer, and Immerman [1992] proved that for each k there are nonisomorphic 3-regular graphs G_k, H_k with $O(k)$ vertices that cannot be distinguished by k-WL.

Our second main result states that a constant dimension suffices to distinguish graphs embeddable in a fixed surface.

THEOREM 9.1. *For every surface* $\mathbb{S}$ *there is a constant k such that k-WL distinguishes any two nonisomorphic graphs embeddable in* $\mathbb{C}$.

We actually prove a slightly stronger result. We say that k-WL *identifies* a graph G if k-WL distinguishes G from all graphs that are not isomorphic to G. We prove that for every surface $\boldsymbol{S}$ there is a constant k such that k-WL identifies every graph G embeddable in $\boldsymbol{S}$.

Otto [1997] proved that for every class $\mathcal{C}$ of graphs closed under disjoint unions, if IFP+C captures PTIME on $\mathcal{C}$ then there is a k such that k-WL distinguishes any two nonisomorphic graphs in $\mathcal{C}$. (It is an interesting open question whether the converse of this holds as well.) Hence Theorem 9.1 follows from Theorem 1.2. Theorem 9.1 also follows easily from the Definable Structure Theorem for Embeddable Graphs 7.1 via definable canonization.

The corresponding result for arbitrary classes with excluded minors follows from Theorem 1.3: *for every class $\mathcal{C}$ of graphs with excluded minors there is a constant k such that k-WL distinguishes any two nonisomorphic graphs in $\mathcal{C}$.* Thus the Weisfeiler-Lehman algorithm yields a simple combinatorial polynomial-time isomorphism test for every class $\mathcal{C}$ of graphs with excluded minors. This algorithm completely avoids the sophisticated group-theoretic machinery of Ponomarenko's algorithm.

Remarkably, Theorem 9.1 and its strengthening to arbitrary classes of graphs with excluded minors are not referring to logic in any way—the Weisfeiler-Lehman algorithm is a purely combinatorial algorithm—yet our proof of the theorem heavily relies on logic. The connection between the Weisfeiler-Lehman algorithm and definability in fixed-point logic with counting and also definability in finite variable infinitary logic with counting has first been observed in Cai et al. [1992]. It has been exploited for graph isomorphism testing in Grohe [2000], Grohe and Verbitsky [2006], Köbler and Verbitsky [2008], and Verbitsky [2007].

10. CONCLUDING REMARKS

In this article, we prove that for every surface $\boldsymbol{S}$, inflationary fixed-point logic with counting IFP+C captures PTIME on the class of all graphs embeddable in $\boldsymbol{S}$. We outline how this result can be extended to all classes of graphs with excluded minors. While the power of IFP+C in this context seems to be well understood, a few interesting classes of graphs remain where it is conceivable, but still open, whether IFP+C captures PTIME on these classes. The prime examples are classes of graphs of bounded rank width [Oum and Seymour 2006] (or equivalently bounded clique width [Courcelle and Olariu 2000]) and the class of unit disk intersection graphs.

To make real progress on the question of whether there is a logic for polynomial time, however, it will be necessary to look at more expressive logics. Choiceless polynomial-time CPT+C with counting [Blass et al. 1999, 2002] and fixed-point logic with a rank operator IFP+R [Dawar et al. 2009] are interesting candidates. Both extend IFP+C, but in completely different directions. The expressive power of both of these logics is not very well understood. It is still open whether one of these logics captures PTIME and how they compare in expressive power. It would also be interesting to find a natural logic, possibly CPT+C or IFP+R, that captures polynomial time on classes of graphs of bounded degree. (It follows from the fact that graphs of bounded degree admit polynomial-time canonization [Babai and Luks 1983] that there is a logic capturing PTIME on these classes, but this logic is very artificial.)

The connection between our results and graph isomorphism testing also raises some interesting open questions. It has been shown in Grohe and Verbitsky [2006] and Verbitsky [2007] that there is a k such that for any two nonisomorphic planar graphs the k-dimensional WL algorithm distinguishes the two graphs after only $O(\log n)$ rounds (where n can be taken as the sum of the sizes of the two graphs). This yields a simple parallel algorithm for testing isomorphism of planar graphs in polylogarithmic time (in AC^1, to be precise). It is open whether these results can be extended to all classes of graphs embeddable in a fixed surface or even all classes of graphs with excluded minors.

Finally, let me reiterate the question of whether all minor-closed classes of graphs, or at least the classes of graphs embeddable in a fixed surface other than the 2-sphere, are IFP-definable. Recall that we prove that all these classes are IFP+C-definable and that the class of planar graphs is IFP-definable.

ELECTRONIC APPENDIX

The electronic appendix to this article is available in the ACM Digital Library.

REFERENCES

BABAI, L. 1981. Moderately exponential bound for graph isomorphism. In *Fundamentals of Computation Theory (FCT'81)*, F. Gécseg, Ed., Lecture Notes in Computer Science, vol. 117, Springer, 34–50.

BABAI, L., ERDÖS, P., AND SELKOW, S. 1980. Random graph isomorphism. *SIAM J. Comput. 9*, 628–635.

BABAI, L. AND LUKS, E. 1983. Canonical labeling of graphs. In *Proceedings of the 15th ACM Symposium on Theory of Computing*. 171–183.

BLASS, A., GUREVICH, Y., AND SHELAH, S. 1999. Choiceless polynomial time. *Ann. Pure Appl. Logic 100*, 141–187.

BLASS, A., GUREVICH, Y., AND SHELAH, S. 2002. On polynomial time computation over unordered structures. *J. Symbol. Logic 67*, 1093–1125.

BODLAENDER, H. 1990. Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees. *J. Algor. 11*, 631–643.

CAI, J., FÜRER, M., AND IMMERMAN, N. 1992. An optimal lower bound on the number of variables for graph identification. *Combinatorica 12*, 389–410.

CHANDRA, A. AND HAREL, D. 1982. Structure and complexity of relational queries. *J. Comput. Syst. Sci. 25*, 99–128.

CONWAY, J. AND GORDON, C. 1983. Knots and links in spatial graphs. *J. Graph Theory 7*, 445–453.

COURCELLE, B. AND OLARIU, S. 2000. Upper bounds to the clique-width of graphs. *Discr. Appl. Math. 101*, 77–114.

DAWAR, A., GROHE, M., HOLM, B., AND LAUBNER, B. 2009. Logics with rank operators. In *Proceedings of the 24th IEEE Symposium on Logic in Computer Science*. 113–122.

DEMAINE, E., HAJIAGHAYI, M., AND KAWARABAYASHI, K. 2005. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*. 637–646.

DIESTEL, R. 2005. *Graph Theory* 3rd Ed. Springer.

EBBINGHAUS, H.-D. AND FLUM, J. 1999. *Finite Model Theory,* 2nd Ed. Springer.

EBBINGHAUS, H.-D., FLUM, J., AND THOMAS, W. 1994. *Mathematical Logic* 2nd Ed. Springer.

FAGIN, R. 1974. Generalized first–order spectra and polynomial–time recognizable sets. In *SIAM-AMS Proceedings on Complexity of Computation*. Vol. 7, R. Karp, Ed., 43–73.

FILOTTI, I. S. AND MAYER, J. N. 1980. A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus. In *Proceedings of the 12th ACM Symposium on Theory of Computing*. 236–243.

FLUM, J. AND GROHE, M. 2000. On fixed-point logic with counting. *J. Symbol. Logic 65*, 777–787.

GRÄDEL, E., KOLAITIS, P., LIBKIN, L., MARX, M., SPENCER, J., VARDI, M., VENEMA, Y., AND WEINSTEIN, S. 2007. *Finite Model Theory and Its Applications*. Springer.

GRÄDEL, E. AND OTTO, M. 1993. Inductive definability with counting on finite structures. In *Proceedings of the Computer Science Logic, 6th Workshop (CSL'92)*. Selected Papers, E. Börger, G. Jäger, H. K. Büning, S. Martini, and M. Richter, Eds., Lecture Notes in Computer Science, vol. 702, Springer, 231–247.

GROHE, M. 1998. Fixed-Point logics on planar graphs. In *Proceedings of the 13th IEEE Symposium on Logic in Computer Science*. 6–15.

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

GROHE, M. 2008. Definable tree decompositions. In *Proceedings of the 23rd IEEE Symposium on Logic in Computer Science*. 406–417.

GROHE, M. 2010. Fixed-Point definability and polynomial time on graphs with excluded minors. In *Proceedings of the 25th IEEE Symposium on Logic in Computer Science*.

GROHE, M. 2011. From polynomial time queries to graph structure theory. *Comm. ACM 54,* 6, 104–112.

GROHE, M. 2012. Descriptive complexity, canonisation, and definable graph structure theory. `http://www2.informatik.hu-berlin.de/~grohe/cap`.

GROHE, M. AND MARIÑO, J. 1999. Definability and descriptive complexity on databases of bounded tree-width. In *Proceedings of the 7th International Conference on Database Theory*. C. Beeri and P. Buneman, Eds., Lecture Notes in Computer Science, vol. 1540, Springer, 70–82.

GROHE, M. AND VERBITSKY, O. 2006. Testing graph isomorphism in parallel by playing a game. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming,* Part I, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., Lecture Notes in Computer Science, vol. 4051, Springer, 3–14.

GROSS, J. AND TUCKER, T. 1987. *Topological Graph Theory*. Wiley.

GUREVICH, Y. 1984. Toward logic tailored for computational complexity. In *Computation and Proof Theory*, M. M. Richter, E. Börger, W. Oberschelp, B. Schinzel, and W. Thomas, Eds., Lecture Notes in Mathematics, vol. 1104, Springer, 175–216.

GUREVICH, Y. 1988. Logic and the challenge of computer science. In *Current Trends in Theoretical Computer Science*, E. Börger, Ed., Computer Science Press, 1–57.

GUREVICH, Y. AND SHELAH, S. 1986. Fixed point extensions of first–order logic. *Ann. Pure Appl. Logic 32*, 265–280.

HELLA, L., KOLAITIS, P., AND LUOSTO, K. 1996. Almost everywhere equivalence of logics in finite model theory. *Bull. Symbol. Logic 2*, 422–443.

HOPCROFT, J. AND TARJAN, R. 1972. Isomorphism of planar graphs (working paper). In *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds., Plenum Press.

HOPCROFT, J. AND WONG, J. 1974. Linear time algorithm for isomorphism of planar graphs. In *Proceedings of the 6th ACM Symposium on Theory of Computing*. 172–184.

IMMERMAN, N. 1982. Relational queries computable in polynomial time (extended abstract). In *Proceedings of the 14th ACM Symposium on Theory of Computing*. 147–152.

IMMERMAN, N. 1986. Relational queries computable in polynomial time. *Inf. Control 68*, 86–104.

IMMERMAN, N. 1987. Expressibility as a complexity measure: Results and directions. In *Proceedings of the 2nd IEEE Symposium on Structure in Complexity Theory*. 194–202.

IMMERMAN, N. 1999. *Descriptive Complexity*. Springer.

IMMERMAN, N. AND LANDER, E. 1990. Describing graphs: A first-order approach to graph canonization. In *Complexity Theory Retrospective*, A. Selman, Ed., Springer, 59–81.

KÖBLER, J. AND VERBITSKY, O. 2008. From invariants to canonization in parallel. In *Proceedings of the 3rd International Computer Science Symposium*. E. Hirsch, A. Razborov, A. Semenov, and A. Slissenko, Eds., Lecture Notes in Computer Science, vol. 5010, Springer, 216–227.

KREUTZER, S. 2002. Expressive equivalence of least and inflationary fixed-point logic. In *Proceedings of the 17th IEEE Symposium on Logic in Computer Science*. 403–413.

LAUBNER, B. 2010. Capturing polynomial time on interval graphs. In *Proceedings of the 25th IEEE Symposium on Logic in Computer Science*. 199–208.

LEVINE, H. 1963. Homotopic curves on surfaces. *Proc. Amer. Math. Soc. 14*, 986–990.

LIBKIN, L. 2004. *Elements of Finite Model Theory*. Springer.

LUKS, E. 1982. Isomorphism of graphs of bounded valance can be tested in polynomial time. *J. Comput. Syst. Sci. 25*, 42–65.

MILLER, G. 1983. Isomorphism of graphs which are pairwise *k*-separable. *Inf. Control 56*, 21–33.

MILLER, G. L. 1980. Isomorphism testing for graphs of bounded genus. In *Proceedings of the 12th ACM Symposium on Theory of Computing*. 225–235.

MOHAR, B. 1999. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM J. Discr. Math. 12*, 6–26.

MOHAR, B. AND THOMASSEN, C. 2001. *Graphs on Surfaces*. Johns Hopkins University Press.

OTTO, M. 1997. *Bounded Variable Logics and Counting—A Study in Finite Models*. Lecture Notes in Logic, vol. 9, Springer.

OUM, S.-I. AND SEYMOUR, P. 2006. Approximating clique-width and branch-width. *J. Combin. Theory, Series B 96*, 514–528.

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

RINGEL, G. 1974. *Map Color Theorem*. Springer.

ROBERTSON, N. AND SEYMOUR, P. 1991. Graph minors X. Obstructions to tree-decomposition. *J. Combin. Theory, Series B 52*, 153–190.

ROBERTSON, N. AND SEYMOUR, P. 1995. Graph minors XIII. The disjoint paths problem. *J. Combin. Theory, Series B 63*, 65–110.

ROBERTSON, N. AND SEYMOUR, P. 1999. Graph minors XVI. Excluding a non-planar graph. *J. Combin. Theory, Series B 77*, 1–27.

ROBERTSON, N. AND SEYMOUR, P. 2004. Graph minors XX. Wagner's conjecture. *J. Combin. Theory, Series B 92*, 325–357.

ROBERTSON, N. AND VITRAY, R. 1990. Representativity of surface embeddings. In *Paths, Flows and VLSI-Layout*, B. Korte, L. Lovász, H. Prömel, and A. Schrijver, Eds., Springer, 293–328.

THOMASSEN, C. 1988. The graph genus problem is NP-complete. *J. Algor. 10*, 458–576.

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

VERBITSKY, O. 2007. Planar graphs: Logical complexity and parallel isomorphism tests. In *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science*. W. Thomas and P. Weil, Eds., Lecture Notes in Computer Science, vol. 4393, Springer, 682–693.

WHITNEY, H. 1932. Congruent graphs and the connectivity of graphs. *Amer. J. Math. 54*, 150–168.

Received May 2011; revised July 2012; accepted July 2012