A Linear Upper Bound on the Weisfeiler-Leman Dimension of Graphs of Bounded Genus

Martin Grohe
RWTH Aachen University, Aachen, Germany
grohe@cs.rwth-aachen.de

Sandra Kiefer
RWTH Aachen University, Aachen, Germany
kiefer@cs.rwth-aachen.de

Abstract

The Weisfeiler-Leman (WL) dimension of a graph is a measure for the inherent descriptive complexity of the graph. While originally derived from a combinatorial graph isomorphism test called the Weisfeiler-Leman algorithm, the WL dimension can also be characterised in terms of the number of variables that is required to describe the graph up to isomorphism in first-order logic with counting quantifiers.

It is known that the WL dimension is upper-bounded for all graphs that exclude some fixed graph as a minor [17]. However, the bounds that can be derived from this general result are astronomic. Only recently, it was proved that the WL dimension of planar graphs is at most 3 [26].

In this paper, we prove that the WL dimension of graphs embeddable in a surface of Euler genus g is at most $4g + 3$. For the WL dimension of graphs embeddable in an orientable surface of Euler genus g, our approach yields an upper bound of $2g + 3$.

2012 ACM Subject Classification Theory of computation → Finite Model Theory; Mathematics of computing → Graph theory

Keywords and phrases Weisfeiler-Leman algorithm, finite-variable logic, isomorphism testing, planar graphs, bounded genus

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.117

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version https://arxiv.org/abs/1904.07216

1 Introduction

The Weisfeiler-Leman (WL) algorithm is a simple combinatorial graph isomorphism test. The 1-dimensional version of the algorithm, also known as colour refinement and naive vertex classification, is known since at least the mid 1960s, and it is used as a subroutine in almost all practical graph isomorphism tools (see, for instance, [9, 25, 34, 35]), but also in machine learning (see, for instance, [1, 22, 29, 37, 40]). The 2-dimensional version can be traced back to an article by Weisfeiler and Leman that appeared 50 years ago [41]. It is closely related to the algebraic theory of coherent configurations. The generalisation to higher dimensions is due to Babai (see [6, 8]), and again it plays an important role as a subroutine in graph isomorphism algorithms, albeit more on the theoretical side. Notably, Babai uses the $(\log n)$-dimensional version in his quasipolynomial isomorphism test [6].

The connection between the WL algorithm and logic was made by Immerman and Lander [24] and Cai, Fürer, and Immerman [8]. They showed that two graphs are distinguished by the k-dimensional WL algorithm if and only if they can be distinguished in the logic C^{k+1}, the $(k+1)$-variable fragment of first-order logic using counting quantifiers of the form $\exists^{\geq m}x$. The connection between the WL algorithm and logical definability is at the core of some of the most interesting developments in descriptive complexity theory (see, for

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 117; pp. 117:1–117:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

example, [17, 23, 39]). Only recently, it has been noted that the WL algorithm (and thus the finite-variable counting logic) has further surprising characterisations. In a breakthrough paper, Atserias and Maneva [4] showed that the dimension k of the WL algorithm required to distinguish two graphs corresponds to the level of the Sherali-Adams relaxation of the natural integer linear program for graph isomorphism testing (also see [21, 33]). This spawned a lot of work relating the WL algorithm to semidefinite programming [5, 38] and algebraic (Gröbner basis) approaches [7, 13] to graph isomorphism testing. These results can also be phrased in terms of propositional proof complexity. The latest facet of the theory is a characterisation in terms of homomorphism counts of graphs of tree width k [10]. Various aspects of the WL algorithm and its relation to logic have been studied in detail in recent years (see, for instance, [2, 3, 12, 27, 28, 31]).

Cai, Fürer, and Immerman [8] proved that for every k there are non-isomorphic (3-regular) graphs G_k, H_k of size $O(k)$ that are not distinguished by the k-dimensional WL algorithm. Thus, as an isomorphism test, the k-dimensional WL algorithm is incomplete. But, in view of the wide variety of seemingly unrelated combinatorial, logical, and algebraic characterisations of the algorithm, *we are convinced that the structural information the algorithm does detect is of fundamental importance.* The basic parameter of the algorithm is the dimension, corresponding to the number of variables in logical and the degree of polynomials in algebraic characterisations. It yields a structural invariant called the *WL dimension* of a graph G [17], defined to be the least k such that the k-dimensional WL algorithm distinguishes G from every graph H not isomorphic to G (we say that k-WL *identifies* G), or equivalently, the least k such that G can be characterised up to isomorphism (or *identified*) in the logic C^{k+1}. It is also convenient to define the *WL dimension* of a class $\mathcal{C}$ of graphs to be the maximum of the WL dimensions of all graphs in $\mathcal{C}$ if this maximum exists, and ∞ otherwise. We see the WL dimension as a measure for the inherent combinatorial or descriptive complexity of a graph or a class of graphs. We are mostly interested in the relation between the WL dimension and other graph invariants.

Work in descriptive complexity shows that the WL dimension is bounded for many natural graph classes, among them trees [24], graphs of bounded tree width [19], planar graphs [14], graphs of bounded genus [15, 16], all graph classes that exclude some fixed graph as a minor [17], interval graphs [30, 32], and graphs of bounded rank width [20]. However, most of these results do not give explicit bounds on the WL dimension, and the bounds that can be derived from the proofs are usually bad. Only recently, the second author of this paper, jointly with Ponomarenko and Schweitzer, established an almost tight bound for planar graphs [26]: the WL dimension of planar graphs is at most 3, and there are planar graphs of WL dimension 2.

In this paper, we establish bounds for graphs that can be embedded into an arbitrary surface, for example, a torus or a projective plane. By the classification theorem for surfaces (see [36, Theorem 3.1.3]), up to homeomorphism (that is, topological equivalence), all surfaces fall into only two countably infinite families, the family $(\boldsymbol{S}_k)_{k\geq 0}$ of orientable surfaces and the family $(\boldsymbol{N}_\ell)_{\ell\geq 1}$ of non-orientable surfaces. For example, the sphere $\boldsymbol{S}_0$, the torus $\boldsymbol{S}_1$, and the double torus $\boldsymbol{S}_2$ are the first three orientable surfaces, and the projective plane $\boldsymbol{N}_1$ and the Klein bottle $\boldsymbol{N}_2$ are the first two non-orientable surfaces. The *Euler genus* $\mathrm{eg}(\boldsymbol{S})$ of a surface $\boldsymbol{S}$ is $2k$ if $\boldsymbol{S}$ is homeomorphic to the orientable surface $\boldsymbol{S}_k$, and ℓ if $\boldsymbol{S}$ is homeomorphic to the non-orientable surface $\boldsymbol{N}_\ell$. We define the *Euler genus* of a graph G to be the least g such that G is embeddable (that is, can be drawn without edge crossings) in a surface of Euler genus g. Ssee Figure 1 for an example.

▶ **Theorem 1.** *The WL dimension of the class of graphs of Euler genus g is at most $4g + 3$.*

For graphs embeddable in orientable surfaces, we can improve the bound further.

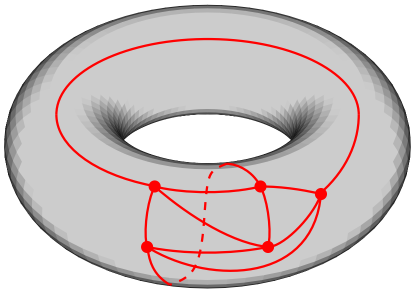

Figure 1 Embedding of K_5 into the torus.

▶ **Corollary 2.** *The WL dimension of the class of graphs embeddable in an orientable surface of Euler genus g is at most $2g + 3$.*

As mentioned above, it was first proved in [15] that the WL dimension of graphs of bounded genus is bounded. A more detailed proof of the same result can be found in the journal paper [16]. The proof of [16] only yields an asymptotically quadratic bound (in terms of the genus) on the dimension, but neither of the two papers gives an explicit bound. It seems that the proof of [15] gives a linear bound, albeit with a large constant factor of at least 80 (not all details are worked out there, so it is difficult to determine the exact bound). The proof in both of these papers is based on the fact that sufficiently large graphs of minimum degree at least 3, embedded in a surface, will have a facial cycle of length at most 6. The proof we give here is completely different. It is based on the straightforward idea of removing a non-contractible cycle to reduce the Euler genus and then using induction. The problem with this idea is that we cannot define non-contractible cycles, but rather only families of such cycles that may intersect in complicated patterns. Understanding these families leads to significant technical complications, but in the end enables us to obtain a much better bound than the simpler proofs of [15, 16]. Our proof is based on a simplified version of a construction from [17, Chapter 15], applied there to graphs "almost embeddable" in a surface.

Outline of the Paper

In Section 2, we present the conventions as well as some topological notions and facts that we use throughout the paper. In Section 3, we introduce the WL dimension and relate it to logic. In Section 4, we present the graph-theoretic machinery that we need in the proof of our main theorem. The proof is outlined in Section 5. The detailed proof is long and complicated and can be found in [18], which also contains further material with respect to all other sections.

2 Preliminaries

2.1 Graphs

We use a standard graph terminology and notation. The only slightly unusual object is our version of coloured graphs. In an *arc-coloured* graph, we colour both vertices and orientations of edges. Formally, an arc-coloured graph is a graph G together with a function $\chi\colon \{(u,u) \mid u \in V(G)\} \cup \{(u,v) \mid \{u,v\} \in E(G)\} \to \mathcal{C}$, where $\mathcal{C}$ is some set of colours. We interpret $\chi(u,u)$ as the colour of the vertex u. Whenever we refer to coloured graphs in this paper, we mean arc-coloured graphs.

2.2 Topology

We have already discussed surfaces and their Euler genus in the introduction. In our presentation and notation, we follow [17, Chapter 9]. As an important notational convention, we always use bold face letters to denote topological spaces. Many more details on surface topology can be found in [17, 36], and in [11, Appendix B].

For a topological space $\boldsymbol{X}$ and a subset $\boldsymbol{Y} \subseteq \boldsymbol{X}$, we define the *boundary* of $\boldsymbol{Y}$ in $\boldsymbol{X}$ to be the set $\boldsymbol{bd_X}(\boldsymbol{Y})$ of all points $x \in \boldsymbol{X}$ such that every neighbourhood of x has a nonempty intersection with both $\boldsymbol{Y}$ and $\boldsymbol{X} \setminus \boldsymbol{Y}$. We omit the subscript $\boldsymbol{X}$ if the space, usually a surface, is clear from the context.

A *closed disk* is a homeomorphic image of $\{x \in \mathbb{R}^2 \mid \|x\| \leq 1\}$ equipped with the usual topology, and an *open disk* is a subspace of $\mathbb{R}^2$ homeomorphic to $\mathbb{R}^2$ (viewed as a topological space). Let $\boldsymbol{g}$ be a simple closed curve in a surface $\boldsymbol{S}$. Then $\boldsymbol{g}$ is *contractible* if it is the boundary of a closed disk in $\boldsymbol{S}$, otherwise g is *non-contractible*. If $\boldsymbol{g}$ is non-contractible, we can obtain one or two surfaces of strictly smaller Euler genus by the following construction: we cut the surface along $\boldsymbol{g}$; what remains is a surface with one or two holes in it. Then we glue a disk onto each hole and obtain one or two simpler surfaces.

It will be important for us to distinguish between graphs in their standard combinatorial form – we refer to them as *abstract graphs* – and *embedded graphs*. The vertices of a graph *embedded* in a surface $\boldsymbol{S}$ are points in $\boldsymbol{S}$, and the edges are simple curves connecting the vertices in such a way that they do not cross. If G is a graph embedded in $\boldsymbol{S}$, we denote by $\boldsymbol{G}$ the subset of $\boldsymbol{S}$ consisting of all points that are either vertices of G or contained in an edge of G. The *faces* of G are the arcwise connected components of $\boldsymbol{S} \setminus \boldsymbol{G}$.

We say that an abstract graph G is *embeddable* into a surface $\boldsymbol{S}$ if it is isomorphic to (the underlying graph of) a graph embedded in $\boldsymbol{S}$. The *Euler genus* $\mathrm{eg}(G)$ of a graph G is the smallest g such that G is embeddable into a surface of Euler genus g. The graphs of Euler genus 0 are precisely the *planar graphs*, because a graph is embeddable into the 2-sphere $\boldsymbol{S}_0$ if and only if it is embeddable into the plane $\boldsymbol{R}^2$. The class of all graphs of Euler genus at most g is denoted by $\mathcal{E}_g$.

A graph G is *polyhedrally embedded* in a surface $\boldsymbol{S}$ if G is embedded in $\boldsymbol{S}$, 3-connected, and every non-contractible simple closed curve $\boldsymbol{g} \subseteq \boldsymbol{S}$ intersects $\boldsymbol{G}$ in at least three points. Just like 3-connected graphs embedded in a plane, polyhedrally embedded graphs have many nice properties that we will exploit here.

2.3 Logic

C is the extension of first-order logic FO by *counting quantifiers* $\exists^{\geq m}x$ with the obvious meaning. C is only a syntactical extension of FO, because $\exists^{\geq m}x\varphi(x)$ is equivalent to $\exists x_1 \dots \exists x_m\Big(\bigwedge_{i \neq j} x_i \neq x_j \wedge \bigwedge_i \varphi(x_i)\Big)$. However, we are mainly interested in the fragments C^k of C consisting of all formulae with at most k variables. If $m > k$, then $\exists^{\geq m}x$ cannot be expressed in the k-variable fragment of FO, this is why we add the counting quantifiers. The logics C^k have played an important role in finite-model theory since the 1980s.

We often write $\varphi(x_1, \dots, x_\ell)$ to indicate that the free variables of φ are among $x_1, \dots, x_\ell$. (Not all of these variables are required to appear in φ.) Then for a graph G and vertices $u_1, \dots, u_\ell \in V(G)$, we write $G \models \varphi(u_1, \dots, u_\ell)$ to denote that G satisfies φ if for all i the variable x_i is interpreted by u_i. Moreover, we write $\varphi[G, u_1, \dots, u_i, x_{i+1}, \dots, x_\ell]$ to denote the set of all $(\ell - i)$-tuples $(u_{i+1}, \dots, u_\ell)$ such that $G \models \varphi(u_1, \dots, u_\ell)$.

The *width* of a formula $\varphi \in \mathsf{C}$ is the maximum number of free variables of a subformula of φ. Clearly, every formula in C^k has width at most k. An important observation that we often use is that every C-formula of width at most k is equivalent to a C^k-formula. We denote the set of all C-formulae of width at most k by $\mathsf{C}^k_{\mathsf{w}}$.

3 The WL Dimension

We start by reviewing the *k-dimensional WL algorithm* (for short: *k-WL*) for $k \geq 1$.

The *atomic type* $\operatorname{atp}(G, \bar{u})$ of a k-tuple $\bar{u} = (u_1, \ldots, u_k)$ of vertices of a (possibly coloured) graph G is the set of all atomic formulae satisfied by these vertices. The exact encoding is not important for us, the relevant property is that tuples $\bar{u} = (u_1, \ldots, u_k)$ and $\bar{v} = (v_1, \ldots, v_k)$ of vertices of graphs G and H, respectively, have the same atomic type if and only if the mapping $u_i \mapsto v_i$ is an isomorphism from the induced subgraph $G[\{u_1, \ldots, u_k\}]$ to the induced subgraph $H[\{v_1, \ldots, v_k\}]$.

Now k-WL is the algorithm that, given a graph G, computes the following sequence of "colourings" C^k_i of $\big(V(G)\big)^k$ for $i \geq 0$ until it returns $C^k_\infty = C^k_i$ for the smallest i such that for all $\bar{u}, \bar{v}$ it holds that $C^k_i(\bar{u}) = C^k_i(\bar{v}) \iff C^k_{i+1}(\bar{u}) = C^k_{i+1}(\bar{v})$. The initial colouring C^k_0 assigns to each tuple its atomic type: $C^k_0(\bar{u}) := \operatorname{atp}(G, \bar{u})$. In the $(i+1)$-st *refinement round*, the colouring C^k_{i+1} is defined by $C^k_{i+1}(\bar{u}) := \big(C^k_i(\bar{u}), M_i(\bar{u})\big)$, where, for $\bar{u} = (u_1, \ldots, u_k)$, $M_i(\bar{u})$ is the multiset

$$\Big\{\!\!\Big\{ \Big(\operatorname{atp}\big(G, (u_1, \ldots, u_k, v)\big), C^k_i(u_1, \ldots, u_{k-1}, v), \\ C^k_i(u_1, \ldots, u_{k-2}, v, u_k), \ldots, C^k_i(v, u_2, \ldots, u_k)\Big) \;\Big|\; v \in V \Big\}\!\!\Big\}$$

We say that k-WL *distinguishes* two graphs G and H if there is some colour c in the range of C^k_∞ such that the number of tuples $\bar{u} \in \big(V(G)\big)^k$ with $C^k_\infty(\bar{u}) = c$ is different from the number of tuples $\bar{v} \in \big(V(H)\big)^k$ with $C^k_\infty(\bar{v}) = c$. We say that k-WL *identifies* G if it distinguishes G from all graphs H not isomorphic to G. The *WL dimension* of G is the smallest k such that k-WL identifies G.

In this paper, we reason about the WL dimension in terms of logic, using the following theorem.

▶ **Theorem 3** ([8, 24]). *Let $k \geq 1$. Let G and H be graphs, possibly coloured, and $\bar{u} = (u_1, \ldots, u_k) \in \big(V(G)\big)^k$ and $\bar{v} = (v_1, \ldots, v_k) \in \big(V(H)\big)^k$. Then the following are equivalent:*
1. $C^k_\infty(\bar{u}) = C^k_\infty(\bar{v})$*;*
2. $G \models \varphi(u_1, \ldots, u_k) \iff H \models \varphi(v_1, \ldots, v_k)$ *for all C^{k+1}-formulae $\varphi(x_1, \ldots, x_k)$.*

We say that a graph G is *identified* by the logic C^k if there is a sentence $\mathsf{iso}_G \in \mathsf{C}^k$ such that for all graphs H we have $H \models \mathsf{iso}_G$ if and only if H is isomorphic to G.

▶ **Corollary 4.** *A graph has WL dimension k if and only if it is identified by C^{k+1}.*

The WL dimension of the class of planar graphs is at most 3 [26]. Using the previous corollary, we can re-phrase this as follows.

▶ **Theorem 5** ([26]). *For every planar graph G there is a C^4-sentence iso_G that identifies G.*

4 Shortest Path Systems, Patches and Necklaces

Here we introduce the graph-theoretic machinery necessary to prove our main theorem. Essentially, the definitions and results of this section are from [17, Chapter 15]. In fact, things are simpler here because [17, Chapter 15] deals with graphs *almost* embedded in a surface, whereas we only need to consider embedded graphs. Sometimes, we need to change the definitions in order to improve the resulting bounds on the WL dimension later. Notably, our *necklaces* play the role of the *belts* in [17], but the definition is slightly different. This also requires an adaptation of the proof that reducing necklaces exist.

▶ **Definition 6.** *Let G be a graph and $u, u' \in V(G)$. A* shortest path system (sps) from u to u' *is a family $\mathcal{Q}$ of shortest paths in G from u to u' such that every shortest path from u to u' in the subgraph $\bigcup_{Q \in \mathcal{Q}} Q$ is contained in $\mathcal{Q}$.*

We let $V(\mathcal{Q}) := \bigcup_{Q \in \mathcal{Q}} V(Q)$ and $E(\mathcal{Q}) := \bigcup_{Q \in \mathcal{Q}} E(Q)$ and $G(\mathcal{Q}) := \big(V(\mathcal{Q}), E(\mathcal{Q})\big) = \bigcup_{Q \in \mathcal{Q}} Q$. We call $\mathcal{Q}$ trivial *if $|V(\mathcal{Q})| \leq 2$, that is, if $G(\mathcal{Q})$ consists of a single vertex or a single edge.*

The height $\mathrm{ht}^{\mathcal{Q}}(v)$ *of $v \in V(\mathcal{Q})$ is the distance from u to v. The vertices in $\bigcap_{Q \in \mathcal{Q}} V(Q)$ are the* articulation vertices *of $\mathcal{Q}$. An articulation vertex v is* proper *if $v \neq u$ and $v \neq u'$. We denote the set of all articulation vertices of $\mathcal{Q}$ by* $\mathrm{art}(\mathcal{Q})$.

For all u, $u' \in V(G)$ such that there is a path from u to u' in G, the canonical sps from u to u' in G *is the set $\mathcal{Q}^G(u, u')$ of all shortest paths from u to u' in G.*

While shortest paths systems are defined with respect to abstract graphs, the following notions are defined with respect to embedded graphs. For the rest of the section, we make the following assumption.

▶ **Assumption 7.** *G is a graph polyhedrally embedded in a surface $\boldsymbol{S}$ of Euler genus $g \geq 1$.*

▶ **Definition 8.** *A* patch *in G is an sps $\mathcal{Q}$ in G such that:*
(i) *$\mathcal{Q}$ has no proper articulation vertices.*
(ii) *There is a closed disk $\boldsymbol{D} \subseteq \boldsymbol{S}$ such that $\boldsymbol{G(\mathcal{Q})} \subseteq \boldsymbol{D}$.* ◢

It can be shown that if $\mathcal{Q}$ is a non-trivial patch (i.e., a patch that does not consist of just a single vertex or a single edge), then there is a unique disk $\boldsymbol{D}(\mathcal{Q})$ such that $\boldsymbol{G(\mathcal{Q})} \subseteq \boldsymbol{D}(\mathcal{Q})$ and there is a cycle $C(\mathcal{Q}) \subseteq G(\mathcal{Q})$ such that $\boldsymbol{bd}\big(\boldsymbol{D}(\mathcal{Q})\big) = \boldsymbol{C}(\mathcal{Q})$. Furthermore, there are two paths $Q, Q' \in \mathcal{Q}$ such that $C(\mathcal{Q}) = Q \cup Q'$.

We call a subgraph $H \subseteq G$ *simplifying* if every connected component of $G \setminus H$ is contained in $\mathcal{E}_{g-1}$. Otherwise, H is *non-simplifying.*

▶ **Lemma 9** ([17], Corollary 15.3.5)**.** *For every non-simplifying subgraph $H \subseteq G$, there is exactly one connected component A^* of $G \setminus H$ with $A^* \notin \mathcal{E}_{g-1}$, and all other connected components are planar.*

A patch $\mathcal{Q}$ is *simplifying* if the graph $G(\mathcal{Q})$ is. It turns out that non-simplifying patches form the basic building blocks of our theory. Let $\mathcal{Q}$ be a non-trivial non-simplifying patch. Let A^* be the unique connected component of $G \setminus V(\mathcal{Q})$ that is not planar (the existence and uniqueness of A^* follows from Lemma 9). Then we define G/A^* to be the graph obtained from G by contracting the subgraph A^* to a single vertex a^*. By [17, Corollary 15.4.5], G/A^* is a 3-connected planar graph. Figure 2 displays a schematic view of a patch $\mathcal{Q}$ with some attached (planar) connected components as well as the non-planar component A^*, the disk $\boldsymbol{D}(\mathcal{Q})$, and the boundary cycle $\boldsymbol{C}(\mathcal{Q})$.

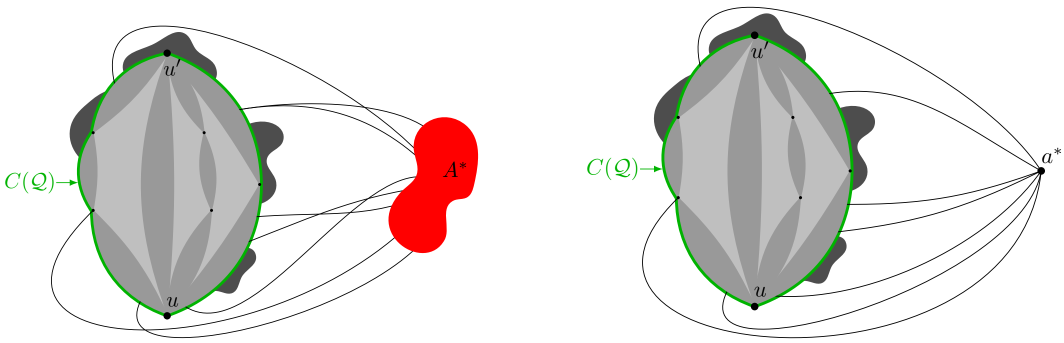

Figure 2 Left: A patch $\mathcal{Q}$ with non-planar component A^* and boundary cycle $C(\mathcal{Q})$. The curve $\boldsymbol{C}(\mathcal{Q})$ is the boundary of the disk $\boldsymbol{D}(\mathcal{Q})$, which consists of the light gray and medium gray areas. Right: the (planar) factor graph G/A^*.

We define the *internal graph* of a non-trivial non-simplifying patch $\mathcal{Q}$ to be the graph $I := I(\mathcal{Q})$ with vertex set $V(I) := V(G) \cap \boldsymbol{D}(\mathcal{Q})$ and edge set $E(I) := \{\boldsymbol{e} \in E(G) \mid \boldsymbol{e} \subseteq \boldsymbol{D}(\mathcal{Q})\}$. Note that $C(\mathcal{Q}) \subseteq I$. The definitions of the graphs $C(\mathcal{Q})$ and $I(\mathcal{Q})$ do not only depend on the abstract graph G and the sps $\mathcal{Q}$, but on the embedding of G in $\boldsymbol{S}$. However, it can be proved that actually the graphs are invariant under embeddings.

▶ **Lemma 10** ([17]). *Let $\mathcal{Q}$ be a non-trivial non-simplifying patch in G. Let G' be a graph embedded in a surface $\boldsymbol{S}'$ of Euler genus g such that G and G' are isomorphic (as abstract graphs), and let f be an isomorphism from G to G'. Then $\mathcal{Q}' := f(\mathcal{Q})$ is a non-simplifying patch in G', and it holds that $f\big(C(\mathcal{Q})\big) = C(\mathcal{Q}')$ and $f\big(I(\mathcal{Q})\big) = I(\mathcal{Q}')$.*

This follows from [17, Lemma 15.4.10]. Intuitively, the reason why this holds is that the 3-connected planar graph G/A^* has a "unique" embedding.

▶ **Corollary 11.** *Let $u, u' \in V(G)$ and $\mathcal{Q} := \mathcal{Q}^G(u, u')$ such that $\mathcal{Q}$ is a non-trivial non-simplifying patch. Let f be an automorphism of G such that $f(u) = u$ and $f(u') = u'$. Then $f\big(C(\mathcal{Q})\big) = C(\mathcal{Q})$ and $f\big(I(\mathcal{Q})\big) = I(\mathcal{Q})$.*

We remark that the analogue of Corollary 11 for simplifying patches does not hold (see [18, Figure 4]). The analysis of simplifying patches is much more involved, and we defer the reader to [18].

The final objects we define in this section are *necklaces*.

▶ **Definition 12.** *A* necklace *in G is a tuple $\mathcal{B} := (u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2)$, where $u^0, u^1, u^2 \in V(G)$ and $\mathcal{Q}^i = \mathcal{Q}^G(u^i, u^{i+1})$ (indices taken modulo 3) is the canonical sps from u^i to u^{i+1}, such that the following conditions are satisfied for every $i \in \{0, 1, 2\}$:*

- *u^0, u^1, u^2 are pairwise distinct.*
- *$V(\mathcal{Q}^i) \cap V(\mathcal{Q}^{i+1}) = \{u^{i+1}\}$ (indices modulo 3).*
- *There is a disk $\boldsymbol{D}^i \subseteq \boldsymbol{S}$ such that $\boldsymbol{G}(\boldsymbol{\mathcal{Q}^i}) \subseteq \boldsymbol{D}^i$.* ⌟

For a necklace $\mathcal{B} := (u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2)$ we write $V(\mathcal{B})$ for the set $\bigcup_{i=0}^{2} \bigcup_{Q \in \mathcal{Q}^i} V(Q)$ and $E(\mathcal{B})$ for $\bigcup_{i=0}^{2} \bigcup_{Q \in \mathcal{Q}^i} E(Q)$, and we let $G(\mathcal{B}) := \big(V(\mathcal{B}), E(\mathcal{B})\big)$. Moreover, we define the set of *articulation vertices* of $\mathcal{B}$ to be $\operatorname{art}(\mathcal{B}) := \bigcup_{i=0}^{2} \operatorname{art}(\mathcal{Q}^i)$.

▶ **Definition 13.** *A necklace $\mathcal{B} := (u^0, \mathcal{Q}^1, u^1, \mathcal{Q}^2, u^2, \mathcal{Q}^3)$ is* reducing *if there are paths $Q^i \in \mathcal{Q}^i$ such that $B := Q^1 \cup Q^2 \cup Q^3$ is a non-contractible cycle.* ⌟

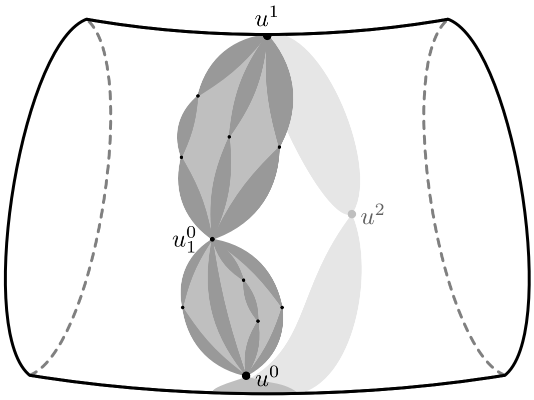

Figure 3 A necklace on a torus section.

We can think of a reducing necklace as a necklace around a handle of our surface (or a crosscap of the surface in the non-orientable case). The beads of the necklace are the disks of the patches that form the necklace. Figure 3 shows a necklace on a torus with articulation vertices u^0, u_1^0, u^1, u^2.

▶ **Lemma 14** (Necklace Lemma)**.** *G has a reducing necklace.*

Essentially, this is [17, Lemma 15.5.8], with the necklaces corresponding to the belts there. But since apart from a renaming, we have also slightly changed the content of the definition of a necklace/belt, the proof needs to be adapted, too. Again, we defer the reader to [18].

5 Upper Bound on the WL Dimension

In this section, we give an outline of the proof of our main theorem (Theorem 1). The full proof can be found in [18].

By the correspondence between k-WL and the logic C^{k+1} stated in Corollary 4, we need to prove that every graph of Euler genus at most g can be identified by a C^{4g+4}-sentence. The proof is by induction on g. The base step $g = 0$ is Theorem 5.

For the inductive step, we make the following assumption.

▶ **Assumption 15.** *$g \geq 1$, and there is a natural number $s \geq 4$ such that every graph in $\mathcal{E}_{g-1}$ can be identified by a C^s-sentence.*

We shall prove that every graph in $\mathcal{E}_g$ can be defined by a C^{s+4}-sentence. Then Theorem 1 follows by induction. After some fairly straightforward reduction steps, which include the reduction to arc-coloured 3-connected graphs from [26], we find that it suffices to prove the following lemma.

▶ **Lemma 16.** *Let G be a coloured graph polyhedrally embedded in a surface $\boldsymbol{S}$ of Euler genus g. Then there is a sentence $\mathsf{iso}_G \in \mathsf{C}^{s+3}$ that identifies G.*

For the rest of the section, we fix a positive integer n. The intended meaning of n is that it is the size of the target graph G. At this point, we have fixed three numerical parameters: the Euler genus g, the number s of variables required to identify graphs of smaller Euler genus, and the order n.

We start the proof by showing that basic objects such as shortest path systems, (pseudo-)patches and (pseudo-)necklaces are definable in the logic C^{s+3}. The following lemma illustrates the type of definability result we can expect, more of the same kind can be found in the full version [18]. Instead of definability in C^k, we actually always study definability in $\mathsf{C}^k_{\mathsf{w}}$ (see Section 2.3) and exploit the fact that every $\mathsf{C}^k_{\mathsf{w}}$-formula is equivalent to a C^k-formula.

▶ **Lemma 17.** *There are formulae* $\mathsf{csps\text{-}vert}(x, x', y) \in \mathsf{C}^3_{\mathsf{w}}$ *and* $\mathsf{csps\text{-}edge}(x, x', y_1, y_2) \in \mathsf{C}^4_{\mathsf{w}}$, *such that for all connected graphs G of order $|G| \leq n$ and all vertices $u, u' \in V(G)$,*

$$\mathsf{csps\text{-}vert}[G, u, u', y] = V\big(\mathcal{Q}^G(u, u')\big),$$
$$\mathsf{csps\text{-}edge}[G, u, u', y_1, y_2] = E\big(\mathcal{Q}^G(u, u')\big).$$

Recall that $\mathcal{Q}^G(u, u')$ is the canonical sps from u to u', that is, the set of all shortest paths from u to u'.

Proof. It is straightforward to define, for every $k \geq 0$, a $\mathsf{C}^3_{\mathsf{w}}$-formula $\mathrm{dist}_{=k}(x, y)$ stating that the distance between the vertices x and y is exactly k. Then we let $\mathsf{csps\text{-}vert}(x, x', y) :=$ $\bigvee_{k=0}^{n} \Big(\mathsf{dist}_{=k}(x, x') \wedge \bigvee_{i=0}^{k} \big(\mathsf{dist}_{=i}(x, y) \wedge \mathsf{dist}_{=k-i}(y, x')\big)\Big)$.

The formula $\mathsf{csps\text{-}edge}(x, x', y_1, y_2)$ can be defined similarly. ◀

With a little more effort, we can prove the following lemma.

▶ **Lemma 18.** *Let $h < g$. Then there is a formula* $\mathsf{csps\text{-}comp\text{-}genus}_h(x, x', y) \in \mathsf{C}^{s+2}_{\mathsf{w}}$ *such that for all connected graphs G of order $|G| \leq n$ and all $u, u', v \in V(G)$ the following holds. Let $\mathcal{Q} := \mathcal{Q}^G(u, u')$, and let A be the connected component of v in $G \setminus G(\mathcal{Q})$ (assuming $v \notin V(\mathcal{Q})$). Then*

$$G \models \mathsf{csps\text{-}comp\text{-}genus}_h(u, u', v) \iff v \notin V(\mathcal{Q}) \text{ and } \operatorname{eg}(A) \leq h.$$

Proof. It follows from Assumption 15 that for every $h < g$, there is a sentence genus_h such that for all graphs G of order $|G| \leq n$, it holds that $G \models \mathsf{genus}_h \iff \operatorname{eg}(G) \leq h$. Indeed, we can simply let genus_h be the disjunction of all sentences iso_H identifying the graphs H with $\operatorname{eg}(H) \leq h$ and $|H| \leq n$.

Using careful bookkeeping and some tricks to reduce the number of variables. we can combine genus_h with the formulae defined in Lemma 17 to obtain the desired formula. ◀

▶ **Corollary 19.** *There is a formula* $\mathsf{csps\text{-}simplifying}(x, x') \in \mathsf{C}^{s+2}_{\mathsf{w}}$ *such that for all connected graphs $G \in \mathcal{E}_g$ of order $|G| \leq n$ and all $u, u' \in V(G)$,*

$$G \models \mathsf{csps\text{-}simplifying}(u, u') \iff \mathcal{Q}^G(u, u') \text{ is simplifying.}$$

The formulae we have defined so far make no reference to an embedding of the input graph. However, if we want to talk about patches and necklaces, we need to take the embedding into account. For the rest of the section, we fix a specific embedded graph G.

▶ **Assumption 20.** *G is a coloured graph of order $|G| = n$ that is polyhedrally embedded in a surface $\boldsymbol{S}$ of Euler genus g.*

It is our goal to construct a $\mathsf{C}^{s+3}_{\mathsf{w}}$-sentence that identifies G. The following lemma is a key step towards this goal, and we find it worthwhile to go into some of the details of its proof.

▶ **Lemma 21.** *There are* C^7_w*-formulae* $\mathsf{int\text{-}vert}(x, x', y)$ *and* $\mathsf{int\text{-}edge}(x, x', y_1, y_2)$ *such that for all vertices* $u, u' \in V(G)$ *for which* $\mathcal{Q} := \mathcal{Q}^G(u, u')$ *is a non-trivial non-simplifying patch, the following holds:*

$$\mathsf{int\text{-}vert}[G, u, u', y] = V\big(I(\mathcal{Q})\big),$$
$$\mathsf{int\text{-}edge}[G, u, u', y_1, y_2] = E\big(I(\mathcal{Q})\big).$$

Intuitively, the lemma says that even though the logical formulae only have access to the abstract graph and the disk of a patch and the internal graph depend on the embedding, we can still define the internal graph. This is non-trivial and somewhat surprising.

Proof. Let $u, u' \in V(G)$ such that $\mathcal{Q} := \mathcal{Q}^G(u, u')$ is a non-trivial non-simplifying patch. Let $\boldsymbol{D} := \boldsymbol{D}(\mathcal{Q})$, $C := C(\mathcal{Q})$, and $I := I(\mathcal{Q})$ (see Section 4).

By Lemma 9, the graph $G \setminus G(\mathcal{Q})$ has a unique non-planar connected component A^*. Since we can detect the planar connected components using the fact that we can identify all planar graphs in C^4, we can also detect A^* as the only non-planar component. This allows us to construct a C^7_w-formula $\mathsf{astar}(x, x', y)$ such that $\mathsf{astar}[G, u, u', y] = V(A^*)$.

Let v_1 be a vertex in $V(\mathcal{Q})$ that is adjacent to A^* and among all such vertices has minimal height in the sps $\mathcal{Q}$, and let h be this height. Since A^* is embedded outside of the disk $\boldsymbol{D}$, the vertex v_1 must be on the boundary cycle C of $\boldsymbol{D}$. There is at most one other vertex of height h on this cycle. Thus, even though v_1 is not unique, there are at most two choices. If there is a second vertex of height h adjacent to A^*, let us call it v_1'. Using the formula $\mathsf{astar}(x, x', y)$ and the fact that vertices of a certain height are definable in C^3_w, we can construct a C^7_w-formula $\varphi_1(x, x', y_1)$ such that v_1 and possibly v_1' are the only vertices in $\varphi_1[G, u, u', y_1]$.

Recall that G/A^* denotes the graph obtained from G by contracting the connected subgraph A^* to a single vertex, which we call a^*, and that the graph G/A^* is a 3-connected planar graph. By Whitney's Theorem, the facial subgraphs (i.e., the subgraphs induced by the boundaries of the faces of an embedding) of a 3-connected plane graph are precisely the chordless non-separating cycles. In particular, they are independent of the embedding. Furthermore, every edge is contained in exactly two of these facial cycles. Let us consider the edge v_1a^* in the graph G/A^*. Let F and F' be the two facial cycles that contain this edge. Both F and F' contain exactly one neighbour of a^* distinct from v_1. Let v_2, v_2' be these neighbours.

By [26, Lemma 22], if we have a 3-connected planar graph H and three vertices w_1, w_2, w_3 on a common facial cycle, then after individualising these three vertices, the 1-dimensional WL algorithm computes a discrete colouring, i.e., a colouring in which every vertex has its own unique colour. By Theorem 3, this implies that for every vertex w of H there is a formula $\psi_{H,w}(z_1, z_2, z_3, y) \in \mathsf{C}^5_\mathsf{w}$ such that $\psi_{H,w}[H, w_1, w_2, w_3, y] = \{w\}$. We apply this to the graph G/A^* and the three vertices a^*, v_1, v_2 and obtain, for every vertex $w \in V(G/A^*) = (V(G) \setminus V(A^*)) \cup \{a^*\}$, a formula $\psi_w(z^*, y_1, y_2, y) \in \mathsf{C}^5_\mathsf{w}$ such that $\psi_w[G/A^*, a^*, v_1, v_2, y] = \{w\}$. From this formula and the formula $\mathsf{astar}(x, x', y)$ we can construct a C^7_w-formula $\widetilde{\psi}_w(x, x', y_1, y_2, z)$ such that $\widetilde{\psi}_w[G, u, u', v_1, v_2, z] = \{w\}$. (Unfortunately, this construction is quite tedious; details can be found in [18].)

Since $A^* \cap \boldsymbol{D} = \emptyset$, we have $V(I) = V(G) \cap \boldsymbol{D} \subseteq V(G \setminus A^*)$. We let

$$\delta(x, x', y_1, y_2, z) := \bigvee_{w \in V(I)} \widetilde{\psi}_w(x, x', y_1, y_2, z).$$

Then $\delta[G, u, u', v_1, v_2, z] = V(I)$. Thus $\delta(x, x', y_1, y_2, z)$ is almost the formula $\mathsf{int\text{-}vert}(x, x', y)$ we want, except that it has two additional parameters v_1, v_2, which we have to get rid of.

We apply [26, Corollary 27], which says that the 3-dimensional WL algorithm *determines orbits* in coloured 3-connected graphs. This means that it distinguishes two vertices if and only if they belong to different orbits of the automorphism group of the given graph. It follows that for every 3-connected planar graph H and for every orbit O of the automorphism group of H, there is a formula $\xi_{H,O}(y_2) \in \mathsf{C}^4_{\mathsf{w}}$ such that $\xi_{H,O}[H, y_2] = O$.

To eliminate the parameter v_2, we apply the corollary to the graph G/A^*, but only after individualising the vertices a^* and v_1. That is, we modify the colouring such that each of the two vertices gets its own colour and is thus fixed by all automorphisms. Let O_2 be the orbit of v_2 in this coloured graph. By the definition of v_2, either $O_2 = \{v_2, v_2'\}$ or $O_2 = \{v_2\}$. Since the graph G/A^* is 3-connected, by eliminating the colour relations for a^* and v_1 at the cost of new free variables z^* and y_1, we obtain a new formula $\psi_2(z^*, y_1, y_2) \in \mathsf{C}^6_{\mathsf{w}}$ such that $\psi_2[G/A^*, a^*, v_1, y_2] = O_2$. We can transform this formula ψ_2 into a $\mathsf{C}^7_{\mathsf{w}}$-formula $\widetilde{\psi}_2(x, x', y_1, y_2)$ such that $\widetilde{\psi}_2[G, u, u', v_1, y_2] = O_2$. We let

$$\delta'(x, x', y_1, z) := \exists y_2 \big(\widetilde{\psi}_2(x, x', y_1, y_2) \wedge \delta(x, x', y_1, y_2, z)\big).$$

If $O_2 = \{v_2\}$, then clearly $\delta'[G, u, u', v_1, z] = \delta[G, u, u', v_1, v_2, z] = V(I)$. So suppose that $O_2 = \{v_2, v_2'\}$, and let f be an automorphism of G with $f(u) = u$, $f(u') = u'$, $f(v_1) = v_1$, and $f(v_2) = v_2'$. By Corollary 11, we have $f\big(V(I)\big) = V(I)$ and thus

$$\begin{aligned} \delta[G, u, u', v_1, v_2', z] &= \delta[f(G), f(u), f(u'), f(v_1), f(v_2), z] \\ &= f\big(\delta[G, u, u', v_1, v_2, z]\big) \\ &= f\big(V(I)\big) = V(I). \end{aligned}$$

It follows that

$$\delta'[G, u, u', v_1, z] = \delta[G, u, u', v_1, v_2, z] \cup \delta[G, u, u', v_1, v_2', z] = V(I).$$

So we have eliminated the parameter v_2. To eliminate v_1, we use a similar argument, which gives us the formula $\mathsf{int\text{-}vert}(x, x', y)$. The formula $\mathsf{int\text{-}edge}(x, x', y_1, y_2)$ can be constructed similarly. ◀

At this point, the proof of Lemma 16 branches into two cases.

Case 1: G does not contain any simplifying patches

By Lemma 14, the graph G contains a reducing necklace. We fix such a necklace $\mathcal{B} = (u^0, \mathcal{Q}^1, u^1, \mathcal{Q}^2, u^2, \mathcal{Q}^3)$. For this, it is sufficient to fix the three vertices u^0, u^1, u^2, because the $\mathcal{Q}^i$ are canonical shortest path systems. We are going to define a subgraph $\mathrm{Cut}(\mathcal{B})$ of G obtained from G by "cutting through the beads". Since the necklace is reducing, the Euler genus of every connected component of $\mathrm{Cut}(\mathcal{B})$ is at most $g - 1$ and we can identify it via a C^s-sentence using Assumption 15. We colour $\mathrm{Cut}(\mathcal{B})$ in such a way that we can reconstruct G and identify it.

Since in this case, there are no simplifying patches, we can define the internal graphs of all patches that form the necklace using Lemma 21. The cut graph $\mathrm{Cut}(\mathcal{B})$ is obtained from G by removing all trivial patches contained in $\mathcal{B}$, all articulation vertices and the internal graphs of all non-trivial patches contained in $\mathcal{B}$. Since the necklace is reducing, there is a non-contractible simple closed curve through the necklace, and we can choose this curve such that it is disjoint from the cut graph. Thus, the cut graph is embeddable in the surface obtained by cutting $\boldsymbol{S}$ along this non-contractible closed curve and therefore has smaller Euler genus.

To prove that G is identified in the logic $\mathsf{C}^{s+3}_{\mathsf{w}}$, we need to show that for every graph $\widehat{G}$ that is not isomorphic to G, there is a $\mathsf{C}^{s+3}_{\mathsf{w}}$-sentence that distinguishes G and $\widehat{G}$. To prove this, we show that if G and $\widehat{G}$ satisfy the same $\mathsf{C}_{\mathsf{w}}^{s+3}$-sentences, then they are isomorphic. We use the fact that the necklace $\mathcal{B}$ is definable in G by $\mathsf{C}_{\mathsf{w}}^{7}$-formulae using the three parameters u^0, u^1, u^2. The same formulae define some object $\widehat{B}$ in $\widehat{G}$. We call $\widehat{B}$ a "pseudo-necklace". It is not necessarily a necklace, because $\widehat{G}$ is not necessarily an embedded graph and objects like patches and necklaces, which depend on an embedding, do not exist in $\widehat{G}$. Nevertheless, from the pseudo-necklace we can define a pseudo-cut graph $\mathrm{Cut}(\widehat{\mathcal{B}})$. Since $\mathrm{Cut}(\mathcal{B})$ is identified in the logic $\mathsf{C}_{\mathsf{w}}^{s}$, we can show that $\mathrm{Cut}(\mathcal{B})$ and $\mathrm{Cut}(\widehat{\mathcal{B}})$ are isomorphic. Since we can reconstruct G and $\widehat{G}$ from the respective (pseudo-)cut graphs, we conclude that G and $\widehat{G}$ are isomorphic.

Case 2: G contains a simplifying patch

This case sounds simpler than the first one: instead of a complicated necklace, here we only need to remove a simplifying patch from our graph. The remaining pieces have smaller Euler genus and thus can be identified in the logic $\mathsf{C}_{\mathsf{w}}^{s}$. Hence, all we need to do is colour the pieces in such a way that we can reconstruct the original graph. The problem with this line of reasoning is that simplifying patches have a much more complicated structure than non-simplifying patches. For example, we cannot define the interior of a simplifying patch in the same way as we did for non-simplifying patches in Lemma 21 since there is not necessarily a non-planar connected component which marks the "outside region" of the patch. Therefore, the idea of the proof is to remove the canonical sps and some interior parts of the corresponding patch, which are actually interiors of non-simplifying subpatches and thus definable by Lemma 21.

More precisely, we fix two vertices u and u' such that $\mathcal{Q} := \mathcal{Q}^G(u, u')$ is a minimal simplifying canonical patch in G, that is, a simplifying canonical patch all of whose proper canonical subpatches are non-simplifying. We extend $\mathcal{Q}$ by the internal graphs of all proper subpatches and obtain a graph J, which is actually embedded in the disk of $\mathcal{Q}$ and therefore planar.

Now we distinguish between two cases. If $J \setminus \{u, u'\}$ is connected, the patch $\mathcal{Q}$ behaves almost like a non-simplifying patch, and we can argue similarly as in Case 1. If $J \setminus \{u, u'\}$ is disconnected, the patch $\mathcal{Q}$ can be split into several so-called *fibres*. The subgraph J contained in the fibres is definable in our logic when we fix one more particular vertex. In fact, the subgraph contained in every single fibre is definable. We exploit this in order to encode in the boundary vertices of the fibres the way in which the remainder of the graph is attached to them, similarly as in Case 1, but due to the possibly complex structure of J a lot more involved.

An improved bound in case the surface is orientable

The combination of Case 1 and Case 2 yields Theorem 1. Exploiting our inductive approach further, we can deduce a better bound in case we know that the given graph is embeddable in an *orientable* surface of Euler genus at most g, as stated in Corollary 2.

Proof of Corollary 2. The Euler genus of an orientable surface is always even. Suppose G is a graph embeddable in an orientable surface of Euler genus g. Since the subgraphs obtained by cutting through the beads are also embeddable in orientable surfaces of smaller Euler genus, their Euler genus is at least 2 smaller than the Euler genus of G. Therefore, proceeding inductively and redefining s to be the number of variables needed for graphs embeddable in orientable surfaces of Euler genus at most $g - 2$, we can improve our bound from Theorem 1 to $2g + 3$. ◀

6 Concluding Remarks

The WL dimension is a measure for the combinatorial and descriptive complexity of a graph. In view of its numerous, seemingly unrelated characterisations in terms of logic, algebra, mathematical programming, and homomorphisms, we can arguably regard the WL dimension as a natural and robust graph invariant.

We have proved an upper bound of $4g+3$ for the WL dimension of graphs of Euler genus g and showed that if G is known to be embeddable in an orientable surface of Euler genus g, the bound improves to $2g+3$. The immediate remaining question is how tight our bound is.

We believe that by refining our arguments in some places it might be possible to reduce the bound from Theorem 1 to $3g+3$ or even $2g+3$; any further improvement seems to require substantial additional ideas. It is conceivable that the WL dimension of planar graphs is 2. If this is the case, the additive term in our bound would automatically drop to 2.

In terms of lower bounds, using the so-called CFI construction [8], it is easy to prove a linear lower bound of $\epsilon \cdot g$ for the WL dimension of graphs of Euler genus g, albeit with a rather small constant $\epsilon > 0$. To close the gap between upper and lower bound, it may be worthwhile to spend some effort on improving the lower bound.

Beyond graphs of bounded genus, we can try to determine the WL dimension of other graph classes and tie the WL dimension to other graph invariants. A natural target would be the class of all graphs that exclude the complete graph K_ℓ as a minor. We know that the WL dimension of this class is bounded [17]. But even an exponential bound on the WL dimension in terms of ℓ would be major progress.

References

1 B. Ahmadi, K. Kersting, M. Mladenov, and S. Natarajan. Exploiting symmetries for scaling loopy belief propagation and relational training. *Machine Learning Journal*, 92(1):91–132, 2013.

2 V. Arvind, J. Köbler, G. Rattan, and O. Verbitsky. On Tinhofer's Linear Programming Approach to Isomorphism Testing. In G. F. Italiano, G. Pighizzini, and D. Sannella, editors, *Proceedings of the 40th International Symposium on Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS, Part II*, volume 9235 of *Lecture Notes in Computer Science*, pages 26–37. Springer Verlag, 2015.

3 V. Arvind, J. Köbler, G. Rattan, and O. Verbitsky. Graph Isomorphism, Color Refinement, and Compactness. *Computational Complexity*, 26(3):627–685, 2017.

4 A. Atserias and E. Maneva. Sherali–Adams relaxations and indistinguishability in counting logics. *SIAM J. Comput.*, 42(1):112–137, 2013.

5 A. Atserias and J. Ochremiak. Definable Ellipsoid Method, Sums-of-Squares Proofs, and the Isomorphism Problem. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 66–75, 2018.

6 L. Babai. Graph Isomorphism in Quasipolynomial Time. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC '16)*, pages 684–697, 2016.

7 C. Berkholz and M. Grohe. Limitations of Algebraic Approaches to Graph Isomorphism Testing. In M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, editors, *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 155–166. Springer Verlag, 2015.

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

9 P. T. Darga, M. H. Liffiton, K. A. Sakallah, and I. L. Markov. Exploiting structure in symmetry detection for CNF. In *Proceedings of the 41st Design Automation Conference*, pages 530–534. ACM, 2004. doi:10.1145/996566.996712.

10 H. Dell, M. Grohe, and G. Rattan. Lovász Meets Weisfeiler and Leman. In I. Chatzigiannakis, C. Kaklamanis, D. Marx, and D. Sannella, editors, *Proceedings of the 45th International*

Colloquium on Automata, Languages and Programming (Track A), volume 107 of *LIPIcs*, pages 40:1–40:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

11 R. Diestel. *Graph Theory.* Springer Verlag, 4th edition, 2010.

12 M. Fürer. On the Combinatorial Power of the Weisfeiler-Lehman Algorithm. In D. Fotakis, A. Pagourtzis, and V. Th. Paschos, editors, *Proceedings of the 10th International Conference on Algorithms and Complexity*, volume 10236 of *Lecture Notes in Computer Science*, pages 260–271. Springer Verlag, 2017.

13 E. Grädel, M. Grohe, B. Pago, and W. Pakusa. A Finite-Model-Theoretic View on Propositional Proof Complexity, 2018. arXiv:1802.09377.

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

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

16 M. Grohe. Fixed-Point Definability and Polynomial Time on Graphs with Excluded Minors. *Journal of the ACM*, 59(5), 2012.

17 M. Grohe. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, volume 47 of *Lecture Notes in Logic.* Cambridge University Press, 2017.

18 M. Grohe and S. Kiefer. A Linear Upper Bound on the Weisfeiler-Leman Dimension of Graphs of Bounded Genus, 2019. arXiv:1904.07216.

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

20 M. Grohe and D. Neuen. Canonisation and Definability for Graphs of Bounded Rank Width, 2019. arXiv:1901.10330.

21 M. Grohe and M. Otto. Pebble Games and Linear Equations. *Journal of Symbolic Logic*, 80(3):797–844, 2015.

22 W. L. Hamilton, Z. Ying, and J. Leskovec. Inductive Representation Learning on Large Graphs. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, pages 1025–1035, 2017. URL: http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs.

23 N. Immerman. *Descriptive Complexity.* Springer Verlag, 1999.

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

25 T. A. Junttila and P. Kaski. Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs. In *Proceedings of the 9th SIAM Workshop on Algorithm Engineering and Experiments.* SIAM, 2007. doi:10.1137/1.9781611972870.13.

26 S. Kiefer, I. Ponomarenko, and P. Schweitzer. The Weisfeiler-Leman dimension of planar graphs is at most 3. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science*, 2017.

27 S. Kiefer and P. Schweitzer. Upper Bounds on the Quantifier Depth for Graph Differentiation in First Order Logic. In M. Grohe, E. Koskinen, and N. Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 287–296, 2016.

28 S. Kiefer, P. Schweitzer, and E. Selman. Graphs Identified by Logics with Counting. In G. F. Italiano, G. Pighizzini, and D. Sannella, editors, *Proceedings of the 40th International Symposium on Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS, Part I*, volume 9235 of *Lecture Notes in Computer Science*, pages 319–330. Springer Verlag, 2015.

29 T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017. URL: https://openreview.net/forum?id=SJU4ayYgl.

30 J. Köbler, S. Kuhnert, B. Laubner, and O. Verbitsky. Interval graphs: Canonical representations in logspace. *SIAM Journal on Computing*, 40(5):1292–1315, 2011.

31 A. Krebs and O. Verbitsky. Universal Covers, Color Refinement, and Two-Variable Counting Logic: Lower Bounds for the Depth. In *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 689–700, 2015.

32 B. Laubner. Capturing Polynomial Time on Interval Graphs. In *Proceedings of the 25th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 199–208, 2010.

33 P. Malkin. Sherali-Adams relaxations of graph isomorphism polytopes. *Discrete Optimization*, 12:73–97, 2014.

34 B. D. McKay. Practical graph isomorphism. *Congressus Numerantium*, 30:45–87, 1981.

35 B. D. McKay and A. Piperno. Practical graph isomorphism, II. *J. Symb. Comput.*, 60:94–112, 2014. doi:10.1016/j.jsc.2013.09.003.

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

37 C. Morris, M. Ritzert, M. Fey, W. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019.

38 R. O'Donnell, J. Wright, C. Wu, and Y. Zhou. Hardness of Robust Graph Isomorphism, Lasserre Gaps, and Asymmetry of Random Graphs. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1659–1677, 2014.

39 M. Otto. *Bounded variable logics and counting – A study in finite models*, volume 9 of *Lecture Notes in Logic*. Springer Verlag, 1997.

40 N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

41 B. Weisfeiler and A. Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series 2*, 1968. English translation by G. Ryabov available at https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf.