LOGICAL COMPLEXITY OF GRAPHS: A SURVEY

OLEG PIKHURKO* AND OLEG VERBITSKY†

Abstract. We discuss the definability of finite graphs in first-order logic with two relation symbols for adjacency and equality of vertices. The *logical depth* $D(G)$ of a graph G is equal to the minimum quantifier depth of a sentence defining G up to isomorphism. The *logical width* $W(G)$ is the minimum number of variables occurring in such a sentence. The *logical length* $L(G)$ is the length of a shortest defining sentence. We survey known estimates for these graph parameters and discuss their relations to other topics (such as the efficiency of the Weisfeiler-Lehman algorithm in isomorphism testing, the evolution of a random graph, or the contribution of Frank Ramsey to the research on Hilbert's *Entscheidungsproblem*). Also, we trace the behavior of the descriptive complexity of a graph as the logic becomes more restrictive (for example, only definitions with a bounded number of variables or quantifier alternations are allowed) or more expressible (after powering with counting quantifiers).

Date: 24 March 2010.

* Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213, USA. This work done under the support of the National Science Foundation (Grant DMS-0758057) and the Alexander von Humboldt Foundation.

† Institute for Applied Problems of Mechanics and Mathematics, 79060 Lviv, Ukraine. This work was done under the support of the Alexander von Humboldt Foundation.

Contents

1. Introduction

1.1. **Basic notions and examples.** We consider the first-order language of graph theory whose vocabulary contains two relation symbols $\sim$ and $=$, respectively for adjacency and equality of vertices. The term *first-order* imposes the condition that the variables represent vertices and hence the quantifiers apply to vertices only. Without quantification over sets of vertices, we are unable to express by a single formula some basic properties of graphs, such as being bipartite, being connected, etc. (see, e.g., [71, Theorems 2.4.1 and 2.4.2]). However, first-order logic is powerful enough to define any *individual* graph. How succinctly this can be done is the subject of this article.

As a starting example, let us say in the first-order language that vertices x and y are at distance at most n from one another. A possible formula $\Delta_n(x,y)$ can look as follows:

$$\begin{aligned}\Delta_1(x,y) &\stackrel{\text{def}}{=} x \sim y \vee x = y,\\ \Delta_n(x,y) &\stackrel{\text{def}}{=} \exists z_1 \ldots \exists z_{n-1}\Big(\Delta_1(x,z_1) \wedge \bigwedge_{i=1}^{n-2} \Delta_1(z_i,z_{i+1}) \wedge \Delta_1(z_{n-1},y)\Big).\end{aligned} \tag{1}$$

By a *sentence* we mean a first-order formula where every variable is bound by a quantifier. If we specify a graph G, a sentence Φ is either *true* or *false* on it. If H is a graph isomorphic to G, then Φ is either true or false on G and H simultaneously. In other words, first-order logic cannot distinguish between isomorphic graphs. In general, we say that a sentence Φ *distinguishes* a graph G from another graph H if Φ is true on G but false on H.

For example, sentence $\forall x \forall y\, \Delta_1(x,y)$ distinguishes a complete graph K_n from any graph H that is not complete. The sentence $\forall x \forall y\, \Delta_{n-1}(x,y)$ distinguishes P_n, the path with n vertices, from any longer path P_m, $m > n$.

Throughout this survey we consider only graphs whose vertex set is finite and non-empty. We say that a sentence Φ *defines* a graph G (up to isomorphism) if Φ distinguishes G from every non-isomorphic graph H.

For example, the single-vertex graph P_1 is defined by sentence $\forall x \forall y\,(x = y)$. If $n \geq 2$, then the path P_n is defined by

$$\begin{aligned}&\forall x \forall y \Delta_{n-1}(x,y) \wedge \neg \forall x \forall y \Delta_{n-2}(x,y)\\ &\qquad\qquad \texttt{to say that the diameter equals } n-1\\ &\wedge \forall x \neg \exists y_1 \exists y_2 \exists y_3 \left(\bigwedge_{i=1,2,3} x \sim y_i \wedge \bigwedge_{i \neq j} \neg(y_i = y_j)\right)\\ &\qquad\qquad \texttt{to say that the maximum degree } \leq 2\\ &\wedge \exists x \neg \exists y_1 \exists y_2 \left(\bigwedge_{i=1,2} x \sim y_i \wedge \neg(y_1 = y_2)\right)\\ &\qquad\qquad \texttt{to say that the minimum degree } \leq 1 \texttt{ (thereby}\\ &\qquad\qquad \texttt{distinguishing from cycles } C_{2n-2} \texttt{ and } C_{2n-1}\texttt{)}\end{aligned} \tag{2}$$

We have already mentioned the following basic fact: *Every finite graph G is definable.*[1] Indeed, let $V(G) = \{v_1, \ldots, v_n\}$ be the vertex set of G and $E(G)$ be its edge set. A sentence defining G could read:

$$\begin{gathered} \exists x_1 \ldots \exists x_n \ \left(\mathrm{Distinct}(x_1, \ldots, x_n) \wedge \mathrm{Adj}(x_1, \ldots, x_n) \right) \\ \wedge\ \forall x_1 \ldots \forall x_{n+1}\ \neg\, \mathrm{Distinct}(x_1, \ldots, x_{n+1}), \end{gathered} \tag{3}$$

where, for the notational convenience, we use the following shorthands

$$\mathrm{Distinct}(x_1, \ldots, x_k) \ \stackrel{\mathrm{def}}{=} \ \bigwedge_{1 \le i < j \le k} \neg\, (x_i = x_j),$$

$$\mathrm{Adj}(x_1, \ldots, x_n) \ \stackrel{\mathrm{def}}{=} \ \bigwedge_{\{v_i, v_j\} \in E(G)} x_i \sim x_j \ \wedge \bigwedge_{\{v_i, v_j\} \notin E(G)} \neg\, (x_i \sim x_j).$$

In other words, we first specify that there are n distinct vertices, list the adjacencies and the non-adjacencies between them, and then state that we cannot find $n + 1$ distinct vertices.

The sentence (3) is an exhaustive description of G and seems rather wasteful. We want to know if there is a more succinct way of defining a graph on n vertices. The following natural succinctness measures of a first-order formula Φ are of interest:

- the *length* $L(\Phi)$ which is the total number of symbols in Φ (each variable symbol contributes 1);
- the *quantifier depth* $D(\Phi)$ which is the maximum length of a chain of nested quantifiers in Φ;
- the *width* $W(\Phi)$ which is the number of variables used in Φ (different occurrences of the same variable are not counted).[2]

Formula Δ_n in (1) was intentionally written in a non-optimal way. Note that $L(\Delta_n) = \Theta(n)$, $D(\Delta_n) = n - 1$, and $W(\Delta_n) = n + 1$. The same distance restriction can be expressed more succinctly with respect to the latter two parameters, namely

$$\begin{aligned} \Delta'_1(x, y) &\ \stackrel{\mathrm{def}}{=}\ \Delta_1(x, y), \\ \Delta'_n(x, y) &\ \stackrel{\mathrm{def}}{=}\ \exists z \left(\Delta'_{\lfloor n/2 \rfloor}(x, z) \wedge \Delta'_{\lceil n/2 \rceil}(z, y) \right), \end{aligned} \tag{4}$$

where $\lceil x \rceil$ (resp. $\lfloor x \rfloor$) stands for the integer nearest to x from above (resp. from below). Now $D(\Delta'_n) = \lceil \log_2 n \rceil$, giving an exponential gain for the quantifier depth! The width can be reduced even more drastically: by recycling variables we can write Δ'_n with only 3 variables in total, achieving $W(\Delta'_n) = 3$.

We now come to the central concepts of our survey. Let us define $L(G)$ (resp. $D(G)$, $W(G)$) to be the minimum of $L(\Phi)$ (resp. $D(\Phi)$, $W(\Phi)$) over all sentences

[1] This fact, though very simple, highlights a fundamental difference between the finite and the infinite: There are non-isomorphic countable graphs satisfying precisely the same first-order sentences (see, e.g., [71, Theorem 3.3.2]).

[2] Grädel [31] defines the width of a formula Φ as the maximum number of free variables in a subformula of Φ. Denote this version by $W'(\Phi)$. Clearly, $W'(\Phi) \le W(\Phi)$ and the inequality can be strict. Nevertheless, the two parameters are closely related: Φ can be rewritten by renaming bound variables in an equivalent form Φ' so that $W(\Phi') = W'(\Phi)$; see [31, Lemma 3.1.4].

Φ defining a graph G. We will call these graph invariants, respectively, the *logical length*, *depth*, and *width* of G.

Example 1.1.

1. Using Δ'_n in place of Δ_n in (2), we see that $D(P_n) < \log_2 n + 3$ and $W(P_n) \le 4$. The reader is encouraged to improve the latter to $W(P_n) \le 3$.
2. The generic defining sentence (3) shows that $L(G) = O(n^2)$ and $D(G) \le n+1$ for every graph G on n vertices.
3. The *complement* of G, denoted by $\overline{G}$, is the graph on the same vertex set $V(G)$ whose edges are those pairs that are not in $E(G)$. One can easily prove that $D(\overline{G}) = D(G)$ and $W(\overline{G}) = W(G)$.

The logical length, depth, and width of a graph satisfy the following inequalities:

$$W(G) \le D(G) < L(G).$$

The latter relation follows from an obvious fact that $D(\Phi) < L(\Phi)$ for any first-order formula Φ. The former follows from a bit less obvious fact that for any first-order formula Φ there is a logically equivalent formula Ψ with $W(\Psi) \le D(\Phi)$.

1.2. **Variations of logic.**

1.2.1. *Fragments.* Suppose that we put some restrictions on the structure of a defining sentence. This may cause an increase in the resources (length, depth, width) that we need in order to define a graph in the straitened circumstances. These effects will be one of our main concerns in this survey. We will deal with restrictions of the following two sorts. We may be allowed to make only a small (constant) number of quantifier alternations or to use only a bounded number of variables. The former is commonly used in logic and complexity theory to obtain hierarchical classifications of various problems. The latter is in the focus of *finite-variable logics* (see, e.g, Grohe [32]). Moreover, the number of variables has relevance to the computational complexity of the graph isomorphism problem, see Section 4.
Bounded number of quantifier alternations. A first-order formula Φ with connectives $\{\neg, \wedge, \vee\}$ is in a *negation normal form* if all negations apply only to relations (one can think that we now do not have negation at all but introduce instead two new relation symbols, for inequality and non-adjacency). It is well known that this structural restriction actually does not make first-order logic weaker: We can always move negations in front of relation symbols without increasing the formula's length more than twice and without changing the quantifier depth and the width.

Given such a formula Φ and a sequence of nested quantifiers in it, we count the number of *quantifier alternations*, that is, the number of successive pairs $\forall\exists$ and $\exists\forall$ in the sequence. The *alternation number* of Φ is the maximum number of quantifier alternations over all such sequences. The *a-alternation logic* consists of all first-order formulas in the negation normal form whose alternation number does not exceed a. We will adhere to the following notational convention: a subscript a will always indicate that at most a quantifier alternations are allowed. For example, $D_a(G)$ is the minimum quantifier depth of a sentence in the a-alternation logic that defines a graph G.

For any graph G on n vertices we have

$$D(G) \le \ldots \le D_{a+1}(G) \le D_a(G) \le \ldots \le D_1(G) \le D_0(G) \le n+1,$$

where the last bound is due to the defining sentence (3).

Bounded number of variables. The *k-variable logic* is the fragment of first-order logic where only k variable symbols are available, that is, the formula width is bounded by k. The restriction of defining sentences to the k-variable logic will be always indicated by a superscript k. To make this notation always applicable, we set $D^k(G) = \infty$ if the k-variable logic is too weak to define G. If $k \ge W(G)$ for a graph G of order n, then we have

$$D(G) \le D^{k+1}(G) \le D^k(G) < n^{k-1} + k,$$

where the last bound will be established in Theorem 4.7 below. Note that the bounds in Example 1.1.1 can be strengthened to $D^3(P_n) < \log_2 n + 3$.

1.2.2. *An extension with counting quantifiers.* We will also enrich first-order logic by allowing one to use expressions of the type $\exists^m \Psi$ in order to say that there are at least m vertices with property Ψ. Those are called *counting quantifiers* and the extended logic will be referred to as *counting logic*. A counting quantifier $\exists^m$ contributes 1 in the quantifier depth irrespectively of the value of m. For the counting logic we will use the "sharp-notation", thus denoting the logical depth and width of a graph G in this logic, respectively, by $D_\#(G)$ and $W_\#(G)$. Clearly, $D_\#(G) \le D(G)$ and $W_\#(G) \le W(G)$. The counting quantifiers often allow us to define a graph much more succinctly. For example, $D_\#(K_n) = W_\#(K_n) = 2$ as this graph is defined by

$$\forall x \forall y\, (x \sim y \vee x = y) \wedge \exists^n x\, (x = x) \wedge \neg \exists^{n+1} x\, (x = x).$$

This is in sharp contrast with the fact that $D(K_n) = W(K_n) = n+1$, where the lower bound follows from the simple observation that n variables are not enough to distinguish between K_n and K_{n+1}.

1.3. **Outline of the survey.** Section 2 specifies notation and proves a couple of basic facts about first-order sentences. The latter are applied to establish an upper bound on the logical length $L(G)$ of a graph in terms of its logical depth $D(G)$ and to estimate from above the number of graphs whose logical depth is bounded by a given parameter k. The existence of such bounds is more important than the bounds themselves that are huge, involving the tower function. Furthermore, we define $D(G,H)$ to be the smallest quantifier depth sufficient to distinguish between non-isomorphic graphs G and H. We will observe that the obvious inequality $D(G,H) \le D(G)$ gives the sharp lower bound on $D(G)$. Thus estimating $D(G)$ reduces to estimating $D(G,H)$ for all $H \not\cong G$

The value of $D(G,H)$ is characterized in Section 3 as the length of the *Ehrenfeucht game* on G and H. Moreover, the logical width admits a characterization in terms of another parameter of the game. Thus, the determination of the logical depth and width of a graph reduces to designing optimal strategies in the Ehrenfeucht game.

In Section 4, the logical width and the logical depth are also characterized, respectively, as the minimum *dimension* and the minimum *number of rounds* such that the so-called *Weisfeiler-Lehman algorithm* returns the correct answer. The algorithm tries to decide whether two input graphs are isomorphic; its one-dimensional version is just the well-known color-refining procedure. Thus, an analysis of the algorithm can give us information on the logical complexity of the input graphs. This relationship is even more advantageous in the other direction: Once we prove that all graphs in some class C have low logical complexity, we immediately obtain an efficient isomorphism test for C.

This paradigm is successful for graphs with bounded treewidth and planar graphs, with good prospects for covering all classes of graphs with an excluded minor. In Section 5.1 we report strong upper bounds for the logical depth/width of graphs in these classes. In Section 5.2 we survey the bounds known in the general case. In particular, if a graph G on n vertices has no *twins*, i.e., vertices with the same neighborhood, then $D(G) < \frac{1}{2}n+3$. The factor of $\frac{1}{2}$ can be improved for graphs with bounded vertex degrees. Here we have to content ourselves with linear bounds in view of a linear lower bound by Cai, Fürer, and Immerman [14]. They constructed examples of graphs with maximum degree 3 such that $W_{\#}(G) > c\,n$ for a positive constant c.

Section 6 discusses the logical complexity of a random graph. We obtain rather close lower and upper bounds for *almost all* graphs. Furthermore, we trace the behavior of the logical depth in the evolutional random graph model $G_{n,p}$ where p is a function of n.

While in Sections 5 and 6 we deal with, respectively, worst case and average case bounds, Section 7 is devoted to the *best case*. More specifically, we define *succinctness function* $q(n)$ to be equal to the minimum of $D(G)$ over all G on n vertices. Since only finitely many graphs are definable with a fixed quantifier depth, $q(n)$ goes to infinity as n increases. It turns out that its growth is inconceivably slow: We show a superrecursive gap between the values of $q(n)$ and n. This phenomenon disappears if we "smoothen" $q(n)$ by considering the least monotonic upper bound for this function: the smoothed succinctness function is very close to the log-star function. Furthermore, the succinctness function can be considered in any logic. Let $q_0(n)$ be its variant for the logic with no quantifier alternation. We can determine $q_0(n)$ with rather high precision: It is also related to the log-star function. The lower bound for $q_0(n)$ implies a superrecursive gap between the graph parameters $D(G)$ and $D_0(G)$, yet another evidence of the weakness of the 0-alternation logic. The tight upper bound for $q_0(n)$ shows that, nevertheless, there are graphs whose definitions, even if quantifiers are not allowed to alternate, can have surprisingly low quantifier depth. We give several methods of explicit constructions of such graphs. These constructions have another interesting aspect. They allow us to show that the previously mentioned tower-function bounds from Section 2 cannot be improved substantially.

Some of the most interesting open questions are collected in Section 8.

1.4. **Other structures.** Some of the results presented in the survey generalize to relational structures over a fixed vocabulary. Such generalizations are often straightforward. For example, the upper bounds on succinctness functions hold true if the vocabulary contains at least one relation symbol of arity more than 1 (since any graph can be trivially represented as a structure over this vocabulary). Extension of the worst case bounds to general structures is also possible but requires essential additional efforts; see [64].

Various definability parameters were investigated also for special structures: colored graphs (Immerman and Lander [45], Cai, Fürer, and Immerman [14]), digraphs and hypergraphs (Pikhurko, Veith, and Verbitsky [63]), bit strings and ordered trees (Spencer and St. John [72]), linear orders (Grohe and Schweikardt [39]).

2. Preliminaries

2.1. **Notation: Arithmetic and graphs.** We define the *tower function* by $Tower(0) = 1$ and $Tower(i) = 2^{Tower(i-1)}$ for each subsequent integer i. Given a function f, by $f^{(i)}(x)$ we will denote the i-fold composition of f. In particular, $f^{(0)}(x) = x$. By $\log n$ we always mean the logarithm base 2. The "inverse" of the tower function, the *log-star* function $\log^* n$, is defined by $\log^* n = \min\{i : Tower(i) \geq n\}$. We use the standard asymptotic notation. For example, $f(n) = \Omega(g(n))$ means that there is a constant $c > 0$ such that $f(n) \geq c\, g(n)$ for all sufficiently large n.

The number of vertices in a graph G is called the *order* of G and is denoted by $v(G)$. The *neighborhood* $N(v)$ of a vertex v consists of all vertices adjacent to v. The *degree* of v is defined by $\deg v = |N(v)|$. The *maximum degree* of a graph G is defined by $\Delta(G) = \max_{v \in V(G)} \deg v$.

The *distance* between vertices u and v in a graph G is defined to be the minimum length of a path from u to v and denoted by $dist(u, v)$. If u and v are in different connectivity components, then we set $dist(u, v) = \infty$. The *eccentricity* of a vertex v is defined by $e(v) = \max_{u \in V(G)} dist(v, u)$.

Let $X \subset V(G)$. The subgraph induced by G on X is denoted by $G[X]$. We denote $G \setminus X = G[V(G) \setminus X]$, which is the result of the removal of all vertices in X from G. If a single vertex v is removed, we write $G - v = G \setminus \{v\}$. A set of vertices X is called *homogeneous* if $G[X]$ is a complete or an empty graph.

A graph is *k-connected* if it has at least $k+1$ vertices and remains connected after removal of any $k - 1$ vertices. 2-connected graphs are also called *biconnected*.

A graph is *asymmetric* if it admits no non-trivial automorphism.

2.2. **A length-depth relation.** We have already mentioned the trivial relation $D(G) < L(G)$. Now we aim at bounding $L(G)$ from above in terms of $D(G)$. We write $G \equiv_k H$ to say that graphs G and H cannot be distinguished by any sentence with quantifier depth k. As it is easy to see, $\equiv_k$ is an equivalence relation. Its equivalence classes will be referred to as *$\equiv_k$-classes*. We say that a sentence Φ *defines* a $\equiv_k$-class α if Φ is true on all graphs in α and false on all other graphs.

Lemma 2.1.

1. *The number of $\equiv_k$-classes is finite and does not exceed $Tower(k+\log^* k+2)$.*
2. *Every $\equiv_k$-class is definable by a sentence Φ with $D(\Phi) = k$ and $L(\Phi) < Tower(k + \log^* k + 2)$.*

Proof. The case of $k = 1$ is easy: There is only one $\equiv_1$-class (consisting of all graphs), which is definable by $\forall x(x = x)$.

Let $k \geq 2$ and $0 \leq s \leq k$. When we write $\bar{z}$, we will mean an s-tuple $(z_1, \ldots, z_s)$ (if $s = 0$, the sequence is empty). If $\bar{u} \in V(G)^s$ and Φ is a formula with s free variables $x_1, \ldots, x_s$, then notation $G, \bar{u} \models \Phi(\bar{x})$ will mean that $\Phi(\bar{x})$ is true on G with each x_i being assigned the respective u_i as its value.

A formula $\Phi(x_1, \ldots, x_s)$ of quantifier depth $k - s$ is *normal* if Φ is built from variables $x_1, \ldots, x_k$ and every maximal sequence of nested quantifiers in Φ has length $k - s$ and quantifies the variables $x_{s+1}, \ldots, x_k$ exactly in this order. A simple inductive syntactic argument shows that any $\Phi(x_1, \ldots, x_s)$ has an equivalent normal formula $\Phi'(x_1, \ldots, x_s)$ of the same quantifier depth as Φ.

We write $G, \bar{u} \equiv_{k,s} H, \bar{v}$ to say that $G, \bar{u} \models \Phi(\bar{x})$ exactly when $H, \bar{v} \models \Phi(\bar{x})$ for every normal formula Φ of quantifier depth $k - s$. A normal formula $\Phi(\bar{x})$ *defines* a $\equiv_{k,s}$-class α if $G, \bar{u} \models \Phi(\bar{x})$ exactly when $G, \bar{u}$ belongs to α. The $\equiv_{k,s}$-equivalence class of $G, \bar{u}$ will be denoted by $[G, \bar{u}]_{k,s}$.

Let $f(k, s)$ denote the number of all $\equiv_{k,s}$-classes and $l(k, s)$ denote the minimum l such that every $\equiv_{k,s}$-class is definable by a normal formula of depth at most $k - s$ and length at most l. Note that relations $\equiv_k$ and $\equiv_{k,0}$ coincide. Thus, our goal is to estimate the numbers $f(k, 0)$ and $l(k, 0)$ from above.

We use the backward induction on s. A $\equiv_{k,k}$-class can be determined by specifying, for each pair of the k elements, whether they are equal and, if not, whether they are adjacent or non-adjacent. There are at most three choices per pair. It easily follows that $f(k, k) \leq 3^{\binom{k}{2}}$ and $l(k, k) < 9k^2$. We are now going to estimate $f(k, s)$ and $l(k, s)$ in terms of $f(k, s + 1)$ and $l(k, s + 1)$. Suppose that each $\equiv_{k,s+1}$-class β is defined by a formula $\Phi_\beta(x_1, \ldots, x_s, x_{s+1})$ whose length is bounded by $l(k, s + 1)$.

Define $S(G, \bar{u}) = \{ [G, \bar{u}, u]_{k,s+1} : u \in V(G)\}$, the set of $\equiv_{k,s+1}$-classes obtainable from $G, \bar{u}$ by specifying one extra vertex. Note that

$$G, \bar{u} \equiv_{k,s} H, \bar{v} \text{ if and only if } S(G, \bar{u}) = S(H, \bar{v}).$$

Indeed, suppose that $S(G, \bar{u}) \neq S(H, \bar{v})$, say, $\beta = [G, \bar{u}, u]_{k,s+1}$ is not in $S(H, \bar{v})$ for some $u \in V(G)$. Then $G, \bar{u} \not\equiv_{k,s} H, \bar{v}$ because formula $\exists x_{s+1} \Phi_\beta$ is true for $G, \bar{u}$ but false for $H, \bar{v}$. Suppose now that $G, \bar{u}$ and $H, \bar{v}$ are distinguishable by a normal formula of quantifier depth $k - s$. As it is easily seen, they are distinguishable by such a formula of the form $\exists x_{s+1} \Phi$. Without loss of generality, assume that the formula $\exists x_{s+1} \Phi$ is true for $G, \bar{u}$ but false for $H, \bar{v}$. Let $u \in V(G)$ be such that $G, \bar{u}, u \models \Phi$. Since Φ distinguishes $G, \bar{u}, u$ from all $H, \bar{v}, v$ with $v \in V(H)$, the class $[G, \bar{u}, u]_{k,s+1}$ is not in $S(H, \bar{v})$ and, hence, $S(G, \bar{u}) \neq S(H, \bar{v})$.

Thus, for a $\equiv_{k,s}$-class α we can correctly define the set of $\equiv_{k,s+1}$-classes accessible from α by $S(\alpha) = S(G, \bar{u})$ for some (in fact, arbitrary) $G, \bar{u}$ in α. It follows from what we have proved that for arbitrary $\equiv_{k,s}$-classes α and α', we have

$$\alpha = \alpha' \text{ if and only if } S(\alpha) = S(\alpha').$$

As an immediate consequence,

$$f(k, s) \leq 2^{f(k,s+1)}.$$

Since $2\cdot\binom{k}{2} \le 2^k$ for every integer $k \ge 1$, we have $f(k,k) \le 2^{2^k} \le \mathit{Tower}(\log^* k+2)$. By the above recursion, we conclude that $f(k,0) \le \mathit{Tower}(k+\log^* k+2)$, which proves Part 1 of the lemma.

Another conclusion is that any $\equiv_{k,s}$-class α can be defined by a normal formula

$$\Phi_\alpha(\bar{x}) \stackrel{\mathrm{def}}{=} \bigwedge_{\beta\in S(\alpha)} \exists x_{s+1}\, \Phi_\beta(\bar{x}, x_{s+1}) \;\wedge\; \forall x_{s+1} \bigwedge_{\beta\notin S(\alpha)} \neg\, \Phi_\beta(\bar{x}, x_{s+1}).$$

Looking at the length of $\Phi_\alpha(\bar{x})$, we obtain the recurrence

$$l(k,s) \le f(k,s+1)(l(k,s+1)+9). \tag{5}$$

Set $g(x) = 2^x(x+9)$. A simple inductive argument shows that

$$f(k,s) \le 2^{g^{(k-s)}(9k^2)} \quad \text{and} \quad l(k,s) \le g^{(k-s)}(9k^2).$$

Define the two-parameter function $\mathit{Tower}(i,x)$ inductively on i by $\mathit{Tower}(0,x) = x$ and $\mathit{Tower}(i+1,x) = 2^{\mathit{Tower}(i,x)}$ for $i \ge 0$. This is a generalization of the old function: $\mathit{Tower}(i,1) = \mathit{Tower}(i)$. One can prove by induction on i that for any $x \ge 5$ and $i \ge 1$ we have

$$g^{(i)}(x) < \mathit{Tower}(i+1,x)/2. \tag{6}$$

Indeed, it is easy to check the validity of (6) for $i = 1$, while for $i \ge 2$ we have

$$g^{(i)}(x) < g(\mathit{Tower}(i,x)/2) < 2^{\mathit{Tower}(i,x)-1} = \mathit{Tower}(i+1,x)/2. \tag{7}$$

We have for all $k \ge 5$ that $9k^2 < \mathit{Tower}(\log^* k+1)$. This follows from $9k^2 < 2^k$ for $k \ge 10$ and can be checked by hand for $5 \le k \le 9$. Thus, for $k \ge 5$, we have by (6) that

$$l(k,0) \le g^{(k)}(9k^2) < \mathit{Tower}(k+1, 9k^2)/2 < \mathit{Tower}(k+1, 9k^2) < \mathit{Tower}(k+\log^* k+2).$$

Routine calculations (omitted) based on (5) and the exact initial values $f(2,2) = 3$, $f(3,3) = 15$, and $f(4,4) = 127$ give Part 2 of the lemma for $2 \le k \le 4$. □

Lemma 2.1.2 gives us a bound for the logical length of a graph in terms of its logical depth. It suffices to notice that each single graph G constitutes a $\equiv_k$-class for $k = D(G)$.

Theorem 2.2 (Pikhurko, Spencer, and Verbitsky [60])**.**

$$L(G) < \mathit{Tower}(D(G) + \log^* D(G) + 2).$$

In fact, [60, Theorem 10.1] states only that $L(G) < \mathit{Tower}(D(G) + \log^* D(G) + O(1))$. Here we went into the trouble of estimating the error term more precisely so that Lemma 2.1.2 and some of its consequences can be stated more neatly.

Lemma 2.1.1 gives the following result.

Theorem 2.3. *The number of graphs with logical depth at most k does not exceed $\mathit{Tower}(k+\log^* k+2)$.*

Notice two further consequences of Lemma 2.1.

Theorem 2.4.

1. *There are at most Tower*$(k+\log^* k+3)$ *pairwise inequivalent sentences about graphs of quantifier depth* k.
2. *Every sentence* Φ *about graphs of quantifier depth* k *has an equivalent sentence* Φ' *with the same quantifier depth and length less than* $3\,\mathit{Tower}(k+\log^* k+2)^2$.

Proof. Note that, if a sentence Φ has quantifier depth k, then the set of all graphs on which Φ is true is the union of some $\equiv_k$-classes. Therefore, there are $2^{f(k)}$ and no more pairwise inequivalent sentences of quantifier depth k, where $f(k)$ is the number of $\equiv_k$-classes. Part 1 now follows from Lemma 2.1.1. By the same reason every sentence Φ of quantifier depth k is equivalent to the disjunction of sentences defining some $\equiv_k$-classes. By Lemma 2.1.2, such disjunction does not need to be longer than $(f(k)+3)\,\mathit{Tower}(k+\log^* k+2)$. This proves Part 2. $\square$

2.3. **Distinguishability vs. definability.** Given two non-isomorphic graphs G and H, we define $D(G,H)$ (resp. $W(G,H)$) to be the minimum of $D(\Phi)$ (resp. $W(\Phi)$) over all sentences Φ distinguishing G from H. Thus, $D(G,H)>k$ if and only if $G\equiv_k H$. Obviously, $D(G,H)=D(H,G)$. Also, $D(G,H)\le D(G)$ and $W(G,H)\le W(G)$. It turns out that these inequalities are tight in the following sense.

Lemma 2.5.

1. $D(G)=\max_{H\not\cong G} D(G,H)$.
2. $W(G)=\max_{H\not\cong G} W(G,H)$.

Proof. 1. For each H non-isomorphic to G fix a sentence Φ_H that distinguishes G from H and has the minimum possible quantifier depth, i.e., $D(\Phi_H)=D(G,H)$. Consider the sentence $\Phi \stackrel{\text{def}}{=} \bigwedge_{H\not\cong G}\Phi_H$. It distinguishes G from each non-isomorphic H and has quantifier depth $\max_H D(\Phi_H)$. Therefore, $D(G)\le\max_H D(G,H)$ as wanted. An obvious drawback of this argument is that the above conjunction over H in Φ is actually infinite. However, we have $D(\Phi_H)\le D(G)$ and there are only finitely many pairwise inequivalent first-order sentences about graphs of bounded quantifier depth, see Theorem 2.4 above. Thus we can obtain a legitimate finite sentence defining G by removing from Φ duplicates up to logical equivalence.

2. Running the same argument, we have to "prune" the infinite conjunction $\bigwedge_{H\not\cong G}\Phi_H$, where $W(\Phi_H)=W(G,H)$. Here we encounter a complication because there are infinitely many inequivalent sentences of the same width. (Consider e.g. the sentences from Example 1.1.1.) However, Theorem 4.7.1 in Section 4 implies that for every H we can additionally require that the depth of Φ_H is at most, for example, n^n+n, where n is the order of G. Now we can proceed as in Part 1 of the lemma. $\square$

Lemma 2.5 stays true in any finite-variable logic, any logic with bounded number of quantifier alternations, the logic with counting quantifiers, and any hybrid thereof. We set $D^k(G,H)=\infty$ if k variables do not suffice to distinguish G from H.

3. Ehrenfeucht games

Let G and H be graphs with disjoint vertex sets. The *r-round k-pebble Ehrenfeucht game on G and H*, denoted by $\mathrm{EHR}_r^k(G,H)$, is played by two players, Spoiler and Duplicator, to whom we may refer as he and she respectively. The players have at their disposal k pairwise distinct pebbles $p_1,\ldots,p_k$, each given in duplicate. A *round* consists of a move of Spoiler followed by a move of Duplicator. At each move Spoiler takes a pebble, say p_i, selects one of the graphs G or H, and places p_i on a vertex of this graph. In response Duplicator should place the other copy of p_i on a vertex of the other graph. It is allowed to move previously placed pebbles to other vertices and place more than one pebble on the same vertex.

After each round of the game, for $1 \le i \le k$ let x_i (resp. y_i) denote the vertex of G (resp. H) occupied by p_i, irrespectively of who of the players placed the pebble on this vertex. If p_i is off the board at this moment, x_i and y_i are undefined. If after every of r rounds the component-wise correspondence $(x_1,\ldots,x_k)$ to $(y_1,\ldots,y_k)$ is a partial isomorphism from G to H, this is a win for Duplicator. Otherwise the winner is Spoiler. The following example should provide the reader with a hint for the solution of the exercise suggested in Example 1.1.1.

Example 3.1. *Spoiler wins $\mathrm{EHR}_4^3(P_n,H)$ if $\Delta(H)\ge 3$.* Assume that H contains no triangle because otherwise Spoiler wins by pebbling its vertices. Let v be a vertex in H of degree at least 3. Spoiler pebbles 3 neighbors of v. Duplicator should pebble 3 distinct pairwise non-adjacent vertices in P_n for otherwise she loses the game. The distance between any two vertices pebbled in H is equal to 2. Unlike to this, some two vertices pebbled in P_n (say, by pebbles p_1 and p_2) are at a larger distance. Spoiler moves p_3 to v. Duplicator is forced to violate the adjacency relation.

The particular case of $\mathrm{EHR}_r^k(G,H)$ in which the number of pebbles is the same as the number of rounds, i.e., $k=r$, deserves a special attention. In this case, the outcome of the game will not be affected if we prohibit moving pebbles from one vertex to another, that is, if we allow the players to play with each p_i exactly once, say, in the i-th round. We denote this variant of $\mathrm{EHR}_r^r(G,H)$ by $\mathrm{EHR}_r(G,H)$ and will mean it whenever the term *Ehrenfeucht game* is used with no specification.

Lemma 3.2. *Suppose that in the 3-pebble Ehrenfeucht game on (G,H) some two vertices $x,y\in V(G)$ at distance n were selected so that their counterparts $x',y'\in V(H)$ are at a strictly larger distance (possibly infinity). Then Spoiler can win in at most $\lceil \log n\rceil$ extra moves.*

Proof. Spoiler sets $u_1=x$, $u_2=y$, $v_1=x'$, $v_2=y'$, and places a pebble on the middle vertex u in a shortest path from u_1 to u_2 (or either of the two middle vertices if $d(u_1,u_2)$ is odd). Let $v\in V(H)$ be selected by Duplicator in response to u. By the triangle inequality, we have $d(u,u_m)<d(v,v_m)$ for $m=1$ or $m=2$. For such m Spoiler resets $u_1=u$, $u_2=u_m$, $v_1=v$, $v_2=v_m$ and applies the same strategy once again. In this way Spoiler ensures that $d(u_1,u_2)<d(v_1,v_2)$ in each round. Eventually, unless Duplicator loses earlier, $d(u_1,u_2)=1$ while $d(v_1,v_2)>1$, that is, Duplicator fails to preserve adjacency.

To estimate the number of moves made, notice that initially $d(u_1, u_2) = n$ and for each subsequent u_1, u_2 this distance becomes at most $f(d(u_1, u_2))$, where $f(\alpha) = (\alpha + 1)/2$. Therefore the number of moves does not exceed the minimum i such that $f^{(i)}(n) < 2$. As $(f^{(i)})^{-1}(\beta) = 2^i\beta - 2^i + 1$, the latter inequality is equivalent to $2^i \geq n$, which proves the bound. □

There is a rather clear connection between Spoiler's strategy designed in the proof of Lemma 3.2 and first-order formula $\Delta'_n(x, y)$ in (4). We will see that, in some strong sense, $\mathrm{EHR}_r(G, H)$ corresponds to first-order logic, while $\mathrm{EHR}^k_r(G, H)$ corresponds to its k-variable fragment. In fact, every logic has its own corresponding game.

In the *k-alternation* variant of $\mathrm{EHR}_r(G, H)$ Spoiler is allowed to switch from one graph to another at most k times during the game, i.e., in at most k rounds he can choose the graph other than that in the preceding round.

In the *counting version* of the game $\mathrm{EHR}^k_r(G, H)$ Spoiler can make a *counting move* consisting of two acts. First, he specifies a set of vertices A in one of the graphs. Duplicator has to respond with a set of vertices B in the other graph so that $|B| = |A|$ (if this is impossible, she immediately loses). Second, Spoiler places a pebble p_i on a vertex $b \in B$. In response Duplicator has to place the other copy of p_i on a vertex $a \in A$. It is clear that, any round with $|A| = 1$ is virtually the same as a round of the standard game.

There is a general analogy between strategies allowing Spoiler to win a game on G and H and first-order sentences distinguishing these graphs: the former can be converted into the latter and vice versa so that the duration of a game will be in correspondence to the quantifier depth and the number of pebbles will be in correspondence to the number of variables.

Theorem 3.3 (The Ehrenfeucht theorem and its variations)**.** *Let G and H be non-isomorphic graphs.*

1. (Ehrenfeucht [24], Fraïssé [28][3]) *$D(G, H)$ equals the minimum r such that Spoiler has a winning strategy in $\mathrm{EHR}_r(G, H)$.*
2. (Pezzoli [59]) *$D_k(G, H)$ equals the minimum r such that Spoiler has a winning strategy in the k-alternation game $\mathrm{EHR}_r(G, H)$.*
3. (Immerman [42], Poizat [65]) *$W(G, H)$ equals the minimum k such that Spoiler has a winning strategy in $\mathrm{EHR}^k_r(G, H)$ for some r.*
4. (Immerman [42], Poizat [65]) *$D^k(G, H)$ equals the minimum r such that Spoiler has a winning strategy in $\mathrm{EHR}^k_r(G, H)$.*
5. (Immerman and Lander [45]) *$W_\#(G, H)$ equals the minimum k such that Spoiler has a winning strategy in the counting version of $\mathrm{EHR}^k_r(G, H)$ for some r. Furthermore, if $k \leq W_\#(G, H)$, then $D^k_\#(G, H)$ equals the minimum r such that Spoiler has a winning strategy in the counting version of $\mathrm{EHR}^k_r(G, H)$.*

We refer the reader to [43, Theorem 6.10] for the proof of Parts 3–5. Part 1 follows from Part 4 in view of the facts that $D(G, H) = \min_k D^k(G, H)$ and that any

[3] It was Ehrenfeucht who formally introduced the game. Prior to Ehrenfeucht, Fraïssé obtained virtually the same result using an equivalent language of partial isomorphisms.

sentence Φ can be equivalently rewritten with the same quantifier depth $D(\Phi)$ and with use of at most $D(\Phi)$ variables.

In view of Lemma 2.5, the Ehrenfeucht theorem provides us with a powerful tool for estimating the logical depth and width of graphs. Consider, for instance, a path P_n. Example 3.1 and Lemma 3.2 are immediately translated into the upper bound $D^3(P_n) < \log n + 3$. On the other hand, a lower bound $D(P_n) \geq \log n - 2$ follows from the existence of a winning strategy for Duplicator in $\textsc{Ehr}_r(P_n, P_{n+1})$ whenever $r \leq \lfloor \log n \rfloor - 1$ (all details can be found in [71, Theorem 2.1.3]).

4. The Weisfeiler-Lehman algorithm

Graph Isomorphism is the problem of recognizing if two given graphs are isomorphic. The best known algorithm (Babai, Luks, and Zemlyachenko [8]) takes time $2^{O(\sqrt{n \log n})}$, where n denotes the number of vertices in the input graphs. Particular classes of graphs for which Graph Isomorphism is solvable more efficiently are therefore of considerable interest. Somewhat surprisingly, a number of important tractable cases are solvable by a combinatorially simple, uniform approach, namely the multidimensional Weisfeiler-Lehman algorithm. The efficiency of this method depends much on the logical complexity of input graphs.

For the history of this approach to the graph isomorphism problem we refer the reader to [4, 14]. We will abbreviate *k-dimensional Weisfeiler-Lehman algorithm* by *k-dim WL*. The 1-dim WL is commonly known as *canonical labeling* or *color refinement algorithm*. It proceeds in rounds; in each round a coloring of the vertices of input graphs G and H is defined, which refines the coloring of the previous round. The initial coloring C^0 is uniform, say, $C^0(u) = 1$ for all vertices $u \in V(G) \cup V(H)$. In the $(i+1)$st round, the color $C^{i+1}(u)$ is defined to be a pair consisting of the preceding color $C^{i-1}(u)$ and the multiset of colors $C^{i-1}(w)$ for all w adjacent to u. For example, $C^1(u) = C^1(v)$ iff u and v have the same degree. To keep the color encoding short, after each round the colors are renamed (we never need more than $2n$ color names[4]). As the coloring is refined in each round, it stabilizes after at most $2n$ rounds, that is, no further refinement occurs. The algorithm stops once this happens. If the multiset of colors of the vertices of G is distinct from the multiset of colors of the vertices of H, the algorithms reports that the graphs are not isomorphic; otherwise, it declares them to be isomorphic. Disappointingly, the output is not always correct. The algorithm may report false positives, for example, if both input graphs are regular with the same vertex degree.

Following the same idea, the k-dimensional version iteratively refines a coloring of $V(G)^k \cup V(H)^k$. The initial coloring of a k-tuple $\bar{u}$ is the isomorphism type of the subgraph induced by the vertices in $\bar{u}$ (viewed as a labeled graph where each vertex is labeled by the positions in the tuple where it occurs). Loosely speaking, the refinement step takes into account the colors of all neighbors of $\bar{u}$ in the Hamming metric. Color stabilization is surely reached in $r < 2n^k$ rounds and, thus, the algorithm terminates in polynomial time for fixed k.

[4]We do not need even more than n because appearance of the $(n+1)$th color indicates non-isomorphism.

Let us give a careful description of the k-dim WL for $k \geq 2$. Given an ordered k-tuple of vertices $\bar{u} = (u_1, \ldots, u_k) \in V(G)^k$, we define the *isomorphism type* of $\bar{u}$ to be the pair

$$\mathrm{tp}(\bar{u}) = \Big(\left\{ (i,j) \in [k]^2 : u_i = u_j \right\}, \left\{ (i,j) \in [k]^2 : \{u_i, u_j\} \in E(G) \right\} \Big),$$

where $[k]$ denotes the set $\{1, \ldots, k\}$. If $w \in V(G)$ and $i \leq k$, we let $\bar{u}^{i,w}$ denote the result of substituting w in place of u_i in $\bar{u}$.

The *r-round k-dim WL* takes as an input two graphs G and H and purports to decide if $G \cong H$. The algorithm performs the following operations with the set $V(G)^k \cup V(H)^k$.

Initial coloring. The algorithm assigns each $\bar{u} \in V(G)^k \cup V(H)^k$ color $C^{k,0}(\bar{u}) = \mathrm{tp}(\bar{u})$ (in a suitable encoding).

Color refinement step. In the i-th round each $\bar{u} \in V(G)^k$ is assigned color

$$C^{k,i}(\bar{u}) = \Big(C^{k,i-1}(\bar{u}), \{\!\!\{ \left(C^{k,i-1}(\bar{u}^{1,w}), \ldots, C^{k,i-1}(\bar{u}^{k,w}) \right) : w \in V(G) \}\!\!\} \Big)$$

and similarly with each $\bar{u} \in V(H)^k$.

Here $\{\!\!\{ \ldots \}\!\!\}$ denotes a multiset. In a weaker *count-free* version of the algorithm, this notation will be interpreted as a set. Let

$$C^{k,r}(G) = \{\!\!\{ C^{k,r}(\bar{u}) : \bar{u} \in V(G)^k \}\!\!\} .$$

Computing an output. The algorithm reports that $G \not\cong H$ if

$$C^{k,r}(G) \neq C^{k,r}(H) \tag{8}$$

and that $G \cong H$ otherwise.

In the above description we skipped an important implementation detail. In order to prevent increasing the length of $C^{k,i}(\bar{u})$ at the exponential rate, we arrange colors of all k-tuples of $V(G)^k \cup V(H)^k$ in the lexicographic order and replace each color with its number before every refinement step.

Furthermore, let

$$\mathit{diag}\, C^{k,r}(G) = \{\!\!\{ C^{k,r}(u^k) : u \in V(G) \}\!\!\} ,$$

where u^k denotes the k-tuple $(u, \ldots, u)$.

Lemma 4.1. *In both the standard and the count-free versions of the k-dim WL, inequality*

$$\mathit{diag}\, C^{k,r}(G) \neq \mathit{diag}\, C^{k,r}(H) \tag{9}$$

implies (8), which in its turn implies

$$\mathit{diag}\, C^{k,r+k-1}(G) \neq \mathit{diag}\, C^{k,r+k-1}(H). \tag{10}$$

Proof. Consider the standard version; the analysis of the count-free case is similar (and even simpler). Note that k-tuples with different equality types never have the same color. Therefore, $C^{k,r}(G)$ and $C^{k,r}(H)$ are different iff their restrictions to some equality type are different. This proves the first implication.

On the other hand, suppose that (8) holds. Let E be an equality type on which $C^{k,r}(G)$ and $C^{k,r}(H)$ differ. Note that each $\bar{u}$ in E contributes color $C^{k,r}(\bar{u})$ (a certain number of times) to color $C^{k,r+k-1}(a^k)$. Moreover, the sum of the contributions over all vertices a is the same for every $\bar{u} \in E$. It follows that, if a color has different multiplicities in $C^{k,r}(G)$ and $C^{k,r}(H)$, its "traces" occur different number of times in $diag\, C^{k,r+k-1}(G)$ and $diag\, C^{k,r+k-1}(H)$, and hence these multisets are distinct. □

As it is easily seen, if ϕ is an isomorphism from G to H, then for all k, i, and $\bar{u} \in V(G)^k$ we have $C^{k,i}(\bar{u}) = C^{k,i}(\phi(\bar{u}))$. This shows that for isomorphic input graphs the output is always correct. If input graphs are non-isomorphic and the dimension k is not big enough, the algorithm can erroneously report isomorphism. A criterion for the optimal choice of the dimension is obtained by Cai, Fürer, and Immerman [14], who discovered a connection between the Weisfeiler-Lehman algorithm and the logical complexity of graphs via the Ehrenfeucht game (for the color refinement algorithm this was done by Immerman and Lander [45]). The success of the standard version of the algorithm depends on distinguishability of the input graphs in the logic with counting quantifiers, while the count-free version is in the same way related to the standard first-order logic.

Referring to the k-dim WL below, we will **always** assume $k \geq 1$ for the standard version of the algorithm and $k \geq 2$ for its count-free version (we can exclude the case of $k = 1$, whose analysis differs by some details, as the count-free 1-dim WL is of no interest: note that it is unable to distinguish between two graphs of order n without isolated vertices).

Given numbers r, l, and $k \leq l$, graphs G, H, and k-tuples $\bar{u} \in V(G)^k$, $\bar{v} \in V(H)^k$, we use notation $\mathrm{EHR}_r^l(G, \bar{u}, H, \bar{v})$ to denote the r-round l-pebble Ehrenfeucht game on G and H with initial configuration $(\bar{u}, \bar{v})$, that is, the game starts on the board with k already pebbled pairs (u_i, v_i). The following lemma is a key element of our analysis.

Lemma 4.2 (Cai, Fürer, and Immerman [14]). *Let $\bar{u} \in V(G)^k$ and $\bar{v} \in V(H)^k$.*

1. *Equality*
$$C^{k,r}(\bar{u}) = C^{k,r}(\bar{v}) \tag{11}$$
holds for (the standard version of) the k-dim WL iff Duplicator has a winning strategy in the counting version of $\mathrm{EHR}_r^{k+1}(G, \bar{u}, H, \bar{v})$.

2. *Equality (11) holds for the count-free version of the k-dim WL iff Duplicator has a winning strategy in (the standard version of) $\mathrm{EHR}_r^{k+1}(G, \bar{u}, H, \bar{v})$.*

Proof. We prove only Part 2 (Part 1 is proved in detail in [14, Theorem 5.2]). We proceed by induction on r. The base case $r = 0$ is straightforward by the definitions of the initial coloring and the game. Assume that the proposition is true for $r - 1$ rounds.

Let x_i and y_i denote the vertices in G and H respectively marked by the i-th pebble pair. Assume (11) and consider the Ehrenfeucht game on G, H with initial configuration $(x_1, \ldots, x_k) = \bar{u}$ and $(y_1, \ldots, y_k) = \bar{v}$. First of all, this configuration is non-losing for Duplicator since (11) implies that $\mathrm{tp}(\bar{u}) = \mathrm{tp}(\bar{v})$. Further, Duplicator can survive in the first round. Indeed, assume that Spoiler in this round selects a vertex a in one of the graphs, say in G. Then Duplicator selects a vertex b in the other graph H so that $C^{k,r-1}(\bar{u}^{i,a}) = C^{k,r-1}(\bar{v}^{i,b})$ for all $i \leq k$. In particular, $\mathrm{tp}(\bar{u}^{i,a}) = \mathrm{tp}(\bar{v}^{i,b})$ for all $i \leq k$. Along with $\mathrm{tp}(\bar{u}) = \mathrm{tp}(\bar{v})$, this implies that $\mathrm{tp}(\bar{u}, a) = \mathrm{tp}(\bar{v}, b)$.

Assume now that in the second round Spoiler removes j-th pebble, $j \le k$. Then Duplicator's task in the rest of the game is essentially to win $\mathrm{EHR}^{k+1}_{r-1}(G, \bar{u}^{j,a}, H, \bar{v}^{j,b})$. Since $C^{k,r-1}(\bar{u}^{j,a}) = C^{k,r-1}(\bar{v}^{j,a})$, Duplicator succeeds by the induction assumption.

Assume now that (11) is false. It follows that $C^{k,r-1}(\bar{u}) \ne C^{k,r-1}(\bar{v})$ (then Spoiler has a winning strategy by the induction assumption) or there is a vertex a in one of the graphs, say in G, such that for every b in the other graph H we have $C^{k,r-1}(\bar{u}^{j,a}) \ne C^{k,r-1}(\bar{u}^{j,b})$ for some $j = j(b)$. In the latter case Spoiler in his first move places the $(k+1)$-th pebble on a. Let b be the vertex selected in response by Duplicator. In the second move Spoiler will remove the $j(b)$-th pebble, which implies that the players essentially play $\mathrm{EHR}^{k+1}_{r-1}(G, \bar{u}^{j,a}, H, \bar{v}^{j,b})$ from now on. By the induction assumption, Spoiler wins. □

Lemma 4.3. *Equality* $\mathit{diag}\, C^{k,r}(G) = \mathit{diag}\, C^{k,r}(H)$ *is true for the standard (resp. count-free) version of the k-dim WL iff Duplicator has a winning strategy in the counting (resp. standard) version of* $\mathrm{EHR}^{k+1}_{r+1}(G, H)$.

Proof. We consider the standard version of the algorithm; the proof for the count-free version is very similar. If the multisets $\mathit{diag}\, C^{k,r}(G)$ and $\mathit{diag}\, C^{k,r}(H)$ are not equal, Spoiler has a winning strategy in the counting game $\mathrm{EHR}^{k+1}_{r+1}(G, H)$. In the first round he makes a counting move that forces pebbling $a \in V(G)$ and $b \in V(H)$ so that $C^{k,r}(a^k) \ne C^{k,r}(b^k)$. The remainder of the game is equivalent to the counting game $\mathrm{EHR}^{k+1}_{r}(G, a^k, H, b^k)$, where Spoiler has a winning strategy by Lemma 4.2.

If the multisets $\mathit{diag}\, C^{k,r}(G)$ and $\mathit{diag}\, C^{k,r}(H)$ are equal, Duplicator is able to play the first round so that $C^{k,r}(a^k) = C^{k,r}(b^k)$ for the pebbled vertices a and b. She wins the remaining game again by Lemma 4.2. □

We say that the r-round k-dim WL *works correctly for a graph G* if its output is correct on all input pairs (G, H) (here H may have any order, not necessary the same as G).

Theorem 4.4. *The r-round k-dim WL works correctly for G if*

$$k \ge W_{\#}(G) - 1 \quad \text{and} \quad r \ge D^{k+1}_{\#}(G) - 1$$

and only if

$$k \ge W_{\#}(G) - 1 \quad \text{and} \quad r \ge D^{k+1}_{\#}(G) - k.$$

The same holds true for the count-free r-round k-dim WL and the standard logic (without counting).

Proof. If $G \cong H$, the output is correct in any case. Suppose that $G \not\cong H$. By Lemma 4.1, inequality (9) is a sufficient condition for the output being correct while (10) is a necessary condition for this. The theorem now follows from Lemma 4.3, the Ehrenfeucht theorem (Theorem 3.3.4,5), and Lemma 2.5.1 along with its counting version. □

By Theorem 4.4, $k \ge W_{\#}(G) - 1$ is both a sufficient and a necessary condition for a successful work of the k-dim WL on all inputs (G, H). As we already discussed, the number of rounds can be taken $r = O(n^k)$. Therefore, Graph Isomorphism is solvable in polynomial time for any class of graphs C with $W_{\#}(G) = O(1)$ for all

$G \in C$. This applies to any class of graphs embeddable into a fixed surface and any class of graphs with bounded treewidth (see Section 5.1).

Sometimes the Weisfeiler-Lehman algorithm gives us even better result, namely the solvability of the isomorphism problem by a parallel algorithm in polylogarithmic time. The concept of polylogarithmic parallel time is captured by the complexity class NC and its refinements:

$$\mathrm{NC} = \bigcup_i \mathrm{NC}^i \text{ and } \mathrm{NC}^i \subseteq \mathrm{AC}^i \subseteq \mathrm{TC}^i \subseteq \mathrm{NC}^{i+1},$$

where NC^i consists of functions computable by circuits of polynomial size and depth $O(\log^i n)$, AC^i is an analog for circuits with unbounded fan-in, and TC^i is an extension of AC^i allowing threshold gates. As it is well known [47], AC^i consists of exactly those functions computable by a CRCW PRAM with polynomially many processors in time $O(\log^i n)$. Grohe and Verbitsky [40] point out that the r-round k-dim WL (resp. its count-free version) is implementable in TC^1 (resp. AC^1) as long as $k = O(1)$ and $r = O(\log n)$. If combined with Theorem 4.4, this gives us the following result.

Theorem 4.5. *Let $k \geq 2$ be a constant.*

1. *Let C be a class of graphs G with $D^k_{\#}(G) = O(\log n)$. Then Graph Isomorphism for C is solvable in* TC^1.
2. *Let C be a class of graphs G with $D^k(G) = O(\log n)$. Then Graph Isomorphism for C is solvable in* AC^1.

We will see applications of Theorem 4.5 in Section 5.1.

Suppose that $k \geq W(G) - 1$ and that we do not know a priori any bounds for $D^{k+1}(G)$. How large has r to be taken in order to ensure that the r-round k-dim WL works correctly for G? An answer is given by an important concept of *color stabilization*, that was already discussed in the beginning of this section. We will regard $C^{k,r}$ as a partition of $V(G)^k \cup V(H)^k$. Let R be the minimum number for which $C^{k,R} = C^{k,R-1}$. Of course, it is enough to check the condition (8) for $r = R$; it cannot change for bigger r. Since each $C^{k,r}$ is a refinement of $C^{k,r-1}$, we have $R \leq v(G)^k + v(H)^k$. In fact, we are able to prove a bit more delicate claim: The Weisfeiler-Lehman algorithm can be terminated as soon as $C^{k,r}$ stabilizes at least within $V(G)^k$.

To make this more precise, we introduce some notation. Denote the restriction of the partition $C^{k,r}$ to $V(G)^k$ by $C^{k,r}_G$. Let $\mathit{Stab}^k(G)$ be the smallest number s such that $C^{k,s}_G = C^{k,s-1}_G$. Note that $\mathit{Stab}^k(G)$ is an individual combinatorial parameter of a graph G, not depending on H (we may think that the k-dim WL is run on a single graph G, which is actually a quite meaningful *canonization mode* of the algorithm).

We now state practical termination rules for the k-dim WL.

Rule 1: Once $C^{k,r}(G) \neq C^{k,r}(H)$, terminate and report non-isomorphism.

Rule 2: Once $r = \mathit{Stab}^k(G)$ and $C^{k,r}(G) = C^{k,r}(H)$, terminate and report isomorphism.

Let us argue that these rules are sound for both versions of the algorithm. Suppose that Rule 2 is invoked. Thus $C^{k,r}_G = C^{k,r-1}_G$ and $C^{k,r}(G) = C^{k,r}(H)$. By the latter equality we also have $C^{k,r-1}(G) = C^{k,r-1}(H)$. It follows that in the r-th round the algorithm achieves a proper color refinement on neither G nor H. Thus, the partition $C^{k,r}$ has been stabilized on $V(G)^k \cup V(H)^k$ and the soundness of Rule 2 follows.

Theorem 4.6.

1. *The r-round k-dim WL recognizes non-isomorphism of G and H if*
$$k \geq W_\#(G,H) - 1 \quad \textit{and} \quad r \geq \mathit{Stab}^k(G).$$
2. *The r-round k-dim WL works correctly for G if*
$$k \geq W_\#(G) - 1 \quad \textit{and} \quad r \geq \mathit{Stab}^k(G).$$
3. *Both claims hold true for the count-free version of the algorithm and the standard logic (with no counting).*

We have seen that good bounds for the logical complexity of graphs imply efficiency of the Weisfeiler-Lehman algorithm on these graphs. Now we will get a couple of noteworthy facts on the logical complexity as a consequence of our analysis of the algorithm.

Theorem 4.7. *Let G be a graph of order n.*

1. *If G is distinguishable from another graph H in the ℓ-variable logic, then $D^\ell(G,H) \leq n^{\ell-1} + \ell - 2$.*
2. *If G is definable in the ℓ-variable logic, then $D^\ell(G) \leq n^{\ell-1} + \ell - 2$.*

Proof. Let $k = \ell - 1$. Comparing the sufficient conditions for the correctness of the r-round k-dim WL given by Theorem 4.6 and the necessary conditions given by Theorem 4.4, we have $D^{k+1}(G,H) \leq \mathit{Stab}^k(G) + k$ provided $k \geq W(G,H) - 1$ and $D^{k+1}(G) \leq \mathit{Stab}^k(G) + k$ provided $k \geq W(G) - 1$. For the former claim we need also the fact, actually established in the proof of Theorem 4.4, that the count-free r-round k-dim WL is able to recognize non-isomorphism of G and H only if $k \geq W(G,H) - 1$ and $r \geq D^{k+1}(G,H) - k$. It remains to notice that $\mathit{Stab}^k(G) \leq n^k - 1$. □

A somewhat weaker bound $D^\ell(G) \leq n^\ell + \ell + 1$ follows from the work of Dawar, Lindell, and Weinstein [19, Corollary 4].

5. Worst case bounds

5.1. **Classes of graphs.** Here we overview known bounds for the logical depth and width for natural classes of graphs. Several interesting definability effects can be observed even when we focus on so simple graphs as trees. This class is considered at the beginning of this section (and will be further discussed in Sections 6 and 7). We will see that many results about trees admit generalization to graphs with bounded treewidth. We further consider planar graphs. Then we briefly discuss more general cases of graphs embeddable into a fixed surface and graphs with an excluded minor, as well as a few sporadic results on other classes.

5.1.1. *Trees.* The following result is based on Edmonds' algorithm, that dates back to the sixties (see, e.g., [15]), and its logical interpretation is due to Immerman and Lander [45].

Theorem 5.1.

1. *The color refinement algorithm succeeds in recognizing isomorphism of trees. Consequently, $W_{\#}(T, T') \le 2$ for every two non-isomorphic trees T and T'.*
2. *$W_{\#}(T) \le 2$ for every tree T.*

Proof. 1. As in Section 4, let C^r denote the coloring appearing after the r-th refinement. Let $N_r(v)$ denote the set of all vertices at the distance at most r from a vertex v. It is not hard to see that, if v is an arbitrary vertex in a tree T, then the subtree spanned by $N_r(v)$ is, up to isomorphism, reconstructible from $C^r(v)$. Let v and v' be arbitrary vertices in trees T and T'. If $T \not\cong T'$, we have $C^r(v) \ne C^r(v')$ at latest for r one greater than the smaller of the eccentricities of v and v'. Therefore, the color refinement algorithm distinguishes between any two non-isomorphic trees. The second statement of Part 1 follows by Lemma 4.2.1 and Theorem 3.3.5.

2. To obtain the desired definability result, we use the equality $W_{\#}(T) = \max_{H \not\cong T} W_{\#}(T, H)$, which is an analog of Lemma 2.5.2 (with a much simpler proof as graphs of different orders are distinguishable with a single counting quantification). Thus, it suffices to prove that $W_{\#}(T, H) \le 2$ whenever $H \not\cong T$. Suppose that H is not a tree for otherwise we are done by Part 1. Also, as it was just mentioned, we can suppose that both T and H have n vertices.

Assume first that H has a connected component T' which is a tree. Note that $T' \not\cong T$ because T' has less than n vertices. Let $v \in V(T)$ and $v' \in V(T')$. Run the color refinement algorithm on input (T, H). As in the proof of Part 1 we have $C^n(v) \ne C^n(v')$ because the coloring C^n on T' is the same as if the algorithm was run on T' instead of H. Therefore, T and H are distinguishable with 2 variables in the counting logic.

If none of the connected components of H is a tree, then H has at least n edges. Since T has exactly $n - 1$ edges, H and T have distinct multisets of vertex degrees and, hence, are distinguishable by a sentence with 2 counting quantifiers. □

The proof of Theorem 5.1 gives us only a linear upper bound $D^2_{\#}(T) = O(n)$ for a tree of order n. We can get a speed-up if we allow more variables.

Theorem 5.2. *For every tree T on n vertices we have*

$$D^3_{\#}(T) < 3 \log n.$$

Proof. By an analog of Lemma 2.5.1 for the counting logic and Theorem 3.3.5, we have to show that Spoiler is able to win the counting game $\mathrm{EHR}^3_r(T, T')$ with some $r < 3 \log n$ for any graph T' non-isomorphic to T. Suppose that T' has the same order n. If T' is disconnected, Spoiler wins (even without counting moves) by Lemma 3.2. If T' is connected and has a cycle, then T and T' have distinct multisets of vertex degrees. Therefore, we will suppose that T' is a tree too.

Every tree T has a single-vertex *separator*, that is, a vertex v such that no branch of $T - v$ has more than $n/2$ vertices; see, e.g., Ore [58, Chapter 4.2]. The idea of

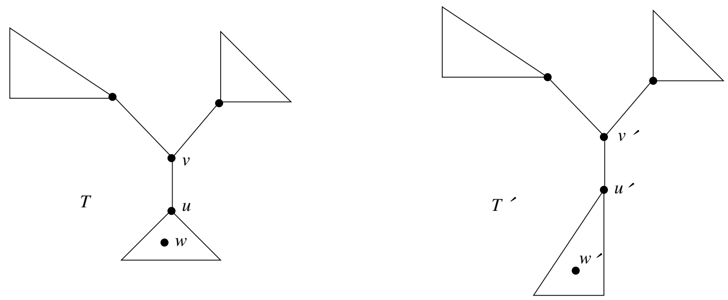

FIGURE 1. A separator strategy of Spoiler.

Spoiler's strategy is to pebble such a vertex and to force further play on some non-isomorphic branches of T and T', where the same strategy can be applied recursively.

Thus, in the first round Spoiler pebbles a separator v in T and Duplicator responds with a vertex v' somewhere in T'. The component of $T - v$ containing a neighbor u of v will be denoted by T_{vu} and considered a rooted tree with the root at u. A similar notation will apply also to T'. In the second round Spoiler makes a counting move and ensures that $u \in N(v)$ and $u' \in N(v')$ are pebbled so that the rooted trees T_{vu} and $T'_{v'u'}$ are non-isomorphic, see Fig. 1. The next goal of Spoiler is to force pebbling adjacent vertices v_1 and u_1 in T_{vu} and adjacent vertices v'_1 and u'_1 in $T'_{v'u'}$ so that $T_{v_1u_1} \not\cong T'_{v'_1u'_1}$, $V(T_{v_1u_1}) \subset V(T_{vu})$, and $v(T_{v_1u_1}) \le v(T_{vu})/2$. Once this is done, the same will be repeated recursively.

To make the transition from T_{vu} to $T_{v_1u_1}$, Spoiler follows three rules.

Rule 1. If T_{vu} has a branch T_{ux} for some $x \in N(u) \setminus \{v\}$ such that $v(T_{ux}) \le v(T_{vu})/2$ and the number of branches isomorphic to T_{ux} is different for T_{vu} and $T'_{v'u'}$, then Spoiler makes a counting move and forces pebbling such x and $x' \in N(u') \setminus \{v'\}$ so that $T_{ux} \not\cong T'_{u'x'}$. The latter two branches will serve as $T_{v_1u_1}$ and $T'_{v'_1u'_1}$. If no such branch is available, Spoiler pebbles a separator w of T_{vu}. Note that Duplicator is forced to respond with a vertex w' in $T'_{v'u'}$. Otherwise we would have $dist(w, u) = dist(w, v) - 1$ while $dist(w', u') = dist(w', v') + 1$. Therefore, some distances among the three pebbled vertices would be different in T and in T' and Spoiler could win in less than $\log v(T_{vu}) + 1$ moves by Lemma 3.2.

Rule 2. If T differs from T' by some branch T_{wx} (having a different number of occurrences in $T' - w'$) that does not contain u, Spoiler makes a counting move with the pebble released from v and forces pebbling such x in T and some x' in T' so that $T_{wx} \not\cong T'_{w'x'}$. These branches will serve as $T_{v_1u_1}$ and $T'_{v'_1u'_1}$. (It is possible that $T'_{w'x'}$ contains u' or T_{wx} contains u but then the distances among u, w, x are not all equal to the distances among u', w', x' and Spoiler quickly wins.)

Rule 3. Denote the branch of $T_{vu} - w$ containing u by $T_{w,u}$ (and similarly for T'). If Rule 2 is not applicable, then $T_{w,u}$ and $T'_{w',u'}$ are non-isomorphic (where an isomorphism would need to respect two pairs of designated vertices, namely u and u' as well as the neighbors of w and w'). Assume the harder subcase that $dist(u, w) = dist(u', w')$. When Spoiler pebbles a vertex y on the path from u to w by moving the pebble from v, Duplicator is forced to pebble the corresponding

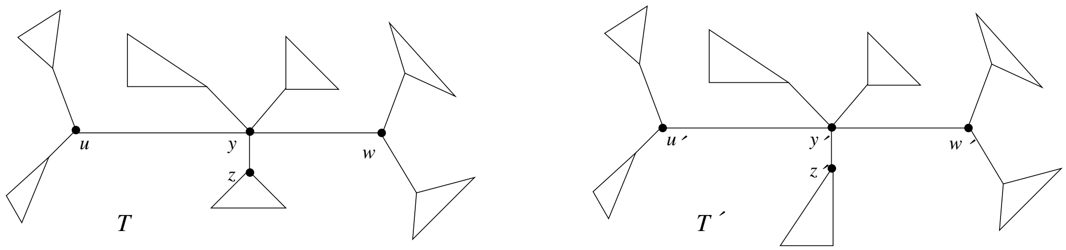

FIGURE 2. Rule 3 invoked.

vertex y' on the path from u' to w'. It is easy to see that an $y \neq w$ can be chosen so that $T - y$ and $T' - y'$ differ by branches containing neither u and u' nor w and w'. Let Spoiler pebble such y as close to w as possible. Note that $y \neq u$ because otherwise Rule 1 was applicable. Now Spoiler can make a counting move with the pebble released from w to force pebbling z and z' for which

$$T_{yz} \not\cong T'_{y'z'}, \tag{12}$$

see Fig. 2. This complies with the goal of finding new $T_{v_1u_1}$ and $T'_{v'_1u'_1}$ because

$$v(T_{yz}) < v(T_{w,u}) \le v(T_{vu})/2. \tag{13}$$

In fact, Duplicator could try to prevent the fulfillment of (12) by forcing a choice of z' such that $T'_{y'z'}$ would contain either u' or w'. In the former case Spoiler could win by using differences between the distances among u, y, z and among u', y', z'. In the latter case (12) would anyway be true because T_{yz} and $T'_{y'z'}$ would have different orders. Indeed, since Rule 2 was not applicable, the choice of y ensures that the branches of $T - y$ and $T' - y'$ containing w and w' are isomorphic. Thus, we would have $v(T'_{y'z'}) = v(T_{y,w}) \ge v(T_{vu})/2$ (where $T_{y,w}$ denotes the branch of $T - y$ containing w) while $v(T_{yz})$ is strictly smaller by (13).

Given T_{vu}, Spoiler finds a new distinguishing branch $T_{v_1u_1}$ in 3 rounds in the worst case. Also, 2 rounds suffice to win the game once the current subtree T_{vu} has at most 4 vertices. The number of transitions from the initial branch of order at most $\lceil n/2 \rceil$ to one with at most 4 vertices is bounded by $\log\lceil n/2 \rceil - 2$ because $v(T_{vu})$ becomes twice smaller each time. Routine calculations (and Lemma 3.2) imply the desired bound on the length of the game. □

The definability of trees in a finite-variable counting logic within logarithmic quantifier depth can also be derived from a work by Etessami and Immerman [26], which also implies that counting quantifiers are here not needed as long as the maximum vertex degree is bounded by a constant.

Curiously, Theorem 5.2 sheds some new light on the history of isomorphism testing for trees. The first record of this history was made by Edmonds, who showed that the problem is solvable in linear time (see Theorem 5.1). Ruzzo [70] found an AC^1 algorithm under the condition that the vertex degrees of input trees are at most logarithmic in the number of vertices. Miller and Reif [54] established an AC^1 upper bound unconditionally. They wrote [54, page 1128]: "No polylogarithmic parallel algorithm was previously known for isomorphism of unbounded-degree trees."

However, the 2-dimensional Weisfeiler-Lehman algorithm has been discussed in the literature at least since 1968 (e.g., [76]) and, as we now see by combining Theorem 5.2 with Theorem 4.5.1, this algorithm does the job for arbitrary trees in NC^2, i.e., in parallel time $O(\log^2 n)$!

To complete this historical overview, we have to mention a result by Lindell [52] who showed that isomorphism of trees is recognizable in logarithmic space. Though Lindell's result is best possible (see Jenner et al. [46]), the solvability of the problem by so simple and natural procedure as the Weisfeiler-Lehman algorithm still remains a noteworthy fact.

Note that $D^2_\#(P_n) = \frac{n}{2} - O(1)$ (it is not hard to see that the color refinement algorithm requires at least $\frac{n}{2} - O(1)$ rounds to distinguish between P_n and the disjoint union of P_{n-3} and C_3). Thus, Theorem 5.2 shows a jump from linear to logarithmic quantifier depth under increase the number of variables just by 1. Such width-depth trade-offs were observed and studied by Fürer [29].

Theorem 5.1 says that 2 variables and counting quantifiers suffice to define any tree. Moreover, we could well manage without counting quantifiers but then we would need to have $\Delta(T)+1$ variables. A simple example of a star, where $W(K_{1,m}) = m+1$, shows that a smaller number is not enough. The following bound is a variant of a result by Immerman and Kozen [44], who consider definability of trees represented by an asymmetric child-parent relation between vertices.

Theorem 5.3. $W(T) \le \Delta(T)+1$ *for any tree* T *with the exceptions of* $T \in \{P_1, P_2\}$.

The logical depth of a tree can be bounded in terms of the maximum degree and the order.

Theorem 5.4 (Bohman et al. [11])**.**

1. *For every tree* T *of order* n *with maximum vertex degree* $\Delta(T) \ge 9$ *we have*
$$D(T) \le \left(\frac{\Delta(T)}{2\log(\Delta(T)/2)} + 3\right)\log n + \frac{3\Delta(T)}{2} + O(1).$$
2. *Let* $D(n,d)$ *be the maximum of* $D(T)$ *over all trees with* n *vertices and maximum degree at most* $d = d(n)$. *If both* d *and* $\log n/\log d$ *tend to infinity, then*
$$D(n,d) = \left(\frac{1}{2} + o(1)\right) d\,\frac{\log n}{\log d}.$$

The upper bounds on $D(T)$ comes from Spoiler's strategy similar to that of the proof of Theorem 5.2, that is, Spoiler pebbles a separator v of the given tree T and then tries to restrict the game to one of the components of $T - v$. Informally speaking, the worst case scenario for Spoiler is when $T - v$ has d components of order about n/d of two different isomorphism types, each occurring half of the time. Then Spoiler may need around $d/2$ extra moves to restrict game to a component of $T - v$ (if the components of the counterpart $T' - v'$ are of these two types but with different multiplicities). Thus, roughly, Spoiler "reduces" the order by factor d using $d/2$ moves, which gives the heuristic for the bound of Theorem 5.4.2. The optimality of this bound is given by a recursive construction of a tree T (and another tree $T' \not\cong T$), where at each recursion step we glue together about $d/2$ trees of two different isomorphism types at a common root.

5.1.2. *Graphs of bounded treewidth.* Informally speaking, the *treewidth* of a graph tells us to which extent the graph is representable as a tree-like structure. This concept appeared in the Robertson-Seymour theory and, aside of its theoretical importance, found a lot of applications in design of algorithms on graphs. We do not go into any detail here, referring instead to the books [21] and [22] that may serve as introductions to, respectively, the structural theory of graphs and the algorithmic applications.

It happens quite often that techniques applicable to trees can be extended to graphs whose treewidth is bounded by a constant. In particular, this is true for the definability parameters.

Theorem 5.5.

1. (Grohe and Marino [38]) *If a graph G has treewidth k, then $W_{\#}(G) \le k+2$.*

2. (Grohe and Verbitsky [40]) *If a graph G on n vertices has treewidth k, then*

$$D_{\#}^{4k+4}(G) < 2(k+1)\log n + 8k + 9.$$

Consequently, isomorphism of graphs whose treewidth does not exceed k is recognizable by the $(4k+3)$-dimensional Weisfeiler-Lehman algorithm in TC^1.

The last claim in the theorem follows a general paradigm provided by Theorem 4.5.1: A low quantifier depth implies solvability of the isomorphism problem in NC. Prior to [40], for graphs with bounded treewidth only polynomial-time isomorphism test of Bodlaender [10] was known. Very recently Das, Torán, and Wagner [17] put the problem in the complexity class LOGCFL.

Like Theorem 5.2, the proof of Theorem 5.5.2 is based on separator techniques. In general, a set $X \subset V(G)$ will be called a *separator* for graph G if any component of $G \setminus X$ has at most $n/2$ vertices. It is well known [69] that all graphs of treewidth k have separators of size $k+1$.

5.1.3. *Planar graphs.* The separator techniques in the study of logical complexity of graphs were introduced by Cai, Fürer, and Immerman [14], who derived a bound $W_{\#}(G) = O(\sqrt{n})$ for planar graphs from the known fact [53] that every planar graph of order n has a separator of size $O(\sqrt{n})$. In fact, this result is a particular case of Theorem 5.5.1 because planar graphs have treewidth bounded by $5\sqrt{5}\,n$; see [2, Proposition 4.5]. Later Grohe [33] proved that $W_{\#}(G)$ for all planar G is actually bounded by a constant.

Without counting quantifiers we cannot have any nontrivial upper bound for the logical depth in terms of the order of a graph as long as a class under consideration contains all trees. However, some natural classes of planar graphs admit such bounds. A plane drawing of a graph is called *outerplanar* if all the vertices lie on the boundary of the outer face. *Outerplanar graphs* are those planar graphs having an outerplanar drawing. The treewidth of any outerplanar graph is at most 2. As it is well known (see, e.g., [41]), any outerplanar graph is representable as a tree of its biconnected components. Note also that an outerplanar graph is biconnected iff it has a Hamiltonian cycle and that such a graph can be geometrically viewed as a dissection of a convex polygon.

Theorem 5.6 (Verbitsky [74, 75])**.**

1. *If G is a biconnected outerplanar graph of order n, then $D(G) < 22\log n+9$.*
2. *For a 3-connected planar graph G of order n we have $D^{15}(G) < 11\log n+45$.*

Part 2 shows another case when Theorem 4.5 is applicable. It gives an AC^1 isomorphism test for 3-connected planar graphs and, by a known reduction of Miller and Reif [54], for the whole class of planar graphs. This complexity bound for the planar graph isomorphism is not new; it follows from the AC^1 isomorphism test for embeddings designed in [54] and the AC^1 embedding algorithm in [67]. As a possible advantage of the Weisfeiler-Lehman approach, note that it is combinatorially much simpler and more direct. In particular, we do not need any embedding procedure here. The best possible complexity bound for the planar graph isomorphism is recently obtained by Datta et al. [18] who design a logarithmic-space algorithm for this problem.

Theorem 5.6.1 is proved in [74] and is based on the existence of a 2-vertex separator in any outerplanar graph. The possibility to avoid counting quantifiers relies on certain rigidity of biconnected outerplanar graphs. The latter is related to the following geometric fact: Any such graph has a unique, up to homeomorphism, outerplanar drawing.

The case of 3-connected planar graphs is much more complicated because the smallest separators in such graphs can have about $\sqrt{n}$ vertices (such examples can be obtained by adding a few edges to the grid graph $P_m \times P_m$). The proof of Theorem 5.6.2 in [75] exploits a strong rigidity property of 3-connected planar graphs: By the Whitney theorem [77], they have a unique, up to homeomorphism, embedding into the sphere. An embedding can be represented as a purely combinatorial structure, called a *rotation system* (see [55]), to which one can extend the concepts of definability, isomorphism, the Ehrenfeucht game etc. Defining rotation systems is a simpler business because they admit a kind of coordinatization and hence an analog of the halving strategy from Lemma 3.2 is available for Spoiler. The most essential ingredient of the proof of Theorem 5.6.2 is a strategy for Spoiler in the Ehrenfeucht game on graphs allowing him to simulate the Ehrenfeucht game on the corresponding rotation systems.

5.1.4. *Graphs with an excluded minor.* No graph with treewidth h has K_{h+2} as a minor. The class of graphs embeddable into a closed 2-dimensional surface S is closed under minors and, as follows from the Robertson-Seymour Graph Minor Theorem, no graph from this class contains a minor of K_h for some $h = h(S)$. Extending his earlier work on graphs embeddable into a fixed surface [34], Grohe [36] recently announced a proof that, if a graph G does not contain K_h as a minor, then $W_\#(G)$ is bounded by a constant $c = c(h)$. The case of $h = 5$ is treated in detail in [35].

Alon, Seymour, and Thomas [2] proved that, if a graph G of order n does not contain a K_h as a minor, then it has a separator of size at most $h^{3/2}\sqrt{n}$. Using this result, for all connected graphs with this property one can prove [74] that $D(G) = O(h^{3/2}\sqrt{n}) + O(\Delta(G)\log n)$.

5.1.5. *Other classes of graphs.* A graph is *strongly regular* if all its vertices have equal degrees and, for some λ and μ, each pair of adjacent vertices has exactly λ common neighbors and each pair of non-adjacent vertices has exactly μ common neighbors. Non-isomorphic graphs with the same order, degree, and parameters λ and μ are standard examples of a failure of the 2-dim WL algorithm. Babai studies the isomorphism problem for this class in [5]. His individualization-and-refinement technique translates into a bound $W_{\#}(G) \le 2\sqrt{n}\log n$ for all strongly regular graphs of a sufficiently large order n with the exception for the disjoint unions of complete graphs and their complements (for which we have $D_{\#}(G) \le 3$). Further improvements are obtained by Spielman [73].

Grohe [37] considers chordal line graphs and proves that they have a bounded logical width in the counting logic. On the other hand, he shows that there are chordal graphs with $W_{\#}(G) = \Omega(n)$ and the same holds true for line graphs. The latter result is obtained by a reduction to the graphs with $W_{\#}(G) = \Omega(n)$ constructed by Cai, Fürer, and Immerman [14] (cf. Theorem 5.7 below). Note that the Cai-Fürer-Immerman graphs are regular of degree 3, where the regularity can be traded for the bipartiteness after a slight modification.

5.2. **General case.**

5.2.1. *Identification problem.* Recall that

$$W_{\#}(G,H) \le W(G,H) \le D(G,H) \text{ and } W_{\#}(G,H) \le D_{\#}(G,H) \le D(G,H).$$

If we are motivated by the graph isomorphism problem, it is quite natural to focus on these parameters under the assumption that G and H have the same order (even without saying that $D_{\#}(G,H) = 1$ otherwise). Distinguishing a graph G from all non-isomorphic H of *the same order* is sometimes called *identification problem.* In particular, we would like to determine or estimate the maximum of $D(G,H)$ (resp. $D_{\#}(G,H)$) as a function of $n = v(G) = v(H)$. Equivalently, what is the minimum $r = r(n)$ such that Spoiler has a winning strategy in $\textsc{Ehr}_r(G,H)$ for all non-isomorphic G and H of order n?

By taking disjoint unions of complete and empty graphs, it is easy to find G and H with $D(G,H) \ge (n+1)/2$. Bounding $D_{\#}(G,H)$ from below is much more subtle issue. Using a nice nontrivial argument, Cai, Fürer, and Immerman [14] came up with a linear lower bound.

Theorem 5.7 (Cai, Fürer, and Immerman [14]). *For infinitely many n there are non-isomorphic graphs G and H both of order n such that $W_{\#}(G,H) \ge c\,n$, where c is a positive constant.*

The calculation of Pikhurko, Veith, and Verbitsky [63, Section 7.5] shows that one can take $c = 0.00465$.

Let us turn to upper bounds. Suppose that $G \not\cong H$ and $v(G) = v(H) = n$. Before reading further, the reader might try to improve the trivial bound $D(G,H) \le n$ at least somewhat. It may be seen as a curious observation that $D(G,H) \le n - 1$

follows from the Harary version of the Ulam Reconstruction Conjecture, open for a long time, claiming that non-isomorphic graphs of equal orders have different sets of vertex-deleted subgraphs.

One solution of this exercise, giving $D(G,H) < n - \frac{1}{4}\log n$, is to apply the Erdős-Szekeres bound on Ramsey numbers. It implies that every graph G of large order n contains a homogeneous set of more than $\frac{1}{2}\log n + \frac{1}{4}\log\log n$ vertices. Spoiler pebbles the complement of such a set S in G. Suppose that the unpebbled set is independent (otherwise we can play on the complementary graphs). If Duplicator is lucky, she manages to pebble the complement to an independent set S' in H so that $G \setminus S \cong H \setminus S'$. Identifying the pebbled parts, Spoiler compares the number of vertices in S and in S' with the same neighborhood. These numbers cannot be identical for G and H and, by $v(G) = v(H)$, Spoiler can demonstrate this using at most $(|S|+1)/2$ further moves in one of the graphs.

After this warm-up, we can state an almost optimal bound.

Theorem 5.8 (Pikhurko, Veith, and Verbitsky [62])**.** *For every two non-isomorphic graphs G and H of the same order n we have $D(G,H) \le (n+3)/2$.*

5.2.2. *General bounds for the logical depth and width.* In the case of the counting logic, Theorem 5.7 provides us with infinitely many graphs G for which $D_{\#}(G) \ge W_{\#}(G) > 0.00465\,n$. As usually, n denotes the order of a graph. An upper bound easily follows from Theorem 5.8: we have $D_{\#}(G) \le 0.5\,n + 1.5$ for all G. Though this bound does not use the power of counting quantifiers at all, we are not aware of any better bound.

Consider the standard first-order logic (without counting). At the first sight, everything is clear here. Indeed, the general upper bound $W(G) \le D(G) \le n+1$ is attained, even for the width, by the complete graph K_n and by the empty graph $\overline{K_n}$. However, these are the only two extremal graphs. In other words, $D(G) \le n$ for all G with exception of $G \in \{K_n, \overline{K_n}\}$. As K_n and $\overline{K_n}$ are the most symmetric graphs, this observation suggests two problems. The first one is to prove a better bound for a class of graphs with restrictions on the automorphism group. The second is to obtain, for as small as possible $l = l(n)$, an explicit or algorithmic description of all order-n graphs whose logical depth (resp. width) exceeds l. We start with the first problem.

Definition 5.9. Let u, v, and s be three vertices and $s \notin \{u,v\}$. We say that s *separates* u and v if s is adjacent to exactly one of the two vertices. Furthermore, we call u and v *twins* if no s separates u and v (or, equivalently, if the transposition of u and v is an automorphism of the graph). A graph is called *irredundant* if it has no twins.

Theorem 5.10 (Pikhurko, Veith, and Verbitsky [62])**.** *If G is irredundant, then $D_1(G) \le (n+5)/2$.*

Theorem 5.10 cannot be improved to a sublinear bound. Indeed, consider mP_4, the disjoint union of m copies of P_4. As it is easily seen, mP_4 is irredundant and $D(mP_4) \ge D(mP_4, (m+1)P_4) > m$ (the reader is welcome to play $\textsc{Ehr}_m(mP_4, (m+1)P_4)$ on Duplicator's side). No sublinear improvement is possible even for connected graphs: the graphs constructed by Cai, Fürer, and Immerman in Theorem 5.7 are irredundant.

We prove Theorem 5.10 based on Lemma 2.5.1 and the Ehrenfeucht Theorem (Theorem 3.3.1). That is, we design a strategy allowing Spoiler to win $\mathrm{EHR}_r(G,H)$ for any $H \not\cong G$, where $r = \lfloor (n+5)/2 \rfloor$. As an important additional feature of the strategy, Spoiler will alternate between the graphs only once. By Theorem 3.3.2, this shows that our bound holds even for the logic with only one quantifier alternation (as it is indicated by the subscript in Theorem 5.10).

Definition 5.11. Let $X \subset V(G)$. Given two vertices $u, v \in V(G) \setminus X$, we call them X-*similar* and write $u \equiv_X v$ if u and v are inseparable by any vertex in X, i.e., if $N(u) \cap X = N(v) \cap X$.

Now, let $y \notin X$. We say that X *sifts out* y if for every $y' \notin X$ the relation $y \equiv_X y'$ implies $y' = y$ (in other words, the vertex y is uniquely identified by its adjacencies to X). Let $\mathcal{S}(X)$ consist of all $x \in X$ and all y sifted out by X. We call X a *sieve*[5] if $\mathcal{S}(X) = V(G)$. Furthermore, X is called a *weak sieve* if $\mathcal{S}(\mathcal{S}(X)) = V(G)$.

Consider the Ehrenfeucht game on non-isomorphic G and H and assume that X is a sieve in G. Let Spoiler pebble all vertices of X. We leave to the reader to verify that Spoiler can win in at most 2 more moves. We now describe a more advanced *Weak Sieve Strategy.*

Lemma 5.12. *If X is a weak sieve in G, then Spoiler is able, for any $H \not\cong G$, to win $\mathrm{EHR}_r(G,H)$ with $r \le |X| + 3$. Moreover, he does not need to jump from one graph to the other more than once during the game.*

Proof. First, Spoiler selects all of X. Let $X' \subset V(H)$ be the Duplicator's reply. Assume that Duplicator has not lost yet. For the notational simplicity let us identify X and X' so that $V(G) \cap V(H) = X = X'$ and the player's moves coincide on X. Let $Y = \mathcal{S}(X)$ in G and $Y' = \mathcal{S}(X')$ in H.

It is not hard to see that Spoiler wins in at most two extra moves unless the following holds. For any $y \in Y \setminus X$ there is a $y' \in Y' \setminus X$ (and vice versa) such that $N(y) \cap X = N(y') \cap X$. Moreover, this bijective correspondence between Y and Y' establishes an isomorphism between $G[Y]$ and $G'[Y']$.

Suppose that this is the case and identify Y with Y'. Let $Z = V \setminus Y$ and $Z' = V' \setminus Y$. Let $z \in Z$ and define

$$W'_z = \{\, z' \in Z' : N(z') \cap Y = N(z) \cap Y \,\}.$$

If $W'_z = \emptyset$, Spoiler wins in at most two moves. First, he selects z. Let Duplicator reply with z'. Assume that $z' \in Z'$ for otherwise she has already lost. As the neighborhoods of z, z' in Y differ, Spoiler can demonstrate this by picking a vertex of Y. If $|W'_z| \ge 2$, then Spoiler selects any two vertices in W'_z and wins with at most one more move, as required.

Hence, we can assume that for any z we have $W'_z = \{f(z)\}$ for some $f(z) \in Z'$. Since each vertex in Z is sifted out by Y, the function f is injective. If $f(Z) \ne Z'$, Spoiler easily wins in two moves. Suppose, therefore, that $f : Z \to Z'$ is a bijection.

[5]Babai [5] uses sieves under the name *distinguishing sets*.

As $G \not\cong H$, the mapping f does not preserve the adjacency relation between some $y, z \in Z$. Now, Spoiler selects both y and z. Duplicator cannot respond with $f(y)$ and $f(z)$; by the definition of f Spoiler can win in one extra move. $\square$

Theorem 5.10 immediately follows from Lemma 5.12 and the next lemma.

Lemma 5.13. *Any irredundant graph G on n vertices has a weak sieve X with $|X| \le (n-1)/2$.*

Proof. Given $X \subset V(G)$, let $\mathcal{C}(X)$ denote the partition of $\overline{X} = V(G) \setminus X$ into $\equiv_X$-equivalence classes. Starting from $X = \emptyset$, we repeat the following procedure. As long as there exists $u \in \overline{X}$ such that $|\mathcal{C}(X \cup \{u\})| > |\mathcal{C}(X)|$, we move u to X. As soon as there is no such u, we arrive at X which is $\mathcal{C}$-*maximal*, that is, $|\mathcal{C}(X \cup \{u\})| \le |\mathcal{C}(X)|$ for any $u \in \overline{X}$. Note that $|\mathcal{C}(X)| \ge |X| + 1$ because this inequality is true at the beginning and is preserved in each construction step. Using also the inequality $|X| + |\mathcal{C}(X)| \le n$, we conclude that $|X| \le (n-1)/2$.

We now prove that the X is a weak sieve. Suppose, to the contrary, that u and v are distinct $\mathcal{S}(X)$-similar vertices in $Z = V(G) \setminus \mathcal{S}(X)$. By the irredundancy of G, these vertices are separated by some s. We cannot have $s \in \mathcal{S}(X)$ by the definition of $\mathcal{S}(X)$-similarity. Thus $s \in Z$. Let C_1 be the class in $\mathcal{C}(X)$ including $\{u, v\}$ and C_2 be the class in $\mathcal{C}(X)$ containing s. Since $s \notin \mathcal{S}(X) \setminus X$, the class C_2 has at least one more element in addition to s. If $C_1 \ne C_2$, moving s to X splits up C_1 and does not eliminate C_2. If $C_1 = C_2$, moving s to X splits up this class and splits up or does not affect the others. In either case $|\mathcal{C}(X)|$ increases, giving a contradiction. $\square$

The proof of Theorem 5.10 is complete. This theorem was significantly extended in [62] giving some progress on the second research problem stated above. In particular, it was shown that one can efficiently check whether or not $D(G) \le (n+5)/2$ for the input graph G of order n and, if this is not true, then one can efficiently compute the exact value of $D(G)$. Also, the same holds for $W(G)$.

This result is interesting in view of the fact that algorithmic computability of the logical depth and width of a graph, even with no efficiency requirements, is unclear. A reason for this is that the question if a given first-order sentence defines some graph is known to be undecidable [60].

The upper bound of $\frac{1}{2}n + O(1)$ can be improved if we impose a restriction on the maximum vertex degree.

Theorem 5.14 (Pikhurko, Veith, and Verbitsky [62]). *Let $d \ge 2$. Let G be a graph of order n with no isolated vertex and no isolated edge. If $\Delta(G) \le d$, then*

$$D_1(G) < c_d\, n + d^2 + d + 4$$

for a constant $c_d = \frac{1}{2} - \frac{1}{4} d^{-2d-5}$.

Theorem 5.14 aims at showing a constant c_d strictly less than $1/2$ rather than at attempting to find the optimum c_d. In the case of $d = 2$, which is simple and included just for uniformity, an optimal bound is $D_1(G) \le n/3 + O(1)$. Without the assumption that G has no isolated vertex and edge, the theorem does not hold for any fixed $c_d < 1/2$. A counterexample is provided by the disjoint union of isolated edges. Even under the stronger assumption that G is connected, Theorem 5.14 still does not admit any sublinear improvement: the Cai-Fürer-Immerman graphs in Theorem 5.7 are connected and have maximum degree 3.

6. Average case bounds

In Section 5 we investigated the maximum values of logical parameters over graphs of order n. Now we want to know its typical values. A natural setting for this problem is given by the Erdős-Rényi model of a random graph $G_{n,p}$. The latter is a random graph on n vertices where every two vertices are connected by an edge with probability p independently of the other pairs. A particularly important case is $G_{n,1/2}$, when we have the uniform distribution on all graphs on a fixed set of n vertices. Whenever we say that for a random graph of order n something happens *with high probability* (abbreviated as *whp*), we mean a probability approaching 1 as $n \to \infty$.

6.1. Bounds for almost all graphs.

6.1.1. *Logic with counting.* We begin with a simple but useful observation about the color refinement algorithm described at the beginning of Section 4: If the coloring of a graph stabilizes with all color classes becoming singletons, it can be considered a canonical vertex ordering. It turns out that this happens for almost all graphs. This result can be used to estimate the logical complexity of almost all graphs, in particular, to show that almost surely $W_{\#}(G_{n,1/2}) = 2$ (Immerman and Lander [45]).

Theorem 6.1.

1. (Babai, Erdős, and Selkow [6]) *2 color refinements split a random graph $G_{n,1/2}$ into color classes which are singletons with probability more than $1 - 1/\sqrt[7]{n}$, for all large enough n. Consequently, $D^2_{\#}(G_{n,1/2}) \le 4$ with this probability.*
2. (Babai and Kučera [7]) *3 color refinements split a random graph $G_{n,1/2}$ into color classes which are singletons with probability more than $1 - 1/2^{cn}$, for a constant $c > 0$ and all large enough n. Consequently, $D^2_{\#}(G_{n,1/2}) \le 5$ with this probability.*

The logical conclusions made in Theorem 6.1 are based on the necessity part of Theorem 4.4. It suffices to notice that, once the color refinement splits the vertex set of an input graph G into singletons, one extra round of the algorithm suffices to distinguish G from any non-isomorphic graph.

Next, we are going to show that the upper bound of Theorem 6.1.1 is best possible. Let C^r_G denote the coloring of the vertex set of a graph G produced by the color refinement procedure in r rounds.

Lemma 6.2. *Whp for $G = G_{n,1/2}$ there exists a non-isomorphic graph H on $V = V(G)$ such that $C^2_G(x) = C^2_H(x)$ for every vertex $x \in V$.*

Proof. Let G be a typical graph of a sufficiently large order n. In particular, we assume that G satisfies Theorem 6.1.1 and that $|\deg x - n/2| \le m$ for every vertex x of G, where we can take e.g. $m = \sqrt{(n \log n)/2}$ by a simple application of

Chernoff's bound (see also [12, Corollary 3.4]). By the Pigeonhole Principle, there is a set U of $u = \lceil n/(2m+1) \rceil$ vertices all having the same degree.

Another property of the random graph G that we assume is that every set $X \subset V$ of size u contains distinct vertices w, x, y, z with $wx, yz \in E(G)$ and $xy, zw \notin E(G)$. Indeed, let us fix a u-set $X \subset V$ and estimate the probability that it violates this property. One can find at least $\binom{u}{2}/5$ edge-disjoint 4-cycles inside the complete graph on X. (For example, picking up cycles one by one arbitrarily, we get enough of them by the classical result of Kővari, Sös, and Turán [49] saying that a C_4-free graph on u vertices has $O(u^{3/2})$ edges.) For each 4-cycle on vertices x_1, x_2, x_3, x_4 in this order, at least one of the relations $x_1x_2, x_3x_4 \in E(G)$ and $x_2x_3, x_4x_1 \notin E(G)$ should be false, this having probability 15/16. By the edge-disjointness, these events for different selected cycles are mutually independent. Hence X violates the desired property with probability is at most $(15/16)^{\binom{u}{2}/5} = o(\binom{n}{u}^{-1})$. Since there are $\binom{n}{u}$ candidates for a bad set X, the probability that it exists is $o(1)$, giving the required.

Hence, the equidegree set U contains vertices w, x, y, z with $wx, yz \in E(G)$ and $xy, zw \notin E(G)$. Let H be obtained from G by removing edges wx, yz and adding edges xy, zw. This operation preserves the degree of every vertex as well as the multiset of degrees of its neighbors, that is, $C^2_G(v) = C^2_H(v)$ for every vertex $v \in V$.

Suppose that G and H are isomorphic. Any isomorphism f must preserve the C^2-colors. Since C^2-classes are all singletons, f has to be the identity map on $V(G) = V(H)$. But then the adjacency between, e.g., w and x is not preserved, a contradiction. The lemma is proved. □

Given a typical $G = G_{n,1/2}$, let H be a graph satisfying Lemma 6.2. Thus, the 2-round color refinement fails to distinguish between G and H. By the sufficiency part of Theorem 4.4, we have $D^2_\#(G) > 3$. As an alternative proof, the reader can design a winning strategy for Duplicator in the counting game $\text{EHR}^2_3(G, H)$. Together with Theorem 6.1.1, this bound gives us the exact value $D^2_\#(G_{n,1/2})$.

Theorem 6.3. *Whp* $D^2_\#(G_{n,1/2}) = 4$.

We always have $D_\#(G) \le D^2_\#(G)$ and, on the other hand, $D_\#(G) \le 2$ implies $D^2_\#(G) \le 2$ because any definition with quantifier depth 2 can be rewritten with using only 2 variables. It follows from Theorem 6.3 that $3 \le D_\#(G_{n,1/2}) \le 4$ whp. Unfortunately, we could not decide whether the typical value of $D_\#(G_{n,1/2})$ is 3 or 4, which seems to be an interesting question.

6.1.2. *Logic without counting.*

Theorem 6.4 (Kim et al. [48])**.** *Fix an arbitrarily slowly increasing function* $\omega = \omega(n)$. *Then we have whp that*

$$\log n - 2\log\log n + \log\log \mathrm{e} + 1 - o(1) \le W(G_{n,1/2}) \le$$
$$\le D_1(G_{n,1/2}) \le \log n - \log\log n + \omega.$$

We first prove the lower bound.

Definition 6.5. For an integer $k \geq 1$, we say that a graph G has the *k-extension property* if, for every two disjoint $X, Y \subset V(G)$ with $|X \cup Y| \leq k$, there is a vertex $z \notin X \cup Y$ adjacent to all $x \in X$ and non-adjacent to all $y \in Y$.

Lemma 6.6. *If both G and H have k-extension property, then $W(G, H) \geq k + 2$.*

Proof. By Theorem 3.3.3 it suffices to design a strategy allowing Duplicator to survive in $\mathrm{E{\scriptsize HR}}^{k+1}(G, H)$ arbitrarily long. Suppose that Spoiler puts pebble p on a new position v in one of the graphs, say, G. Let X (resp. Y) denote the set of pebbled vertices in H whose counter-parts in G are adjacent (resp. non-adjacent) to v. Duplicator moves the other copy of p to a vertex z with the given adjacencies to $X \cup Y$ whose existence is guaranteed by the k-extension property. □

Lemma 6.7. *Let $\epsilon > 0$ be a real constant. Then the k-extension property holds for $G_{n,1/2}$ whp for any $k \leq \log n - 2 \log \log n + \log \log \mathrm{e} - \epsilon$.*

Proof. Let n be large. Any particular X and Y with $|X \cup Y| = k$ falsify the k-extension property with probability $(1 - 2^{-k})^{n-k}$. Since the number of such pairs is $\binom{n}{k}2^k$, a random graph $G_{n,1/2}$ does not have the k-extension property with probability at most

$$\binom{n}{k}2^k(1-2^{-k})^{n-k} \leq n^k(1-2^{-k})^n \leq \exp\left\{k \ln n - n2^{-k}\right\}.$$

The former inequality is true only if $k \geq 4$ but this makes no problem because the k-extension property implies itself for all smaller values of parameter k. Since the function $f(x) = x \ln n - n2^{-x}$ is monotone, the k-extension property fails with the probability bounded from above by

$$\exp\left\{f(\log n - 2\log\log n + \log\log \mathrm{e} - \epsilon)\right\} =$$
$$= \exp\left\{(\ln 2)\left(-2^{\epsilon} + 1 + o(1)\right)\log^2 n\right\} = o(1),$$

as it was claimed. □

Fix $\epsilon > 0$. Let n be sufficiently large and set $k = \lfloor \log n - 2 \log\log n + \log\log \mathrm{e} - \epsilon \rfloor$. By Lemma 6.7, $G = G_{n,1/2}$ has the k-extension property whp. Let H be a graph which also possesses the k-extension property and is non-isomorphic to G. The existence of such a graph follows also from Lemma 6.7: Given G, let $H = G_{n,1/2}$ be another, independent copy of a random graph. It should be only noticed that $H \cong G$ with probability at most $n!2^{-\binom{n}{2}} = o(1)$. By Lemma 6.6, we have

$$W(G) \geq W(G, H) \geq k + 2 > \log n - 2 \log\log n + \log\log \mathrm{e} + 1 - \epsilon,$$

thereby proving the lower bound of Theorem 6.4.

To prove the upper bound, we employ the *Weak Sieve Strategy* that was designed in Section 5.2.2. Lemma 5.12 allows us to estimate the parameter $D_1(G)$ by the size of a weak sieve existing in G. The upper bound of Theorem 6.4 follows from Lemmas 6.8 below, that gives us a good enough bound for the size of a weak sieve in a random graph. The paper [48] states a slightly weaker upper bound than that in Theorem 6.4 (namely, $\omega = C \log\log\log n$ there). The current more precise estimate is due to Joel Spencer (unpublished).

Lemma 6.8. *Fix an arbitrarily slowly increasing function $\omega = \omega(n)$. Then whp $G_{n,1/2}$ has a weak sieve of size at most $\log n - \log\ln n + \omega$.*

Proof. We will consider a random graph $G_{n,1/2}$ on an n-vertex set V. Fix $X \subset V$ with $|X| = \log n - s$, where $s = \log\ln n - \omega$. We generate $G_{n,1/2}$ in two stages.

Stage 1: reveal the edges between X and $V \setminus X$ (needless to say, each such edge appears with probability 1/2 independently of the others). Our goal at this stage is to show that $\mathcal{S}(X)$ is large whp.

A fixed $y \in V \setminus X$ is sifted out by X with probability

$$(1 - 2^{-|X|})^{n-|X|-1} = \exp\{-2^s(1 + o(1))\} = n^{-2^{-\omega}(1+o(1))} = n^{-o(1)}.$$

By linearity of expectation

$$\mathbb{E}[\,|\mathcal{S}(X) \setminus X|\,] = (n - |X|)n^{-o(1)} = n^{1-o(1)}.$$

We can now apply the martingale techniques to show that whp $|\mathcal{S}(X) \setminus X|$ is concentrated near its mean value. More precisely, we need the following estimate:

$$\mathbb{P}\Big[\,|\mathcal{S}(X) \setminus X| < \mathbb{E}[\,|\mathcal{S}(X) \setminus X|\,] - 2\lambda\sqrt{n - |X|}\,\Big] < \mathrm{e}^{-\lambda^2/2} \tag{14}$$

for any $\lambda > 0$, where $\mathbb{P}[\,A\,]$ denotes the probability of an event A.

To prove it, consider the probability space consisting of all functions $g : V \setminus X \to 2^X$. Define a random variable L on this space by setting $L(g)$ to be equal to the number of values in 2^X taken on by g exactly once. Note that, if g and g' differ only at one point, then $|L(g) - L(g')| \le 2$. Construct an appropriate martingale as explained in the Alon-Spencer book [3, Chapter 7.4]. Namely, let $V \setminus X = \{y_1, \ldots, y_m\}$ and define a sequence of auxiliary random variables $X_0, X_1, \ldots, X_m$ by $X_i = \mathbb{E}\left[\frac{1}{2}L(g) \mid g(y_j) = h(y_j) \text{ for all } j \le i\right]$. By Azuma's inequality (see [3, Theorems 7.2.1 and 7.4.2]), for all $\lambda > 0$ we have

$$\mathbb{P}\left[\,L(g) < \mathbb{E}[\,L(g)\,] - 2\lambda\sqrt{m}\,\right] < \mathrm{e}^{-\lambda^2/2},$$

which is exactly what is claimed by (14).

By (14) we have whp that

$$|\mathcal{S}(X) \setminus X| \ge n^{1-o(1)}.$$

Conditioning on $\mathcal{S}(X)$ satisfying this bound, we go to the next stage of generating $G_{n,1/2}$.

Stage 2: reveal the edges inside $V \setminus X$. It is enough to show that $V \setminus \mathcal{S}(X) \subset \mathcal{S}(\mathcal{S}(X) \setminus X)$ whp. If the last claim is false, then there are $z, z' \in V \setminus \mathcal{S}(X)$ having the same adjacencies to $\mathcal{S}(X) \setminus X$. This happens with probability no more than

$$\binom{n}{2} 2^{-|\mathcal{S}(X)\setminus X|} < n^2 2^{-n^{1-o(1)}} = o(1).$$

The proof is complete. $\square$

Theorem 6.4 shows rather close lower and upper bounds for the logical width and depth of a random graph $G_{n,1/2}$. Surprisingly, even this can be improved.

Theorem 6.9 (Kim et al. [48]). *For infinitely many n we have whp*

$$D_2(G_{n,1/2}) \le \log n - 2\log\log n + \log\log \mathrm{e} + 6 + o(1).$$

This upper bound is at most by $5+o(1)$ larger than the lower bound of Theorem 6.4. It follows that, for infinitely many n, the parameters $D_i(G)$ with $i \ge 2$, $D(G)$, and $W(G)$ are all concentrated on at most 6 possible values (while some extra work, see [48, Section 4.3], gives a 5-point concentration).

6.1.3. *Bounds for trees.*

Theorem 6.10 (Bohman et al. [11]). *Let T_n denote a tree on the vertex set $\{1,2,\ldots,n\}$ selected uniformly at random among all n^{n-2} such trees. Whp we have $W(T_n) = (1+o(1))\frac{\log n}{\log\log n}$ and $D(T_n) = (1+o(1))\frac{\log n}{\log\log n}$.*

The lower bound for $W(T_n)$ immediately follows from the following property of a random tree: whp T_n has a vertex adjacent to $(1+o(1))\frac{\log n}{\log\log n}$ leaves. Note that the upper bound for $D(T_n)$ does not follow directly from Theorem 5.4 because whp $\Delta(T_n) = (1+o(1))\frac{\log n}{\log\log n}$ (see Moon [56]).

6.2. **An application: The convergency rate in the zero-one law.** We will write $G \models \Phi$ to say that a sentence Φ is true on a graph G. Let $p_n(\Phi) = \mathbb{P}[\,G_{n,1/2} \models \Phi\,]$. The *0-1 law* established by Glebskii et al. [30] and, independently, by Fagin [27] says that, for each Φ, $p_n(\Phi)$ approaches 0 or 1 as $n \to \infty$. Denote the limit by $p(\Phi)$.

Define the *convergency rate function* for the 0-1 law by

$$R(k,n) = \max_{\Phi}\left\{\, |p_n(\Phi) - p(\Phi)| : D(\Phi) \le k \right\}.$$

Note that the maximization here can be restricted to a finite set by Theorem 2.4.1. Therefore, the standard version of the 0-1 law implies that $R(k,n) \to 0$ as $n \to \infty$ for any fixed k. Naor, Nussboim, and Tromer [57] showed that $R(\log n - 2\log\log n, n) \to 0$. Another result in [57] states that one can choose $p(n) = 1/2 + o(1)$ and $k(n) = (2+o(1))\log n$ such that the probability that $G_{n,p}$ has a $k(n)$-clique is bounded away from 0 and 1. Thus for this probability $p(n)$ the 0-1 law does not hold with respect to formulas of depth $k(n)$.

The following theorem sharpens slightly the first part of the above result and improves on the second part in two aspects: we do not need to change the probability $p = 1/2$ and we get an almost best possible upper bound.

Theorem 6.11. *Let $g(n) = \log n - 2\log\log n + \log\log \mathrm{e} + c$ with constant c.*

1. *If $c < 1$, then $R(g(n), n) \to 0$ as $n \to \infty$.*
2. *If $c > 6$, then $R(g(n), n)$ does not tend to 0 as $n \to \infty$. More strongly, for every $\gamma \in [0,1]$ there is a sequence of formulas $\Phi_{n_1}, \Phi_{n_2}, \ldots$ (where $n_i < n_{i+1}$) with $D(\Phi_{n_i}) \le g(n_i)$ such that $p_{n_i}(\Phi_{n_i}) \to \gamma$ as $i \to \infty$.*

Part 2 follows from Theorem 6.9. The latter implies that (for infinitely many n) actually *any* property $\mathcal{P}$ of graphs on n vertices can be “approximated” by a first-order sentence of depth at most $g(n)$. Indeed, take the conjunction of defining formulas over all graphs in $\mathcal{P}$ of order n and depth at most $g(n)$. The omitted graphs constitute negligible proportion of all graphs by Theorem 6.9.

We now prove Part 1. Like the proof in [57] we use the extension property, but we argue in a slightly different way.

Proof of Part 1. Let E_k denote a first-order statement of quantifier depth k expressing the $(k-1)$-extension property. Lemma 6.7 provides us with an infinitesimal $\alpha(n)$ such that, as long as

$$k \le g(n), \tag{15}$$

we have

$$1 - p_n(E_k) \le \alpha(n). \tag{16}$$

Let Φ be a first-order statement with $D(\Phi) = k$. Under the condition (15), we have to estimate $|p_n(\Phi) - p(\Phi)|$ from above by a function depending only on n (for $\Phi = E_k$ this is done by (16)). It suffices to do this for k larger than a constant k_0. We have to fix k_0 large enough. Note that $g(x)$ is monotone for $x \ge \mathrm{e}^2$. First of all, we require that $k_0 \ge g(\mathrm{e}^2)$; then we can speak of the value of $g^{-1}(k_0)$. Moreover, we choose k_0 so that $1 - \alpha(n) \ge \sqrt{3}/2$ whenever $n \ge g^{-1}(k_0)$. In the following we suppose that $k \ge k_0$. By our choice of k_0, we have

$$p_n(E_k) \ge \frac{\sqrt{3}}{2} \tag{17}$$

whenever k and n satisfy (15).

Let G' and G'' be two independent copies of $G_{n,1/2}$. By Lemma 6.6,

$$\mathbb{P}\,[\,D(G', G'') > k\,] \ge \mathbb{P}\,[\,G' \models E_k \;\&\; G'' \models E_k\,] = p_n(E_k)^2.$$

On the other hand,

$$\mathbb{P}\,[\,D(G', G'') > k\,] \le \mathbb{P}\,[\,G' \text{ and } G'' \text{ are not distinguished by } \Phi\,] = p_n(\Phi)^2 + (1 - p_n(\Phi))^2.$$

It follows that

$$2\,p_n(\Phi)(1 - p_n(\Phi)) \le 1 - p_n(E_k)^2.$$

Define N_Φ to be the maximum number n for which $|p_n(\Phi) - p(\Phi)| > 1/2$ (and let $N_\Phi = 0$ if no such n exists). For all

$$n > N_\Phi, \tag{18}$$

we immediately obtain

$$|p_n(\Phi) - p(\Phi)| \le 1 - p_n(E_k)^2 \le 2(1 - p_n(E_k)).$$

By (16), this gives us a desired bound

$$|p_n(\Phi) - p(\Phi)| \le 2\,\alpha(n) = o(1)$$

provided condition (15) holds. We are done modulo an extra assumption in (18). To complete the proof, it suffices to show that condition (18) actually follows from condition (15) even when $N_\Phi > 0$.

Without loss of generality, suppose that $p(\Phi) = 0$. Denote $N = N_\Phi$ and let $M = N + 1$. By the definition of N_Φ, we have $p_M(\Phi) \le 1/2$. Let G_N and G_M be independent random graphs with, respectively, N and M vertices. Note that

$$\mathbb{P}\,[\,D(G_N, G_M) \le k\,] \ge \mathbb{P}\,[\,G_N \models \Phi \;\&\; G_M \not\models \Phi\,] > \frac{1}{4}.$$

On the other hand,

$$\mathbb{P}\,[\,D(G_N, G_M) > k\,] \ge \mathbb{P}\,[\,G_N \models E_k \;\&\; G_M \models E_k\,] = p_N(E_k)p_M(E_k).$$

It follows that

$$p_N(E_k)p_M(E_k) < \frac{3}{4}.$$

Since this contradicts (17) for $n = N$ or $n = M$, we conclude that $k > g(N)$. Comparing this with our assumption (15), we obtain (18) by the monotonicity of $g(x)$ for $x \ge \mathrm{e}^2$. $\square$

6.3. **The evolution of a random graph.** We now take a dynamical view on a random graph $G_{n,p}$ by letting the edge probability p vary. With p varying from 0 to 1, $G_{n,p}$ evolves from empty to complete. We want to trace the changes of its logical complexity during the evolution. Since the definability parameters do not change when we pass to the complement of a graph, we can restrict ourselves to case $p \le 1/2$.

When p is a constant, one can estimate $D(G)$ within additive error $O(\log\log n)$.

Theorem 6.12 (Kim et al. [48])**.** *If $0 < p \le 1/2$ is constant, then whp*

$$\log_{1/p} n - c_1 \ln\ln n - O(1) \le W(G_{n,p}) \le D(G_{n,p}) \le \log_{1/p} n + c_2 \ln\ln n,$$

where $c_1 = 2\ln^{-1}(1/p)$ and $c_2 = (2 + o(1))\,(-p\ln p - (1-p)\ln(1-p))^{-1} - c_1$.

Sketch of Proof. Similarly to Theorem 6.4, the lower bound is based on the k-extension property. However, the proof of the upper bound is quite different. In particular, we have hardly any control on the alternation number in this result. The argument is rather complicated so we give only a brief sketch, concentrating more on its logical rather than probabilistic component.

Let $G = G_{n,p}$ be typical and $G' \not\cong G$ be arbitrary. Let $V = V(G)$ and $V' = V(G')$. For a sequence X of vertices, let $V_X = \{y \in V : \forall\, x \in X\ xy \in E(G)\}$ and $G_X = G[V_X]$. Let the analogous notation (with primes) apply to G'. If there is $x \in V$ such that for every $x' \in V'$ we have $G_x \not\cong G'_{x'}$, then Spoiler selects x. Whatever Duplicator's reply $x' \in V'$ is, Spoiler reduces the game to non-isomorphic graph G_x and $G'_{x'}$. We expect that $|V_x| = (p + o(1))n$ and G_x is also 'typical'. Thus Spoiler used one move to reduce the order of the random graph by a factor of p, which should lead to the upper bound $D(G) \le (1 + o(1))\log_{1/p} n$.

Suppose now that there are $x \in V$ and distinct $y', z' \in V'$ such that $G_x \cong G'_{y'} \cong G'_{z'}$. Spoiler selects $y' \in V'$. Assume that Duplicator replies with $y = x$, for otherwise $G_y \not\cong G'_{y'}$ and Spoiler proceeds as above. Now Spoiler selects z'; let $z \in V$ be the Duplicator's reply. We can assume that $G_{y,z} \cong G'_{y',z'}$, for otherwise Spoiler applies the inductive strategy to the $(G_{y,z}, G_{y',z'})$-game, where the order of the random graph is reduced by factor $(1+o(1))\,p^2$. Let $U=V_{y,z}$ and $U'=V'_{y',z'}$. A first moment calculation shows that there is vertex $v\in V_y\setminus U$ such that no vertex of $V_z\setminus U$ has the same neighborhood in U as v. Let Spoiler select v and let $v'\in V'_{y'}\setminus U'$ be the Duplicator's reply. Two copies $G'_{y'}$ and $G'_{z'}$ of a 'typical' graph G_x have a large vertex intersection. Another first moment calculation shows that whp there is only one way to achieve this, namely that the (unique) isomorphism $f: V'_{y'}\to V'_{z'}$ between $G'_{y'}$ and $G'_{z'}$ is in fact the identity on U'. But then $f(v')$ has the same adjacencies to U' as v'. Spoiler selects $f(v')$ and wins the game in at most one extra move.

Finally, up to a symmetry it remains to consider the case that there is a bijection $g: V\to V'$ such that for any $x\in V$ we have $G_x\cong G'_{g(x)}$.

As $G\not\cong G'$, there are $y,z\in V$ such that g does not preserve the adjacency between y and z. Spoiler selects y. We can assume that Duplicator replies with $y'=g(y)$ for otherwise Spoiler reduces the game to G_y. Now, Spoiler selects z to which Duplicator is forced to reply with $z'\neq g(z)$. Let $w=g^{-1}(z')$. Assume that $G_{y,z}\cong G_{y',z'}$ for otherwise Spoiler applies the inductive strategy to these graphs. But then $G_{y,z}$ is an induced subgraph of $G_w\cong G'_{z'}$, a property that we do not expect to see in a random graph.

In order to convert this rough idea into a rigorous proof one has to show that whp as long as the subgraphs $G_{x_1,x_2,\ldots}$ that can appear in the game are sufficiently large, they have all required properties. Also, one has to design Spoiler's strategy to deals small subgraphs of $G_{n,p}$ at the end of the game. All details can be found in [48, Section 3]. □

It is interesting to investigate the behavior, e.g., of $D(G_{n,p})$ when $p=p(n)$ tends to zero. In particular, it is open whether, for every constant $\delta\in(0,1)$ and $n^{-\delta}\le p(n)\le 1/2$ we have whp $D(G_{n,p})=O(\log n)$.

Some restriction on $p(n)$ from below is necessary here. Indeed, let G be an arbitrary non-empty graph (i.e., G has at least one edge) and let G' be obtained from G by adding one more isolated vertex. It is easy to see that $W(G,G')>d_0(G)$ and $D(G,G')>d_0(G)+1$, where $d_0(G)$ denotes the number of isolated vertices of G. It follows that

$$W(G)\ge d_0(G)+1 \quad\text{and}\quad D(G)\ge d_0(G)+2. \tag{19}$$

It is well known (see, e.g., [12]) that

$$d_0(G_{n,p})=(\mathrm{e}^{-pn}+o(1))\,n \tag{20}$$

whp as long as $p=O(n^{-1})$. In particular, we have $W(G_{n,p})=(1-o(1))n$ whenever $p=o(n^{-1})$.

In some cases, the lower bounds (19) are sharp.

Lemma 6.13. *Let $c_F(G)$ denote the number of connected components in a graph G isomorphic to a graph F. Suppose that a non-empty graph G satisfies*

$$c_F(G)+v(F)\le d_0(G)+1,\quad \textit{for every component } F \textit{ of } G. \tag{21}$$

Then $W(G)+1=D(G)=D_1(G)=d_0(G)+2$.

Proof. Let us show that $D_1(G,H) \le d_0(G)+2$ for any $H \not\cong G$. Let F be such that $c_F(H) \ne c_F(G)$. For definiteness suppose that $c_F(H) > c_F(G)$. Spoiler marks $c_F(G)+1$ components of H which are isomorphic to F by pebbling one vertex in each of them. Duplicator is forced either to mark one of the F-components of G twice (by pebbling two vertices, say, u and v in it) or to mark a component F' of G which is not isomorphic to F. In the former case Spoiler wins by pebbling a path from u to v. In the latter case Spoiler pebbles completely the F-component of H corresponding to F'. Duplicator is forced to pebble a connected part F'' of F'. If she has not lost yet, then $F'' \cong F$ and hence F'' is a proper subgraph of F'. Spoiler wins by pebbling another vertex in F' which is adjacent to a vertex in F''. Altogether at most $d_0(G)+2$ moves are made.

It remains to prove the upper bound on the width. The last move may require using the $(d_0(G)+2)$-th pebble. However, for this purpose Spoiler can reuse a pebble placed earlier in a component different from F'. This trick is unavailable only if $c_F(G)=1$ and $c_F(H)=0$ or if $c_F(G)=0$ and $c_F(H)=1$. In both cases Spoiler can win in at most $v(F)+1$ rounds (and at most one alternation). Moreover, if this number is at least $d_0(G)+2$, then $c_F(G)=0$ and G has no component with $v(F)$ or more vertices by (21). In this case, Spoiler can win in at most $v(F)$ moves. □

Theorem 6.14 (Kim et al. [48], Bohman et al. [11]). *If $p=c/n$ with $c=c(n)\ge 0$ being an arbitrary bounded function of n, then $D(G_{n,p}) = (\mathrm{e}^{-c}+o(1))n$ whp.*

Sketch of Proof. It is well known that, observing the evolution process in the scale $p=c/n$, at the point $c=1$ we encounter the phase transition. If $c<1-\epsilon$, whp all components of $G_{n,p}$ have $O(\log n)$ vertices each; if $c>1+\epsilon$, there appears a unique exception, the so-called *giant* component with a linear number of vertices.

One can check that, for any $c<\alpha-\epsilon$, Condition (21) holds whp (even for the giant component if it exists), where $\alpha = 1.1918...$ is a root of some explicit equation, see [48, Theorem 19]. Then, by Lemma 6.13 and Equality (20), $W(G_{n,p})$ and $D_1(G_{n,p})$ (and all parameters in between) are $(\mathrm{e}^{-c}+o(1))\,n$. When c is larger than $\alpha+\epsilon$, then whp the giant component of $G_{n,p}$ violates (21): Its order exceeds the number of isolated vertices. This case is handled in [11] as follows.

Denote the giant component of $G=G_{n,p}$ by M. Given $H \not\cong G$, we have to design a strategy allowing Spoiler to fast enough win the Ehrenfeucht game on G and H. The strategy in the proof of Lemma 6.13 does not work only if $c_M(H)=0$ or $c_M(H)\ge 2$. We adapt it for these cases so that Spoiler, instead of selecting all vertices of M, plays an optimal strategy for M using at most $D(M)+\log n+1$ moves (instead of $v(M)+1$ moves as earlier).

First, we can assume that no component of H has diameter n or more. Otherwise Spoiler pebbles u and v at distance n in H. For Duplicator's responses u' and v' in G we have either $\mathit{dist}(u',v')<n$ or $\mathit{dist}(u',v')=\infty$. Hence Spoiler wins in less than $\log n+1$ moves.

Second, we can assume that Duplicator always respects the connectivity relation (two vertices are in the relation if they are connectable by a path). Indeed, suppose that u and v belong to the same connected component F in one of the graphs while their counter-parts u' and v' are in different components of the other graph. Then Spoiler wins in less than $\log diam(F) + 1$ moves.

Under this assumption, Spoiler easily forces that, starting from the 2nd round, the play goes on components of G and H, of which exactly one is isomorphic to M. One of the main results of [11] states that whp

$$D(M) = O\left(\frac{\ln n}{\ln \ln n}\right), \tag{22}$$

which implies that Spoiler is able to win quickly and proves the theorem.

The upper bound (22) is obtained roughly as follows. By iteratively removing vertices of degree 1 from the giant component M, one obtains the *core* C of M (that is, C is a maximum subgraph with minimum degree at least 2). The *kernel* K of G is the *serial reduction* of C, that is, we iterate the following to obtain K: If there is a vertex x of degree 2, then we remove x but add edge $\{y, z\}$, where y and z are the two neighbors of x. The kernel may have loops and multiple edges and has to be modeled as a *colored graph*. The original graph G can be encoded by specifying its kernel K and the structure of rooted trees that correspond to each vertex or edge of K, the latter being viewed as a total coloring of K. It happens that whp every vertex x of K can be *identified* by a small-depth formula Φ_x with one free variable (that is $K, x \models \Phi_x$ while $K, y \not\models \Phi_x$ for every other vertex $y \in V(K)$) in the first-order language of colored graphs. Thus one can define K succinctly by stating that for every $x \in V(K)$ there is a unique vertex satisfying Φ_x, that every vertex satisfies Φ_x for some $x \in V(K)$, and by listing the adjacencies between vertices identified by Φ_x and Φ_y for every $x, y \in V(K)$. The core C can now be defined by specifying the length of the path corresponding to each edge of K, while the giant component M can be defined by specifying the random rooted trees hanging on the vertices of C using Theorem 6.10 (which relies in part on Theorem 5.4). □

The bound (22) is optimal up to a constant factor. This follows from the fact that whp the giant component M has a vertex v adjacent to at least $(1-\epsilon)\log n/\log\log n$ leaves. (Indeed, consider the graph $M' \not\cong M$ that is obtained from M by attaching an extra leaf at v.) We believe that the lower bound is sharp, that is, whp $D(M) = (1 + o(1))\log n/\log\log n$, but we were not able to settle this question.

Finally, we consider edge probabilities $p = n^{-\alpha}$ with rational $\alpha \in (0, 1)$. Such p occur as threshold functions for (non-)appearance of particular graphs as induced subgraphs in $G_{n,p}$. What is relevant to our subject is that such p show an irregular behavior of $G_{n,p}$ with respect to first-order properties.

Since the treatment of the general case of rational α would require a considerable amount of technical work, the paper [48] focuses on a sample value $\alpha = 1/4$, when $D(G_{n,p})$ falls down and becomes so small as it is essentially possible (cf. Section 7).

Theorem 6.15 (Kim et al. [48]). *If $p = n^{-1/4}$, then whp*

$$\log^* n - \log^* \log^* n - 1 \le D(G_{n,p}) \le D_3(G_{n,p}) \le \log^* n + O(1).$$

Sketch of Proof. The upper bound is based on the following ideas. Let the predicate $C(x_1, x_2, x_3, x_4)$ state that these 4 distinct vertices have no common neighbor. Its probability is $(1-p)^{n-4} = \mathrm{e}^{-1} + o(1)$ and its values over different 4-tuples are rather weakly correlated. Thus, if for a set A and a vertex $v \notin A$, we define $H_v(A)$ be the 3-uniform hypergraph on A with $x_1, x_2, x_3 \in A$ being a hyperedge if and only if $C(v, x_1, x_2, x_3)$ holds, then $H_v(A)$ behaves somewhat like a random hypergraph. As it is shown in [48, Lemma 21], one can find 4 vertices such that their common neighborhood A is relatively large (namely, $|A| = \lfloor \ln^{0.3} n \rfloor$) and yet there are vertices a, m such that hypergraphs $H_a(A)$ and $H_m(A)$ encode in some way the multiplication and addition tables for an initial interval of integers. Also, any integer can be succinctly defined in first-order logic with arithmetic operations. Roughly speaking, in order to define an integer j, one can write it in binary $j = b_k \ldots b_1$ and specify for every $i \le k$ the i-th bit b_i; crucially, the same binary expansion trick can be used recursively to specify the index i, and so on. This allows us to identify vertices A with very small depth. Next, we consider the set B of vertices of G that have exactly 4 neighbors in A and are uniquely determined by this. Again, the vertices of B are easy to identify (just list the 4 neighbors in A). Finally, if A was chosen carefully, then each vertex w of G is uniquely identified by the hypergraph $H_w(B)$. (The reason that we need an intermediate set B is that the number of possible 3-uniform hypergraphs $H_w(A)$ is at most $2^{\binom{|A|}{3}} < n - |A|$, that is, too small.) Of course, many technical difficulties arise when one tries to realize this approach.

The lower bound in Theorem 6.15 is very general. We use only the simple fact that any particular unlabeled graph with m edges is the value of $G_{n,p}$, where $p = n^{-1/4}$, with probability at most

$$n! p^m (1-p)^{\binom{n}{2}-m} \le n!(1-p)^{\binom{n}{2}} \le \exp\left(-(1/2 - o(1))n^{7/4}\right).$$

Let $F(k)$ be the number of non-isomorphic graphs definable with depth at most k. Then $\mathbb{P}[\, D(G) \le k \,] \le F(k) \exp\left\{-(1/2 - o(1))n^{7/4}\right\}$. By Theorem 2.3, $F(k) \le \mathit{Tower}(k + 2 + \log^* k)$. If $k = \log^* n - \log^* \log^* n - 2$, we have $F(k) \le 2^n$ and hence $\mathbb{P}[\, D(G) \le k \,] = o(1)$. $\square$

The above idea (arithmetization of certain vertex sets in graphs) has been previously used by Spencer [71, Section 8] to obtain non-convergence and non-separability results on the example of $G_{n,p}$ with $p = n^{-1/3}$.

So far we have considered the evolution of the logical complexity of a random graph in the standard logic with no counting. We conclude this section with an extension of Theorem 6.1.

Theorem 6.16 (Czajka and Pandurangan [16])**.** *Let $p(n)$ be any function of n such that $\frac{\omega(n) \log^4 n}{n \log\log n} \le p(n) \le 1/2$ where $\omega(n) \to \infty$ as $n \to \infty$. Then 2 color refinements split a random graph $G_{n,p}$ into color classes which are singletons with probability that is higher than $1 - n^{-c}$ for each constant $c > 0$ and all large enough n. Consequently, $D^2_{\#}(G_{n,p}) \le 4$ with this probability.*

Note that, in the case of $p = 1/2$, this result improves the probability bound in Part 1 of Theorem 6.1, while the probability bound in Part 2 is still better.

By elaborating on the argument of Lemma 6.2, we are able to supply Theorem 6.16 with the matching lower bound (that is, $D^2_{\#}(G_{n,p}) \ge 4$ whp) within the range $4\sqrt{\log n/n} \le p \le 1/2$.

7. Best-case bounds: Succinct definitions

As in the preceding sections, we consider the logical depth of graphs with a given number of vertices n. We know that the maximum value $D(G) = n + 1$ is attained by the complete and empty graphs (and only by them) and that the typical values lie around $\log n$ (see Theorem 6.4). Now we are going to look at the minimum. We already have a good starting point: By Theorem 6.15, there are graphs with

$$D_3(G) \le \log^* n + O(1).$$

In order to get such examples, we have to generate a random graph with the edge probability $n^{-1/4}$. In Section 7.1 we give three explicit constructions achieving the same bound. In Section 7.2 we introduce the *succinctness function* $q(n) = \min\{\, D(G) : v(G) = n\}$ and give an account of what is known about it. Section 7.3 is devoted to the question of how succinctly we can define graphs if we are not allowed to make quantifier alternations. In Section 7.4 the bounds on the succinctness function are applied to proving separations results for logical parameters of graphs, in particular, for $D(G)$ and $L(G)$.

7.1. **Three constructions.**

7.1.1. *First method: Padding.* We describe a "padding" operation that was invented by Joel Spencer (unpublished). It converts any graph G to an exponentially larger graph G^* with the logical depth larger just by 1. G^* includes G as an induced subgraph. In addition, for every subset X of $V = V(G)$, the graph G^* contains a vertex v_X. Denote the set of these vertices by V'. There is no edge inside V' but there are some edges between V and V'. Specifically, $v \in V$ is adjacent to v_X iff $v \in X$. In particular, $v_\emptyset$ is isolated and $N(v_V) = V$.

Vertex v_V will play a special role in our first-order definition of G^*. First of all, we will say that there is a vertex c (assuming $c = v_V$) whose neighborhood spans in G^* a subgraph isomorphic to G. This can be done by relativizing a formula Φ_G defining G to $N(c)$. That is, each universal quantification $\forall x(\Psi)$ in Φ_G has to be modified to

$$\forall_{x\in N(c)}(\Psi) \overset{\text{def}}{=} \forall x(x \sim c \to \Psi)$$

and each existential quantification to

$$\exists_{x\in N(c)}(\Psi) \overset{\text{def}}{=} \exists x(x \sim c \wedge \Psi).$$

Denote the relativized version of Φ_G by $\Phi_G|_{N(c)}$. Note that relativization does not change the quantifier depth. A sentence defining G^* can now look as follows:

$$\Phi_{G^*} \overset{\text{def}}{=} \exists c\Big(\Phi_G|_{N(c)} \wedge \forall_{x\notin N(c)}(N(x) \subset N(c)) \wedge \forall_{x_1\notin N(c)}\forall_{x_2\notin N(c)}(N(x_1) \ne N(x_2)) \\ \wedge \forall_{x_1\notin N(c)}\forall_{y\in N(x_1)}\exists_{x_2\notin N(c)}(N(x_2) = N(x_1) \setminus \{y\})\Big),$$

where we use harmless shorthands for simple first-order expressions.

It is easy to see that

$$D(\Phi_{G^*}) = \max\{D(\Phi_G), 4\} + 1$$

and that, if Φ_G is a $\exists^*\forall^*\exists^*\forall^*$-*formula* (that is, every chain of nested quantifiers is a string of this form), then Φ_{G^*} stays in this class as well. Consider now a sequence of graphs G_k where $G_1 = P_1$, the single-vertex graph, and $G_{k+1} = (G_k)^*$. Since $v(G^*) = v(G) + 2^{v(G)}$, we have $v(G_k) \geq Tower(k-1)$. It follows that $D_3(G_k) \leq \log^* v(G_k) + 3$.

7.1.2. *Second method: Unite and conquer.* Suppose that we have a set C of n-vertex graphs, each of logical depth at most d. Our goal is to construct a much larger set C^* of graphs with a much larger number of vertices n^* and logical depth bounded by $d+3$. An additional technical condition is that all the graphs have diameter 2. We know from Theorem 6.4 that almost all graphs on n vertices have logical depth less than $\log n$ and it is well known that they have diameter 2. Choosing a sufficiently large n, we can start with C being the class of all such graphs. Since almost all graphs are asymmetric, we have $|C| = (1-o(1))2^{\binom{n}{2}}$. Just for the notational simplicity, we prefer that $|C|$ is even.

For each $S \subset C$ such that $|S| = |C|/2$, the set C^* contains graph

$$G_S = \overline{\bigsqcup_{G \in S} G},$$

that is, we take the vertex disjoint union of all graphs in S and complement it. For convenience, we bound the logical depth of the complement $\overline{G_S} = \bigsqcup_{G \in C} G$ rather than that of G. Given an arbitrary $H \not\cong \overline{G_S}$, we analyze the Ehrenfeucht game on the two graphs.

If H has a connected component of diameter at least 3, Spoiler pebbles vertices u and v in H at the distance exactly 3 from one another. For Duplicator's responses u' and v' in $\overline{G_S}$, either $dist(u', v') \leq 2$ or $dist(u', v') = \infty$. In any case, Spoiler wins within the next 2 moves. Suppose from now on that all components of H have diameter at most 2. This condition allows us to assume that Duplicator respects the connectivity relation for otherwise Spoiler wins with one extra move (which will be added to the total count of rounds).

If one of the graphs, $\overline{G_S}$ or H, has a connected component A non-isomorphic to any component of the other graph, Spoiler pebbles a vertex in A. Let B be the component of the other graph where Duplicator responds. Starting from the second round, Spoiler plays the Ehrenfeucht game on non-isomorphic graphs A and B and wins in at most d moves.

If such a component does not exist, $\overline{G_S}$ must have a component A with at least two isomorphic copies in H. Then in the first two rounds Spoiler pebbles vertices in these two. Duplicator is forced at least once to respond in a component B of $\overline{G_S}$ non-isomorphic to A, which is an already familiar configuration.

Thus, $D(G_S)$ can be at most 3 larger than the maximum logical depth of graphs in C. At the same time G_S has the much larger number of vertices, namely $n^* = n|C|/2$. It follows that $D(G) < \log\log v(G)$ for any G in C^*.

Note that any graph in C^* is the complement of a disconnected graph and hence has diameter 2. This allows us to iterate the construction. Say, for any $G \in (C^*)^*$ we get $D(G) < \log\log\log v(G)$ and so on. If we fix the initial class C, the iteration procedure gives us graphs with $D(G) < 3\log^* v(G) + O(1)$. This bound is worse than in the preceding section but the extra factor of 3 can be eliminated if Spoiler plays somewhat more ingeniously (see [61]).

7.1.3. *Third method: Asymmetric trees.* The two previous examples were artificially constructed with the aim to ensure low quantifier depth. Now we present a natural class of graphs admitting succinct definability.

The *radius* of a graph G is defined by $r(G) = \min_{v\in V(G)} e(v)$, where $e(v)$ denotes the eccentricity of a vertex v. A vertex v is *central* if $e(v) = r(G)$. Any tree has either one or two central vertices (see, e.g., [58, Chapter 4.2]).

Lemma 7.1. *Let T be an asymmetric tree with $r(T) \ge 6$. Then $D(T) \le r(T) + 2$.*

Proof. We will design a strategy for Spoiler in the Ehrenfeucht game on T and a non-isomorphic graph T'. The reader that took the effort to reconstruct the proof of Theorem 5.3 will now definitely benefit.

We can assume that T and T' have equal diameters (in particular, T' is connected) for else Spoiler wins in less than $\log r(T)+4$ moves by Lemma 3.2. If T' is a non-tree, let Spoiler pebble a vertex v' on a cycle in T'. By this move Spoiler forces the game on $T\setminus v$ and $T'\setminus v'$, where v is Duplicator's response in T. If v is a leaf, Spoiler wins in two moves. Otherwise $T \setminus v$ is disconnected, while $diam(T' \setminus v') \le 3\, diam(T') \le 6\, r(T)$. Lemma 3.2 applies again and Spoiler wins in less than $\log r(T) + 6$ moves. Assume, therefore, that T' is a tree too.

Call a tree *diverging* if every vertex w splits it into pairwise non-isomorphic branches, where each branch is considered rooted at the respective neighbor of w (an isomorphism of rooted trees has to match their roots). Any asymmetric tree is obviously diverging. On the other hand, if a tree is diverging, it is either asymmetric or has a single nontrivial automorphism and the latter transposes two central vertices.

Suppose that T' is diverging. In the first round Spoiler pebbles a central vertex v of T and Duplicator responds with a vertex v' in T'. As it is easily seen, at least one of $T \setminus v$ and $T' \setminus v'$ has a branch B non-isomorphic to any branch in the other tree. Spoiler restricts further play to B by pebbling its root. Continuing in this fashion, that is, each time finding a matchless subbranch, Spoiler forces pebbling two paths in T and T' emanating from v and v' respectively. Spoiler wins at latest when the path in T reaches a leaf.

So suppose that T' is not diverging. Let v' be a central vertex of T' and u' be a vertex at the maximum possible distance from v' with the property that $T' \setminus u'$ has two isomorphic branches B' and B''. Spoiler pebbles the path from v' to u' and the two neighbors of u' in B' and B''. From this point Spoiler can play as before because B' and B'' are diverging and only one of them can be isomorphic to the corresponding branch pebbled by Duplicator in T. $\square$

Lemma 7.1 shows that asymmetric trees are definable with quantifier depth not much larger than their radius. On the other hand, asymmetric trees can grow in breadth, having a huge number of vertices. More precisely, there are asymmetric trees with $v(T) \geq \mathit{Tower}(r(T)-1)$. Indeed, let r_k denote the number of asymmetric rooted trees of height k. A simple recurrence

$$r_0 = r_1 = 1, \quad r_k = (2^{r_{k-1}} - 1)2^{\sum_{i=0}^{k-2} r_i}$$

shows that $r_k = \mathit{Tower}(k) - \mathit{Tower}(k-1)$ for all $k \geq 1$. Let T_k be the tree of radius $k+1$ with a single central vertex c such that the set of branches growing from c consists of all r_k pairwise non-isomorphic rooted trees of height k. (The reader will now surely recognize another instance of the unite-and-conquer method!) It is clear that T_k is asymmetric and $v(T_k) \geq r_k(k+1) + 1 > \mathit{Tower}(k)$. Combining it with Lemma 7.1, we obtain $D(T_k) \leq k+3 \leq \log^* v(T_k) + 2$.

With a little extra work, trees with low logical depth can be constructed on any given number of vertices. It turns out that the log-star bound is essentially the best what can be achieved for trees.

Theorem 7.2 (Pikhurko, Spencer, and Verbitsky [60])**.** *For every n there is a tree T on n vertices with $D(T) \leq \log^* n + 4$. On the other hand, for all trees T on n vertices we have $D(T) \geq \log^* n - \log^* \log^* n - O(1)$.*

We will see in the next section that the lower bound of Theorem 7.2 cannot be extended to the class of all graphs.

7.2. **The succinctness function.** Define the *succinctness function* by

$$q(n) = \min \{ D(G) : v(G) = n \} .$$

Since only finitely many graphs are definable with a fixed quantifier depth (see Theorem 2.3), we have $q(n) \to \infty$ as $n \to \infty$. The examples collected in Section 7.1 show that $q(n)$ increases rather slowly. Let $q_a(n)$ denote the version of $q(n)$ for definitions with at most a quantifier alternations. The padding construction from Section 7.1.1 gives us

$$q_3(n) \leq \log^* n + 3 \tag{23}$$

for infinitely many n and, by Theorem 6.15, this bound holds actually for all n, perhaps with a worst additive constant.

Is the log-star bound best possible? The answer is surprising enough: in some strong sense it is but, at the same time, it is very far from being tight. First, let us elaborate on the latter claim.

A *prenex formula* is a formula with all its quantifiers being in front. In this case there is a single sequence of nested quantifiers and the quantifier rank is just the number of quantifiers occurring in a formula. The superscript *prenex* will mean that we allow defining sentences only in prenex form. Thus, $q_a^{\mathit{prenex}}(n)$ is equal to the minimum quantifier depth of a prenex formula with at most a quantifier alternations that defines a graph on n vertices. We obviously have $D(G) \leq D_a(G) \leq D_a^{\mathit{prenex}}(G)$.

Recall that $L_a(G)$ denotes the minimum length of a sentence defining G with at most a quantifier alternations. Since a quantifier-free formulas with k variables is equivalent to a disjunctive normal form over $2\binom{k}{2}$ relations between the variables, we obtain also relation

$$L_a(G) = O(h(D_a^{prenex}(G))) \text{ where } h(k) = k^2 2^{k^2}. \tag{24}$$

Recall that a *general recursive function* is an everywhere defined recursive function.

Theorem 7.3 (Pikhurko, Spencer, and Verbitsky [60]). *There is no general recursive function f such that $f(q_3^{prenex}(n)) \geq n$ for all n.*

The theorem implies a superrecursive gap between $v(G)$ and $D_3(G)$ or even $L_3(G)$. In particular, the values of $q_3(n)$ are infinitely often inconceivably smaller even than the values of $\log^* n$. More generally, if a general recursive function $l(n)$ is monotone nondecreasing and tends to infinity, then

$$q(n) < l(n) \text{ for infinitely many } n, \tag{25}$$

which actually means that the succinctness function admits no reasonable lower bound.

The proof of Theorem 7.3 is based on simulation of a Turing machine M by a prenex formula Φ_M in which a computation of M determines a graph satisfying Φ_M and vice versa. Such techniques were developed in the classical research on Hilbert's *Entscheidungsproblem* by Turing, Trakhtenbrot, Büchi and other researchers (see [13] for survey and references). An important feature of our simulation is that it works if we restrict the class of structures to graphs. As a by-product, we obtain another proof of Lavrov's result [51] that the first-order theory of finite graphs is undecidable. The proof actually shows the undecidability of the $\forall^*\exists^p\forall^s\exists^t$-fragment of this theory for some p, s, and t.

We now have to explain why bound (23), though not sharp, is best possible in some sense. Let us define the *smoothed succinctness function* $q^*(n)$ to be the least monotone nondecreasing integer function bounding $q(n)$ from above, that is,

$$q^*(n) = \max_{m \leq n} q(m).$$

The following theorem shows that $q^*(n) = (1 + o(1)) \log^* n$ and, therefore, the log-star function is a nearly optimal *monotone* upper bound for the succinctness function $q(n)$.

Theorem 7.4 (Pikhurko, Spencer, and Verbitsky [60]).

$$\log^* n - \log^* \log^* n - 2 \leq q^*(n) \leq \log^* n + 4.$$

Though the lower bound contains a nonconstant lower order term, it can hardly be distinguished from a constant: for example, $\log^* \log^* n = 3$ for $n = 10^{80}$, which is a rough estimate of the number of elementary particles in the observable universe.

Proof. Theorem 7.2 implies that $q(n) \leq \log^* n + 4$ for all n. Since this bound is monotone, it is a bound on $q^*(n)$ as well. The lower bound for $q^*(n)$ can be derived from Theorem 2.3. According to it, at most $Tower(k + \log^* k + 2)$ graphs are definable with quantifier depth k. Given $n > \mathit{Tower}(3)$, let k be such that $\mathit{Tower}(k + 2 + \log^* k) < n \le \mathit{Tower}(k + 3 + \log^*(k + 1))$. It follows that $k > \log^* n - \log^* \log^* n - 4$. By the Pigeonhole Principle, there will be some $m \le n$ for which no graph of order precisely m is defined with quantifier depth at most k. We conclude that $q^*(n) \ge q(m) > k$ and hence $q^*(n) \ge \log^* n - \log^* \log^* n - 2$. □

We defined $q^*(n)$ to be the "closest" to $q(n)$ monotone function. Notice that $q(n)$ itself lacks the monotonicity, deviating from $q^*(n)$ infinitely often (set $l(n)$ to be the lower bound in Theorem 7.4 and apply (25)).

7.3. **Definitions with no quantifier alternation.** It is interesting to observe how the succinctness function changes when we put restrictions on the logic. Note that all what we have stated about the succinctness function for first-order logic actually holds true for its fragment with 3 quantifier alternations. Now we consider the first-order logic with no quantifier alternation, consisting of purely existential and purely universal formulas and their monotone Boolean combinations (of course, all negations are supposed to stay in front of relation symbols). It is easy to see that any sentence with no quantifier alternation is equivalent to a sentence in the *Bernays-Schönfinkel class*. The latter consists of prenex formulas in which the existential quantifiers all precede the universal quantifiers, as in

$$\Phi \stackrel{\text{def}}{=} \exists x_1 \ldots \exists x_k \forall y_1 \ldots \forall y_l \Psi(\bar{x}, \bar{y}), \tag{26}$$

where Ψ is quantifier-free. This fragment of first-order logic is provably weak.

To substantiate this claim, consider the *finite satisfiability problem*: Given a first-order sentence Φ about graphs, one has to decide whether or not there is a finite graph satisfying Φ. More generally, let $\mathrm{Spectrum}(\Phi)$ consist of all those n such that there is a graph on n vertices satisfying Φ. Thus, the problem is to decide whether $\mathrm{Spectrum}(\Phi)$ is nonempty.

The question about algorithmic solvability of the finite satisfiability problem for graphs is related to Hilbert's *Entscheidungsproblem*. Lavrov [51] (see also [25, Theorem 3.3.3]) proved that, in general, this problem is unsolvable even for sentences without equality. However, if we consider only sentences in the Bernays-Schönfinkel class, the problem becomes decidable. This directly follows from the following simple observation showing that a nonempty spectrum always contains a certain small number.

Lemma 7.5. *Suppose that a first-order sentence Φ is of the form (26). If Φ is satisfiable, then* $\mathrm{Spectrum}(\Phi)$ *contains k or a smaller number.*

Proof. Assume that Φ is true on a graph G with more than k vertices and let $U \subset V(G)$ be the set of vertices $x_1, \ldots, x_k$ whose existence is claimed by Φ. Note that the induced subgraph $G[U]$ satisfies Φ as well. □

The solvability of the finite satisfiability problem for the Bernays-Schönfinkel class was observed by Ramsey in [68]. He actually proved a stronger result and his famous combinatorial theorem appeared there as a technical tool. Recall that a set is *cofinite* if it has finite complement.

Theorem 7.6 (Ramsey [68])**.** *Any sentence about graphs Φ in the Bernays-Schönfinkel class has either finite or cofinite spectrum. More specifically, if Φ is of the form (26), then either* $\mathrm{Spectrum}(\Phi)$ *contains no number equal to or greater than $2^k 4^l$ or it contains all numbers starting from $k+l$.*

Proof. Assume that Φ is true on a graph G with at least $2^k 4^l$ vertices and let $U \subset V(G)$ consist of vertices $x_1, \ldots, x_k$ whose existence is claimed by Φ. Recall that *Ramsey number* $R(l)$ is equal to the minimum R such that every graph with R or more vertices contains a homogeneous set of l vertices. As it is well known, $R(l) < 4^l$. By the Pigeonhole Principle, $V(G) \setminus U$ contains a subset W of $R(l)$ vertices with the same neighborhood within U. Let X be a homogeneous set of l vertices in $G[W]$. Note that $G[U \cup X]$ satisfies Φ and that X is a set of l twins in this graph. Cloning the twins, we can obtain a graph that satisfies Φ and has any number of vertices larger than $k+l$. □

Note that this theorem allows us not only to decide whether $\mathrm{Spectrum}(\Phi)$ is empty but also to completely determine it.

After this small historical excursion, let us turn back to the definability with no quantifier alternation. First of all, note that even without quantifier alternation all graphs remain definable (see (3)) and, hence, the parameter $D_0(G)$ is well defined.

Theorem 7.7 (Pikhurko, Spencer, and Verbitsky [60])**.** *$D_0(G)$ is a computable parameter of a graph.*

Proof. Given $m \geq 0$, one can algorithmically construct a finite set U_m consisting of 0-alternating sentences of quantifier depth m so that every 0-alternating sentence of quantifier depth m has an equivalent in U_m. To decide if $D_0(G) \leq m$, for each sentence $\Upsilon \in U_m$ satisfied by G we have to check if Υ can be satisfied by another graph G'. We first reduce Υ to an equivalent statement Ψ in the Bernays-Schönfinkel class. Suppose that Ψ has k existential quantifiers. It suffices to test all $G' \not\cong G$ with at most $k+1$ vertices. Indeed, if Φ is true on a graph with more than $k+1$ vertices then, by the argument used to prove Lemma 7.5, Φ is as well true on its induced subgraphs with $k+1$ and k vertices (one of which is not isomorphic to G). □

We cannot prove anything similar for $D(G)$ or even $D_1(G)$. The proof of Theorem 7.7 is essentially based on the decidability of whether or not a 0-alternating sentence is defining for some graph. However, in general this problem is undecidable (see [60]).

For the logic with no quantifier alternation, the succinctness function has much more regular behavior.

Theorem 7.8 (Spencer, Pikhurko, and Verbitsky [61])**.**

$$\log^* n - \log^* \log^* n - 2 \leq q_0(n) \leq \log^* n + 22.$$

The lower bound has to be contrasted to Theorem 7.3. It gives us a kind of a quantitative confirmation of the fact that the 0-alternation fragment of first-order logic is strictly less powerful. The upper bound improves upon the alternation number in (23) attaining the optimum. The proof of this bound is based on the unite-and-conquer construction in Section 7.1.2, where more subtle analysis is needed in order to achieve the zero alternation number. All the details can be found in [61].

Proof of Theorem 7.8 (lower bound). Given n, denote $k = q_0(n)$ and fix a graph G on n vertices such that $D_0(G) = k$. The same relation between $L_a(G)$ and $D_a(G)$ as in Theorem 2.2 is proved in [61]. By this result, G is definable by a 0-alternating sentence Υ of length less than $\mathit{Tower}(k + \log^* k + 2)$. Convert Υ to an equivalent sentence Φ in the Bernays-Schönfinkel class and note that $D(\Phi) \le L(\Upsilon)$. By Lemma 7.5, Φ must be true on some graph with at most $D(\Phi)$ vertices. Since Φ is true only on G, we have

$$n \le D(\Phi) \le L(\Upsilon) < \mathit{Tower}(k + \log^* k + 2).$$

This implies that

$$\log^* n \le k + \log^* k + 2. \tag{27}$$

Suppose on the contrary to our claim that $k \le \log^* n - \log^* \log^* n - 3$. Then $\log^* k \le \log^* \log^* n$ and (27) implies that

$$\log^* n \le (\log^* n - \log^* \log^* n - 3) + \log^* \log^* n + 2,$$

which is a contradiction, proving the claimed bound. □

Using the lower bound of Theorem 7.8 and the absence of any recursive linkage between $q_3(n)$ and n, we are able to show a superrecursive gap between two parameters in the logical depth hierarchy

$$D(G) \le D_3(G) \le D_2(G) \le D_1(G) \le D_0(G).$$

Theorem 7.9 (Pikhurko, Spencer, and Verbitsky [60])**.** *There is no general recursive function f such that $D_0(G) \le f(D_3(G))$ for all graphs G.*

Proof. Assume that such an f exists. Let G_n be a graph for which $D_3(G_n) = q_3(n)$. Then

$$f(q_3(n)) = f(D_3(G_n)) \ge D_0(G_n) \ge q_0(n) \ge \log^* n - \log^* \log^* n - 2.$$

This implies that $\mathit{Tower}(2f(q_3(n))) \ge n$, contradictory to Theorem 7.3. □

We have seen weighty evidences that the 0-alternating sentences are strictly less expressive than the sentences of the same quantifier depth with quantifier alternations. It is quite surprising that, nevertheless, sometimes we can prove for $D_0(G)$ upper bounds which are just a little worse than the best known bounds for $D(G)$. The following results should be compared with Theorems 5.4, 5.8, and 6.4.

Theorem 7.10.

1. (Bohman et al. [11]) *Let $D_0(n, d)$ denote the maximum of $D_0(T)$ over all trees with n vertices and maximum degree at most $d = d(n)$. If both d and $\log n/ \log d$ tend to infinity, then $D_0(n, d) \le (1 + o(1))\frac{d \log n}{\log d}$.*
2. (Pikhurko, Veith, and Verbitsky [62]) $D_0(G, H) \le \frac{n+5}{2}$ *for all non-isomorphic graphs G and H with the same number of vertices n.*
3. (Kim et al. [48]) $D_0(G_{n,1/2}) \le (2 + o(1)) \log n$ *with high probability.*

We conclude this subsection with a demonstration of somewhat surprising strength of the Bernays-Schönfinkel class. We say that a sentence Φ *identifies* a graph G if it distinguishes G from any non-isomorphic graph of the same order. Let $BS(G)$ denote the minimum quantifier depth of Φ in the Bernays-Schönfinkel class identifying G.

We already discussed the identification problem in Section 5.2.1. Note, however, a striking difference. While in Section 5.2.1 we could make the conjunction of all sentences Φ_H distinguishing G from another graph H of the same order, now we have to distinguish G from all such H by a single prenex sentence!

Theorem 7.11 (Pikhurko and Verbitsky [64]).

1. *For any graph G of order n, we have $BS(G) \le \frac{3}{4}n + \frac{3}{2}$.*
2. *With high probability we have $BS(G_{n,1/2}) \le (2 + o(1)) \log n$. Moreover, the latter bound holds true even if the number of universal quantifiers in an identifying formula is restricted to 2.*

7.4. **Applications: Inevitability of the tower function.** Succinctly definable graphs can be used to show that the tower function is sometimes unavoidable in relations between logical parameters of graphs. We first observe that the relationship between the logical depth and the logical length in Theorem 2.2 is "nearly" tight.

Theorem 7.12 (Pikhurko, Spencer, and Verbitsky [60]).[6] *There are infinitely many pairwise non-isomorphic graphs G with $L(G) \ge Tower(D(G) - 7)$.*

Proof. The proof is given by a simple counting argument. A first-order sentence Φ defining a graph G determines a natural binary encoding of G (up to isomorphism) of length $O(L(\Phi) \log L(\Phi))$. It follows that at most $m = 2^{O(k \log k)}$ graphs can have logical depth less than k. By the Pigeonhole Principle, there is $n \le m + 1$ such that $L(G) \ge k$ for all G on n vertices. For all these graphs we have

$$L(G) \ge \log \log n, \tag{28}$$

if k is chosen sufficiently large. By Theorem 7.2, there is a graph G_n on n vertices with

$$D(G_n) < \log^* n + 5. \tag{29}$$

Combining (29) and (28), we obtain the desired separation of $L(G_n)$ from $D(G_n)$. Increasing the parameter k, we can have infinitely many such examples. □

One of the consequences of Theorem 7.3 is that prenex formulas are sometimes unexpectedly efficient in defining a graph. We are now able to show that, nevertheless, they generally cannot be competitive against defining formulas with no restriction on structure. More specifically, we have simple relations

$$D(G) \le D^{prenex}(G) < L(G) \le L^{prenex}(G). \tag{30}$$

Combining the second inequality with Theorem 2.2, we obtain

$$D^{prenex}(G) < Tower(D(G) + \log^* D(G) + 2)$$

and we can now see that this relationship between $D^{prenex}(G)$ and $D(G)$ is not so far from being optimal.

Corollary 7.13. *There are infinitely many pairwise non-isomorphic graphs G with $D^{prenex}(G) \ge Tower(D(G) - 8)$.*

[6] In [60] we stated a better bound $L(G) \ge Tower(D(G) - 6) - O(1)$, which was proved for the variant of $L(G)$ where variable x_i contributes $\log i$, rather than just 1, to the formula length.

The proof of Theorem 7.12 gives us actually a better bound, though somewhat cumbersome, namely $L(G) \geq T/(c \log T)$ with $T = \mathit{Tower}(D(G) - 6)$ and c a constant. Corollary 7.13 follows from here simply by noticing that parameters $D^{prenex}(G)$ and $L(G)$ are exponentially close. The latter fact follows from (30) and a version of (24), namely

$$L(G) = O(h(D^{prenex}(G))) \text{ where } h(x) = x^2 2^{x^2}.$$

In conclusion we note that the tower function is essential also in the upper bound for the number of graphs definable with quantifier depth k given by Theorem 2.3.

Corollary 7.14. *There are at least $(1 - o(1))\mathit{Tower}(k - 2)$ first-order sentences of quantifier depth k defining pairwise non-isomorphic graphs and, hence, being pairwise inequivalent.*

Proof. In Section 7.1.3 we noticed that there are exactly $r_h = \mathit{Tower}(h) - \mathit{Tower}(h - 1)$ non-isomorphic rooted trees of height h. It follows that there are at least $2^{r_h} - r_h - 1 = (1 - o(1))\mathit{Tower}(h + 1)$ asymmetric trees of radius $h + 1$. By Lemma 7.1, each of them is definable with quantifier depth $h + 3$. □

A lower bound of $\mathit{Tower}(k - 2)$ for the number of pairwise inequivalent sentences of quantifier depth k is shown by Spencer [71, Theorem 2.2.2].

8. Open problems

Many questions remain open, some of which are included in the main text of the survey alongside the known related results. For reader's convenience we collect a few open problems here that we consider most interesting.

Tomasz Łuczak (Conference on Random Structures and Algorithms, Poznań, 2003) asked if $D(G)$, or $W(G)$, is a computable function of the input graph G.

While the factor of $1/2$ in Theorem 5.8 is best possible, we do not know if it can be improved for logic with counting. Surprisingly, we could not resolve even the following question. Is there $\epsilon > 0$ such that for every graph G of sufficiently large order n we have $W_{\#}(G) \leq (\frac{1}{2} - \epsilon)n$?

Recall that no sublinear bound is generally possible here because Cai, Fürer, and Immerman [14] constructed graphs with linear width in the counting logic; see Theorem 5.7. Automorphisms of these graphs play an essential role in establishing this lower bound. It would be very interesting to estimate $W_{\#}(G)$ from above for asymmetric G. Again, we have only the bound $D_{\#}(G) \leq (n + 3)/2$ as a straightforward corollary of Theorem 5.8, where no restriction on the automorphism group is supposed.

Another research direction, with applications to the graph isomorphism problem, is identification of natural classes of graphs with $W_{\#}(G)$ bounded by a constant; see Sections 5.1.4 and 5.1.5. Recently, Laubner [50] proves such a bound for interval graphs. His approach is based on the fact that any maximal clique in an interval graph is definable as the common neighborhood of some two vertices. This prevents any straightforward extension to the class of circular-arc graphs, where the number of maximal cliques can be exponential.

A result of Dawar, Lindell, and Weinstein [19] (see also Theorem 4.7) implies an upper bound for $D_{\#}(G)$ in terms of $W_{\#}(G)$ and the order n of G, where $W_{\#}(G)$ disappointedly occurs at the exponent. Can this bound be improved? At the moment we cannot even exclude that $D_{\#}(G) = O(W_{\#}(G) \log n)$. If the latter bound was true for $D^k_{\#}(G)$ with $k = O(W_{\#}(G))$, this would have important consequences for isomorphism testing by Theorem 4.5.

Where do we need the power of counting quantifiers? To keep far away from the trivial example of a complete or empty graph, suppose that a graph G is asymmetric. Is it true or not that $W(G) = O(W_{\#}(G) \log n)$? A random graph shows that this bound would be best possible.

We are still far from having a complete evolutionary picture of the logical complexity for a random graph. Let $\delta \in (0, 1)$ be fixed and p be an arbitrary function of n with $n^{-\delta} \le p \le \frac{1}{2}$. Is it true that whp $D(G_{n,p}) = O(\log n)$?

The local behavior of the succinctness function $q(n)$, that was defined in Section 7.2, is unclear. While it is trivial that $q(n+1) \le q(n) + 1$, we do not know, for example, if $q(n+1) \ge q(n) - C$ for some constant C and all n.

In accordance with our notation system, let $q^k(n)$ denote the succinctness function for the k-variable logic. By slightly modifying the proof of Lemma 7.1, one can show that $q^3(n) \le (1 + o(1)) \log^* n$ for all n. Since the satisfiability problem for the 3-variable logic is undecidable (see, e.g., [32]), it is not excluded that an analog of Theorem 7.3 can be established for $q^3(n)$.

Given a fixed k, how far apart from one another can the values of $D(G)$ and $D^k(G) < \infty$ be?

Theorem 7.9 says that there is no recursive link between $D_3(G)$ and $D_0(G)$. Can one show a super-recursive gap between $D_a(G)$ and $D_b(G)$ for some $b > a > 0$ or, at least, between $D(G)$ and $D_1(G)$?

Though the case of trees was thoroughly investigated throughout the survey, this class of graphs deserves further attention. One may expect that many logical questions for trees are easier. Note in this respect that the first-order theory of finite trees is decidable due to Rabin [66]. Nevertheless, we do not know, for example, whether or not the logical depth $D(T)$ of a tree T is a computable parameter (while it is not hard to show that the logical width $W(T)$ is computable in logarithmic space).

Disappointingly, we were able to collect just a very few results on the logical length for this survey. From the fact that there are $2^{(1/2+o(1))\, n^2}$ non-isomorphic graphs of order n, it is easy to derive that whp $L(G_{n,1/2}) = \Omega\left(\frac{n^2}{\log n}\right)$. The obvious general upper bound is $O(n^2)$. This leaves open the question what the logical length of a typical graph is. Also, it would be very interesting to find explicit examples of graphs with large $L(G)$. Pseudo-random graphs can be natural candidates. For example, it is well known (Blass, Exoo, and Harary [9]) that Paley graphs share the first-order properties of a truly random graph. Techniques for estimating the length of a first-order formula are worked out, e.g., by Adler and Immerman [1], Dawar et al. [20], Grohe and Schweikardt [39].

Acknowledgment. The authors are thankful to Joel Spencer for the fruitful collaboration on the subject of this survey and for allowing them to use his unpublished ideas in the proof of Lemma 6.8 and in Section 7.1.1.

References

[1] M. Adler, N. Immerman. An $n!$ lower bound on formula size. *ACM Transactions on Computational Logic* 4:296–314 (2003).
[2] N. Alon, P. Seymour, R. Thomas. A separator theorem for nonplanar graphs. *J. Am. Math. Soc.* 3:801–808 (1990).
[3] N. Alon, J. Spencer. *The probabilistic method*. 3d ed., Wiley (2008).
[4] L. Babai. Automorphism groups, isomorphism, reconstruction. Chapter 27 of the *Handbook of Combinatorics,* 1447–1540. R. L. Graham, M. Grötschel, L. Lovász Eds. Elsevier Publ. (1995).
[5] L. Babai. On the complexity of canonical labeling of strongly regular graphs. *SIAM J. Comput.* 9:212–216 (1980).
[6] L. Babai, P. Erdős, S. M. Selkow. Random graph isomorphism. *SIAM J. Comput.* 9:628–635 (1980).
[7] L. Babai, L. Kučera. Canonical labeling of graphs in linear average time. In: *Proc. of the 20th IEEE Symp. Found. Computer Sci.* 39–46 (1979).
[8] L. Babai and E. M. Luks. Canonical labeling of graphs. In: *Proc. of the 15th ACM Symp. on Theory of Computing* 171–183 (1983).
[9] A. Blass, G. Exoo, F. Harary. Paley graphs satisfy all first-order adjacency axioms. *J. Graph Theory* 5:435–439 (1981).
[10] H. L. Bodlaender. Polynomial algorithms for Graph Isomorphism and Chromatic Index on partial k-trees. *Journal of Algorithms* 11:631–643 (1990).
[11] T. Bohman, A. Frieze, T. Luczak, O. Pikhurko, C. Smyth, J. Spencer, O. Verbitsky. first-order definability of trees and sparse random graphs. *Combinatorics, Probability and Computing* 16:375-400 (2007).
[12] B. Bollobás. *Random graphs*. 2d ed., Cambridge Univ. Press, 2001.
[13] E. Börger, E. Grädel, Y. Gurevich. *The classical decision problem.* Springer (1997).
[14] J.-Y. Cai, M. Fürer, N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica* 12:389–410 (1992).
[15] C. J. Colbourn, K. S. Booth. Linear time automorphism algorithms for trees, interval graphs, and planar graphs. *SIAM Journal on Computing* 10:203–225 (1981).
[16] T. Czajka, G. Pandurangan. Improved random graph isomorphism. *Journal of Discrete Algorithms* 6:85–92 (2008).
[17] B. Das, J. Torán, F. Wagner. Restricted space algorithms for isomorphism on bounded treewidth graphs. *Proc. of the 27th Ann. Symp. on Theoretical Aspects of Computer Science*, the Leibniz International Proceedings in Informatics series, 227–238 (2010).
[18] S. Datta, N. Limaye, P. Nimbhorkar, T. Thierauf, F. Wagner. Planar graph isomorphism is in Log-Space. In: *Proc. of the 24th Ann. Conf. on Computational Complexity*, 203–214 (2009).
[19] A. Dawar, S. Lindell, S. Weinstein, Infinitary logic and inductive definability over finite structures. *Information and Computation* 119:160–175 (1995).
[20] A. Dawar, M. Grohe, S. Kreutzer, N. Schweikardt. Model theory makes formulas large. In: *Proc. of the 34th Int. Colloquium on Automata, Languages and Programming.* Lecture Notes in Computer Science Vol. 4596, 913–924 (2007).
[21] R. Diestel. *Graph theory.* 2nd ed. Berlin: Springer (2000).
[22] R. G. Downey, M. R. Fellows. *Parameterized complexity.* Springer-Verlag (1998).
[23] H.-D. Ebbinghaus, J. Flum. *Finite model theory.* Springer Verlag, 2nd rev. ed. (1999).
[24] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundam. Math.* 49:129–141 (1961).

[25] Y. Ershov, I. Lavrov, A. Taimanov, M. Taitslin. Elementary theories (In Russian). *Uspekhi Matematicheskikh Nauk* 20:37–108 (1965). English translation in *Russian Math. Surveys* 20:35–105 (1965).

[26] K. Etessami, N. Immerman. Tree canonization and transitive closure. *Information and Computation* 157:2–24 (2000).

[27] R. Fagin. Probabilities on finite models. *J. Symb. Logic* 41:50–58 (1976).

[28] R. Fraïssé. Sur quelques classifications des systems de relations. *Publ. Sci. Univ. Alger* 1:35–182 (1954).

[29] M. Fürer. Weisfeiler-Lehman refinement requires at least a linear number of iterations. In: *Proc. of the 28th Int. Colloquium on Automata, Languages, and Programming.* Lecture Notes in Computer Science Vol. 2076, 322–333 (2001).

[30] Y. Glebskii, D. Kogan, M. Liogonkii and V. Talanov. Range and fraction of satisfiability of formulas in the restricted predicate calculus. *Kibernetika*, Kyiv, 2:17–28 (1969).

[31] E. Grädel. Finite model theory and descriptive complexity. In: *Finite Model Theory and Its Applications.* Texts in Theoretical Computer Science, an EATCS Series. Springer, pages 125–230 (2007).

[32] M. Grohe. Finite variable logics in descriptive complexity theory. *The Bulletin of Symbolic Logic* 4:345–398 (1998).

[33] M. Grohe. Fixed-point logics on planar graphs. In: *Proc. of the Ann. Conf. on Logic in Computer Science* 6–15 (1998).

[34] M. Grohe. Isomorphism testing for embeddable graphs through definability. In: *Proc. of the 32nd ACM Ann. Symp. on Theory of Computing,* 63–72 (2000).

[35] M. Grohe. Definable tree decompositions. In: *Proc. of the 23rd Ann. IEEE Symp. on Logic in Computer Science,* 406–417 (2008).

[36] M. Grohe. Fixed-point definability and Polynomial Time. In: *Computer Science Logic (CSL 2009). Proceedings.* E. Grädel, R. Kahle Eds. Lecture Notes in Computer Science, Vol. 5771, 20–23 (2009).

[37] M. Grohe. Fixed-point definability and polynomial time on chordal graphs and line graphs. E-print: *http://arxiv.org/abs/1001.2572* (2010).

[38] M. Grohe, J. Marino. Definability and descriptive complexity on databases of bounded tree-width. In: *Proc. of the 7th Int. Conf. on Database Theory*, Lecture Notes in Computer Science 1540, Springer-Verlag, 70–82 (1999).

[39] M. Grohe, N. Schweikardt. The succinctness of first-order logic on linear orders, *Logical Methods in Computer Science* 1(1:6):1–25 (2005).

[40] M. Grohe, O. Verbitsky. Testing graph isomorphism in parallel by playing a game. In: *Proc. of the 33rd Int. Colloquium on Automata, Languages, and Programming.* Lecture Notes in Computer Science Vol. 4051, 3–14 (2006).

[41] F. Harary. *Graph theory.* Addison-Wesley, Reading MA (1969).

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

[43] N. Immerman. *Descriptive complexity.* Springer-Verlag (1999).

[44] N. Immerman, D. Kozen. Definability with bounded number of bound variables. *Information and Computation* 83:121–139 (1989).

[45] N. Immerman, E. Lander. Describing graphs: a first-order approach to graph canonization. In: *Complexity theory retrospective*, pages 59–81. Springer, New York (1990).

[46] B. Jenner, J. Köbler, P. McKenzie, J. Torán. Completeness Results for Graph Isomorphism. *Journal of Computer and System Sciences* 66:549–566 (2003).

[47] R. M. Karp, V. Ramachandran. Parallel algorithms for shared-memory machines. In: *Algorithms and complexity. Handbook of theoretical computer science.* Vol. A. J. Van Leeuwen Ed. Amsterdam etc.: Elsevier Science Publishers 869–941 (1990).

[48] J.-H. Kim, O. Pikhurko, J. Spencer, O. Verbitsky. How complex are random graphs in first-order logic? *Random Structures and Algorithms* 26:119–145 (2005).

[49] P. Kővari, V.T. Sös, P. Turán". On a problem of K. Zarankiewicz. *Colloq. Math.* 3:50–57 (1954).
[50] B. Laubner. Capturing polynomial time on interval graphs. E-Print: *http://arxiv.org/pdf/0911.3799v1* (2009).
[51] I. Lavrov. Effective inseparability of the sets of identically true and finitely refutable formulae for certain elementary theories (In Russian). *Algebra i Logika* 2:5–18 (1963).
[52] S. Lindell. A logspace algorithm for tree canonization. In: *Proc. of the 24th Ann. ACM Symp. on Theory of Computing* 400–404 (1992).
[53] R. J. Lipton, R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math.* 36:177-189 (1979).
[54] G. L. Miller, J. H. Reif. Parallel tree contraction. Part 2: further applications. *SIAM Journal on Computing* 20:1128–1147 (1991).
[55] B. Mohar, C. Thomassen. *Graphs on surfaces.* The John Hopkins University Press (2001).
[56] J. W. Moon. On the maximum degree in a random tree. *Michigan Math. J.*, 15:429–432, 1968.
[57] M. Naor, A. Nussboim, E. Tromer. Efficiently constructible huge graphs that preserve first order properties of random graphs. In: *Theory of Cryptography.* Lecture Notes in Computer Science, Vol. 3378, 66–85 (2005).
[58] O. Ore. *Theory of graphs.* American Mathematical Society, Providence, R.I., 1962.
[59] E. Pezzoli. Computational complexity of Ehrenfeucht-Fraïssé games on finite structures. In: *Proc. of the 12th Conf. on Computer Science Logic 1998.* G. Gottlob, E. Grandjean, K. Seyr Eds. Lecture Notes in Computer Science 1584, Springer-Verlag, 159–170 (1999).
[60] O. Pikhurko, J. Spencer, O. Verbitsky. Succinct definitions in first-order graph theory. *Annals of Pure and Applied Logic* 139:74–109 (2006).
[61] O. Pikhurko, J. Spencer, O. Verbitsky. Decomposable graphs and definitions with no quantifier alternation. *European Journal of Combinatorics* 28:2264-2283 (2007).
[62] O. Pikhurko, H. Veith, O. Verbitsky. The first-order definability of graphs: upper bounds for quantifier depth. *Discrete Applied Mathematics* 154:2511–2529 (2006).
[63] O. Pikhurko, H. Veith, O. Verbitsky. The first-order definability of graphs: upper bounds for quantifier rank. E-print: *http://arxiv.org/abs/math.CO/0311041* (2003).
[64] O. Pikhurko, O. Verbitsky. Descriptive complexity of finite structures: saving the quantifier rank. *Journal of Symbolic Logic* 70:419–450 (2005).
[65] B. Poizat. Deux ou trois choses que je sais de L_n. *J. Symbolic Logic* 47:641–658 (1982).
[66] M. O. Rabin. Decidability of second order theories and automata on infinite trees. *Trans. of the AMS* 141:1–35 (1965).
[67] V. Ramachandran, J. Reif. Planarity testing in parallel. *J. Comput. Syst. Sci.* 49:517–561 (1994).
[68] F. Ramsey. On a problem of formal logic. *Proc. of the London Math. Soc. 2-nd series,* 30:264–286 (1930).
[69] N. Robertson, P.D. Seymour. Graph minors II. Algorithmic aspects of tree-width. *J. Algorithms* 7:309–322 (1986).
[70] W. L. Ruzzo. On uniform circuit complexity. *J. Comput. System Sci* 21:365–383 (1981).
[71] J. Spencer. *The strange logic of random graphs.* Springer Verlag (2001).
[72] J. Spencer, K. St. John. The complexity of random ordered structures. *Annals of Pure and Applied Logic* 152:174–179 (2008)
[73] D.A. Spielman. Faster isomorphism testing of strongly regular graphs. In: *Proc. of the 28th ACM Symp. on Theory of Computing* 576–584 (1996).
[74] O. Verbitsky. The first-order definability of graphs with separators via the Ehrenfeucht game. *Theoretical Computer Science* 343:158–176 (2005).
[75] O. Verbitsky. Planar graphs: logical complexity and parallel isomorphism tests. In: *Proc. of the 24th Ann. Symp. on Theoretical Aspects of Computer Science*, W. Thomas, P. Weil Eds. Lecture Notes in Computer Science Vol. 4393, 682–693 (2007).

[76] B. Yu. Weisfeiler, A. A. Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia, Seriya 2*, 9:12–16 (1968). In Russian.

[77] H. Whitney. 2-isomorphic graphs. *Amer. Math. J.* 55:245–254 (1933).