

# Compact Labelings For Efficient First-Order Model-Checking

Bruno Courcelle<sup>1,2</sup> and Cyril Gavoille<sup>1</sup> and Mamadou Moustapha Kanté<sup>1</sup>

*Université de Bordeaux, LaBRI, CNRS  
351 cours de la Libération  
33405 Talence, France*

---

## Abstract

We consider graph properties that can be checked from labels, i.e., bit sequences, of logarithmic length attached to vertices. We prove that there exists such a labeling for checking a first-order formula with free set variables in the graphs of every class that is *nicely locally clique-width-decomposable*. This notion generalizes that of a *nicely locally tree-decomposable* class. The graphs of such classes can be covered by graphs of bounded *clique-width* with limited overlaps. We also consider such labelings for *bounded* first-order formulas on graph classes of *bounded expansion*. Some of these results are extended to counting queries.

*Key words:* First-Order Logic; Labeling Scheme; Local Clique-Width; Local Tree-Width; Locally Bounded Clique-Width.

---

## 1 Introduction

The model-checking problem for a class of structures  $\mathcal{C}$  and a logical language  $\mathcal{L}$  consists in deciding for a given pair  $(S, \varphi)$  where  $S \in \mathcal{C}$  and  $\varphi \in \mathcal{L}$  if  $S \models \varphi$ , i.e., if  $S$  satisfies the property expressed by  $\varphi$ . More generally, if  $\varphi$  is a formula with free variables  $x_1, \dots, x_m$ , one may ask whether  $S$  satisfies  $\varphi(a_1, \dots, a_m)$  where  $a_1, \dots, a_m$  are values given to  $x_1, \dots, x_m$ . One may also wish to list the set of  $m$ -tuples  $(a_1, \dots, a_m)$  that satisfy  $\varphi$  in  $S$ , or simply count them.

Polynomial time algorithms for these problems (for fixed  $\varphi$ ) exist for certain classes of structures and certain logical languages. In this sense, graphs of

---

<sup>1</sup> Supported by the GRAAL project of “Agence Nationale pour la Recherche”.

<sup>2</sup> Member of “Institut Universitaire de France”.

bounded degree “fit” with *first-order* logic [13,30] and graphs of bounded tree-width or clique-width “fit” with *monadic second-order* logic. Frick and Grohe [16,17,19] have defined *Fixed-Parameter Tractable* algorithms for first-order model-checking problems on classes of graphs that may have unbounded degree and tree-width (definitions and examples are given in Section 4), and our results will concern such classes. We will also use graph classes of *bounded expansion*, a notion introduced by Nešetřil and Ossona de Mendez [28].

We will use similar tools for the following labeling problem. Given a class of graphs  $\mathcal{C}$  and a property  $P(x_1, \dots, x_m, Y_1, \dots, Y_q)$  of vertices  $x_1, \dots, x_m$  and of sets of vertices  $Y_1, \dots, Y_q$  of graphs in  $\mathcal{C}$ , the aim is to design two algorithms: an algorithm  $\mathcal{A}$  that attaches to each vertex  $x$  of a given graph of  $\mathcal{C}$  a label  $L(x)$ , defined as a sequence of 0’s and 1’s, and an algorithm  $\mathcal{B}$  that checks the property  $P(x_1, \dots, x_m, Y_1, \dots, Y_q)$  by using the labels and no other information about the considered graph. This latter algorithm takes as input the labels  $L(x_1), \dots, L(x_m)$  and the sets of labels<sup>3</sup>  $L(Y_1), \dots, L(Y_q)$  and tells whether  $P(x_1, \dots, x_m, Y_1, \dots, Y_q)$  is true. An *f-labeling scheme* for a class of structures  $\mathcal{C}$  is a pair  $(\mathcal{A}, \mathcal{B})$  of algorithms solving the labeling problem and using labels of length at most  $f(n)$  for the graphs of  $\mathcal{C}$  with  $n$  vertices. Results of this type have been established for monadic second-order logic by Courcelle and Vanicat [10], and for particular properties (connectivity queries, that are expressible in monadic second-order logic) by Courcelle and Twigg in [9] and by Courcelle et al. in [6].

Let us review the motivations for looking for *compact labelings*. By *compact*, we mean that labels have length of order less than  $O(n)$ , where  $n$  is the number of vertices of the graph, hence in particular of length  $\log^{O(1)}(n)$ . In distributed computing over a communication network with underlying graph  $G$ , nodes must act according to their local knowledge only. This knowledge can be updated by message passing. Due to space constraints on the local memory of each node, and on the sizes of messages, a distributed task cannot be performed by representing the whole graph  $G$  in each node or in each message. It must rather manipulate compact representations of  $G$ , distributed in a balanced way over the graph. For an example, the routing task may use routing tables that are sublinear in the size of  $G$  (preferably of poly-logarithmic size), and short addresses transmitted in the headers of messages (of poly-logarithmic size too). As surveyed in [20] many distributed tasks can be optimized by the use of labels attached to vertices. Such labels should be usable even when the network has node or link crashes. They can be constructed by *forbidden-set labeling schemes* as defined in [9]. In this framework, the local information can be updated by transmitting to all surviving nodes the list of (short) labels of all defected nodes and links, so that the surviving nodes can update their local information, e.g., their routing tables.

---

<sup>3</sup> Distinct vertices have distinct labels. We let  $L(Y)$  be the set  $\{L(x) \mid x \in Y\}$ .

Let us comment about using set arguments. The forbidden (or defective) parts of a network are handled as sets of vertices passed to a query as an argument. This means that the algorithm  $\mathcal{A}$  computes the labels once and for all, independently of the possible forbidden parts of the network. In other words the labeling supports node deletions from the given network. (Edge deletions are supported in the labelings of [6] and [9].) If the network is augmented with new nodes and links, the labels must be recomputed. We leave this incremental extension as a topic for future research. Set arguments can be used to handle deletions, but also constraints or queries like: “what are the nodes that are at distance at most 3 of  $X$  and  $Y$ ” where  $X$  and  $Y$  are two specified sets of nodes.

This article is organized as follows. In Section 2 we give some definitions regarding first-order logic and we define the notions of clique-width and of labeling schemes. Section 3 deals with first-order logic and the locality of first-order properties. In Section 4 we define the notions of *local bounded clique-width* and of *nicely locally cwd-decomposable* class of graphs (*cwd* abbreviates “clique-width”). We give some examples and preliminary results. Section 5 is devoted to the proofs of the main results. In Section 6 we extend some results to counting queries.

## 2 Definitions

Our results concern graph properties expressed by logical formulas, which assumes that graphs are defined as (or represented by) relational structures. All graphs and relational structures will be finite. We refer to [12] for our graph terminology.

A *relational signature* is a finite set  $\mathcal{R} := \{R, R', \dots\}$  of relation symbols, each of which given with an *arity*  $ar(R) \geq 1$ . A finite relational  $\mathcal{R}$ -*structure*  $S$  is defined as  $\langle D_S, (R_S)_{R \in \mathcal{R}} \rangle$  where  $R_S \subseteq D_S^{ar(R)}$ . The set  $D_S$  is called the *domain* of  $S$ . A relational signature  $\mathcal{R}$  is *binary* if  $ar(R) \leq 2$  for all  $R \in \mathcal{R}$ . A relational structure is binary if it is an  $\mathcal{R}$ -structure for some binary signature  $\mathcal{R}$ . We let  $\mathcal{R}_i$  be the set of symbols of arity  $i$ .

We now define *first-order* logic and *monadic second-order* logic on relational structures. Let  $\mathcal{R}$  be a relational signature. *Atomic* formulas over  $\mathcal{R}$  are  $x = y$ ,  $x \in X$  and  $R(x_1, \dots, x_{ar(R)})$  for all relations  $R$  in  $\mathcal{R}$ . A first-order formula (FO formula for short) over  $\mathcal{R}$  is a formula formed from atomic formulas over  $\mathcal{R}$  with Boolean connectives  $\wedge, \vee, \neg, \Rightarrow$  and first-order quantifications  $\exists x$  and  $\forall x$ . We may have free set variables. Monadic second-order formulas (MSO formulas for short) over  $\mathcal{R}$  are formed as FO formulas over  $\mathcal{R}$  by using also set quantifications  $\exists X, \forall X$ . By formulas (FO or MSO) we mean formulas written

with the signature appropriate for the considered relational structures. If the free variables of a formula  $\varphi$  are among  $x_1, \dots, x_m$  and  $Y_1, \dots, Y_q$ , we will write this formula  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$ . A *sentence* is a formula without free variables. We write  $S \models \varphi$  to mean that the sentence  $\varphi$  is satisfied by the relational structure  $S$ .

Our results will only concern binary relational structures because they correspond to colored graphs. We will be able to use notions like clique-width that do not apply to general relational structures.

Let  $\mathcal{R}$  be a binary relational signature with  $\mathcal{R}_1 := \{p_a \mid a \in C_1\}$ ,  $\mathcal{R}_2 := \{edg_a \mid a \in C_2\}$  and  $C_1 \cap C_2 = \emptyset$  (without loss of generality). A  $(C_1, C_2)$ -graph  $G$  (we will say a *colored graph* if we need not specify  $(C_1, C_2)$ ) is a directed graph such that every vertex has a possibly empty set of colors from  $C_1$ , every edge has a color in  $C_2$ , and no two edges from  $x$  to  $y$  (for any two vertices  $x$  and  $y$ ) have the same color. Such a graph is identified with the  $\mathcal{R}$ -structure  $\langle V_G, (edg_a)_{a \in C_2}, (p_a)_{a \in C_1} \rangle$  where  $V_G$  is the set of vertices,  $p_a(x)$  holds if and only if  $x$  has color  $a$  and  $edg_a(x, y)$  holds if and only if an edge from  $x$  to  $y$  is colored by  $a$ <sup>4</sup>. Every  $\mathcal{R}$ -structure corresponds in this way to a  $(C_1, C_2)$ -graph.

We will also use undirected graphs as tools to describe properties of  $(C_1, C_2)$ -graphs. If  $G$  is a colored graph, we let  $und(G)$  be the simple loop-free undirected graph such that  $V_{und(G)} := V_G$ , and two distinct vertices  $x$  and  $y$  in  $V_{und(G)}$  are adjacent in  $und(G)$  if and only if they are adjacent in  $G$ .

The *distance* of  $x$  and  $y$  in  $G$ , denoted by  $d_G(x, y)$ , is defined as their distance in  $und(G)$ , i.e., as the number of edges of a shortest (undirected) path linking them. If  $X$  is a subset of  $V_G$  and  $t$  a positive integer, we denote by  $N_G^t(X)$  the set  $\{y \in V_G \mid d_G(x, y) \leq t \text{ for some } x \text{ in } X\}$ . (Clearly,  $X \subseteq N_G^t(X)$  for every  $t$ .) We denote by  $G[X]$  the induced subgraph of  $G$  with vertex set  $X$  and colors inherited from  $G$  in the obvious way. We also denote by  $G - X$  the graph  $G[V_G \setminus X]$  and by  $G - x$  the graph  $G[V_G \setminus \{x\}]$  when  $x$  is a vertex of  $G$ . If  $m \geq 1$ , we denote by  $und(G)^m$  the simple loop-free undirected graph with vertex set  $V_G$ , and two distinct vertices  $x$  and  $y$  are adjacent if and only if  $d_G(x, y) \leq m$ .

A graph  $G$  has *arboricity* at most  $k$  if  $und(G)$  is the union of  $k$  edge-disjoint forests. Hence, this notion does not depend on loops, colors and edge directions. A class of graphs has arboricity at most  $k$  if each graph has arboricity at most  $k$ .

The *tree-width* [4] of a graph  $G$  depends only on  $und(G)$ . Tree-width is a graph parameter that yields many algorithmic properties surveyed by Grohe [19] and Kreutzer [25]. The survey [5] by Bodlaender presents tree-width and

<sup>4</sup> We allow multiple edges with different colors.

recent developments about this notion. *Clique-width* [8] is another graph parameter that yields interesting algorithmic results. It is sensitive to colors and directions of edges. The original definition of clique-width in [8] concerns only uncolored graphs. However, it can be easily extended to colored graphs [3,15].

**Definition 2.1 (Clique-Width of Colored Graphs)** *We let  $C$  be the pair  $(C_1, C_2)$  of sets of colors for vertices and edges. In order to construct  $(C_1, C_2)$ -graphs, we will use the set  $[k] := \{1, 2, \dots, k\}$ , for  $k \geq 1$ , to color also vertices, with one and only one color for each vertex<sup>5</sup>. A  $k$ - $C$ -graph (or  $k$ -graph if  $C_1 = \emptyset$ ,  $|C_2| = 1$ )  $G$  is defined as  $\langle V_G, (edg_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1}, lab_G \rangle$  where  $lab_G : V_G \rightarrow [k]$  is a total function and the other components are as defined above. We define several operations on  $k$ - $C$ -graphs.*

- (1) *For  $k$ - $C$ -graphs  $G$  and  $H$  such that  $V_G \cap V_H = \emptyset$ , we define  $G \oplus H$  as the  $k$ - $C$ -graph  $K := \langle V_G \cup V_H, (edg_{aK})_{a \in C_2}, (p_{aK})_{a \in C_1}, lab_K \rangle$  where*

$$p_{aK}(x) := \begin{cases} p_{aG}(x) & \text{if } x \in V_G, \\ p_{aH}(x) & \text{if } x \in V_H, \end{cases} \quad \text{for all } a \in C_1.$$

$$edg_{aK}(x, y) := \begin{cases} edg_{aG}(x, y) & \text{if } x, y \in V_G, \\ edg_{aH}(x, y) & \text{if } x, y \in V_H, \end{cases} \quad \text{for all } a \in C_2.$$

$$lab_K(x) := \begin{cases} lab_G(x) & \text{if } x \in V_G, \\ lab_H(x) & \text{if } x \in V_H. \end{cases}$$

*If  $V_G \cap V_H \neq \emptyset$ , we replace  $H$  by an isomorphic copy  $H'$  such that  $V_H \cap V_{H'} = \emptyset$ . The graph  $G \oplus H$  is well-defined up to isomorphism. We call  $G \oplus H$  the disjoint union of  $G$  and  $H$ .*

- (2) *For a  $k$ - $C$ -graph  $G$ , for a color  $b$  in  $C_2$ , and for distinct  $i, j \in [k]$ , we define  $\eta_{i,j}^b(G)$  as the  $k$ - $C$ -graph  $K := \langle V_G, (edg_{aK})_{a \in C_2}, (p_{aG})_{a \in C_1}, lab_G \rangle$  where*

$$edg_{aK} := \begin{cases} edg_{aG} & \text{if } a \neq b, \\ edg_{bG} \cup \{(x, y) \mid x, y \in V_G, x \neq y \text{ and } lab_G(x) = i, lab_G(y) = j\} & \text{if } a = b. \end{cases}$$

*This operation adds  $b$ -colored edges from each vertex colored by  $i$  to each vertex colored by  $j$ . But, no such edge is added when it already exists.*

- (3) *For a  $k$ - $C$ -graph  $G$ , and for distinct  $i, j \in [k]$ , we define  $\rho_{i \rightarrow j}(G)$  as the  $k$ - $C$ -graph  $K := \langle V_G, (edg_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1}, lab_K \rangle$  where*

$$lab_K(x) := \begin{cases} j & \text{if } lab_G(x) = i, \\ lab_G(x) & \text{otherwise.} \end{cases}$$

<sup>5</sup> We assume that  $(C_1 \cup C_2) \cap [k] = \emptyset$ .

This operation changes each color  $i$  into  $j$ .

- (4) For each  $i \in [k]$  and each  $A \subseteq C_1 \cup C_2$ , the constant symbol  $\mathbf{i}_A$  denotes a  $k$ - $C$ -graph with a single vertex  $x$  with  $\text{lab}_{\mathbf{i}_A}(x) = i$  and such that  $p_{a\mathbf{i}_A}(x)$  holds if and only if  $a \in A \cap C_1$ , and  $\text{edg}_{a\mathbf{i}_A}(x, x)$  holds if and only if  $a \in A \cap C_2$ . We let  $C_{C,k}$  be the set  $\{\mathbf{i}_A \mid i \in [k], A \subseteq C_1 \cup C_2\}$ .

We let  $F_{C,k}$  be the set  $\{\oplus, \eta_{i,j}^a, \rho_{i \rightarrow j} \mid i, j \in [k], a \in C_2\}$ . Each term  $t$  in  $T(F_{C,k}, C_{C,k})$  has a value  $\text{val}(t)$ : it is the  $k$ - $C$ -graph obtained by evaluating  $t$  according to clauses (1)-(4). The clique-width of a (colored) graph  $G$ , denoted by  $\text{cwd}(G)$ , is the minimum  $k$  such that  $G$  is isomorphic to  $\text{val}(t)$  for some term  $t$  in  $T(F_{C,k}, C_{C,k})$ . We write  $F_k$  and  $C_k$  if  $C_1 = \emptyset$  and  $|C_2| = 1$ .

We recall two statements that are easy adaptations of similar results proved in [8]. We denote by  $\text{twd}(G)$  the tree-width of a graph  $G$ .

**Fact 2.2 ([8])** *Let  $(C_1, C_2)$  be a pair of sets of colors and let  $G$  and  $H$  be two  $(C_1, C_2)$ -graphs.*

- (i) *If  $H$  is an induced subgraph of  $G$ , then  $\text{cwd}(H) \leq \text{cwd}(G)$ .*
- (ii) *If  $G$  and  $H$  are disjoint, then  $\text{cwd}(G \oplus H) = \max\{\text{cwd}(G), \text{cwd}(H)\}$ .*
- (iii) *If  $\text{twd}(G) = k$ , then  $\text{cwd}(G) \leq f(k, |C_2|)$  for some function  $f$ .*

One cannot exchange  $\text{twd}$  and  $\text{cwd}$  in Fact 2.2 (iii) because cliques have clique-width 2 and unbounded tree-width. For fixed  $k$ , there exists a cubic-time algorithm that given an undirected graph  $G$ , either outputs that it has clique-width at least  $k + 1$  or outputs a term  $t$  in  $T(F_{k'}, C_{k'})$  that defines  $G$  with  $k' = 2^{k+1} - 1$  [22,29]. This algorithm can be adapted to colored graphs with  $k' = g(k)$  for some function  $g$  [24]. Also, every property expressible in MSO logic can be checked in cubic-time for colored graphs in classes of bounded clique-width by combining the results of [7] and of [22,24,29]. The survey by Kamiński et al. [23] presents recent results on clique-width.

We now define the notion of *bounded expansion* [28]. As tree-width, it is independent of colors of vertices and/or edges. Graph classes with *bounded expansion* have several equivalent characterizations. We will use the following one.

**Definition 2.3 (Bounded Expansion)** *A class  $\mathcal{C}$  of colored graphs has bounded expansion if for every integer  $p$ , there exists a constant  $N$  such that for every colored graph in  $\mathcal{C}$ , one can partition its vertex set in at most  $N$  parts such that any  $i$  parts, for  $i \leq p$ , induce a subgraph of tree-width at most  $i - 1$ . We will denote  $N$  by  $N(\mathcal{C}, p)$ .*

The case  $i = 1$  of Definition 2.3 implies that each part is an independent set (no two distinct vertices are adjacent), hence the corresponding partition can

be seen as a *proper vertex-coloring*. We finish these preliminary definitions by introducing the notion of *labeling scheme*.

**Definition 2.4 (Labeling Scheme)** *Let  $\mathcal{R}$  be a relational signature and  $S$  an  $\mathcal{R}$ -structure. A labeling of  $S$  is an injective mapping  $J : D_S \rightarrow \{0, 1\}^*$  (or into some more convenient set, like  $\Sigma^*$  where  $\Sigma$  is a finite alphabet). If  $Y$  is a subset of  $D_S$ , we let  $J(Y)$  be the set  $\{J(y) \mid y \in Y\}$ . Hence,  $J(Y) \neq J(Y')$  if  $Y \neq Y'$ .*

*Let  $\varphi(\bar{x}, \bar{Y})$  be an MSO formula over  $\mathcal{R}$  where  $\bar{x}$  is an  $m$ -tuple of FO variables and  $\bar{Y}$  a  $q$ -tuple of set variables. Let  $\mathcal{C}$  be a class of  $\mathcal{R}$ -structures and  $f : \mathbb{N} \rightarrow \mathbb{N}$  an increasing function. An  $f$ -labeling scheme supporting the query defined by  $\varphi$  in the  $\mathcal{R}$ -structures of  $\mathcal{C}$  is a pair  $(\mathcal{A}, \mathcal{B})$  of algorithms doing the following:*

- (1)  $\mathcal{A}$  constructs for each  $S$  in  $\mathcal{C}$ , a labeling  $J$  of  $S$  such that  $|J(a)| = O(f(n))$  for every  $a \in D_S$ , where  $n = |D_S|$ .
- (2)  $\mathcal{B}$  takes as input an  $(m + q)$ -tuple  $(J(a_1), \dots, J(a_m), J(W_1), \dots, J(W_q))$  and says correctly whether  $S \models \varphi(\bar{a}, \bar{W})$ .

If we have constructed  $f$ -labelings  $J_1, \dots, J_p$  for formulas  $\varphi_1, \dots, \varphi_p$ , then  $J(x)$  defines as  $(J_1(x), \dots, J_p(x))$  is an  $f$ -labeling that supports  $\varphi_1, \varphi_2, \dots, \varphi_p$ , and clearly, any Boolean combination of these formulas.

Labeling schemes based on logical descriptions of queries by MSO formulas have been first defined by Courcelle and Vanicat [10] for graphs of bounded clique-width (whence also of bounded tree-width). We recall this result. If  $\bar{W}$  is a  $q$ -tuple of sets, we let  $|\bar{W}| = |W_1| + \dots + |W_q|$  and if  $\bar{a}$  is an  $m$ -tuple of vertices, we let  $|\bar{a}| = m$ .

**Theorem 2.5** *Let  $k$  be a positive integer and  $(C_1, C_2)$  a pair of disjoint sets. For every MSO formula  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  over the binary signature corresponding to  $(C_1, C_2)$ , there exists a log-labeling scheme  $(\mathcal{A}, \mathcal{B})$  for  $\varphi$  on the class of  $(C_1, C_2)$ -graphs of clique-width at most  $k$ . Moreover, if the input  $(C_1, C_2)$ -graph has  $n$  vertices, the algorithm  $\mathcal{A}$  computes the labels  $J(x)$  of all vertices  $x$  in time  $O(n^3)$  or in time  $O(n \cdot \log(n))$  if the clique-width expression of the graph is given. Given  $J(a_1), \dots, J(a_m)$  and  $J(W_1), \dots, J(W_q)$ , the algorithm  $\mathcal{B}$  checks whether  $\varphi(\bar{a}, \bar{W})$  holds in time  $O(\log(n) \cdot (|\bar{W}| + 1))$ . For  $n$ -vertex  $(C_1, C_2)$ -graphs of tree-width at most  $k$ , a variant of algorithm  $\mathcal{A}$  can build the labelings in time  $O(n \cdot \log(n))$ .*

The proof of Theorem 2.5 combines the construction of [10] that works for graphs given with their decompositions, and the parsing results by Bodlaender [4] for tree-width, and by Hliněný, Oum and Seymour [22,29] and Kanté [24] for clique-width (discussed above). Labeling schemes for distance and connectivity queries in graphs of bounded clique-width and in planar graphs have been given respectively by Courcelle and Twigg in [9] and by Courcelle et al. in [6].

In the present article, we consider graph classes of unbounded clique-width, and in particular, graph classes that are *locally decomposable* [16,17] and graph classes of bounded expansion [28]. So, MSO logic is out of reach for such classes and we will consider FO logic over colored graphs, i.e., binary relational structures.

### 3 Bounded and Local First-Order Formulas

The definitions given below concern binary relational structures called, from now on, graphs since they correspond to colored graphs as explained in Section 2. Formulas are written over fixed binary relational signatures that we do not specify all the time.

We still call *sentence* an FO formula without free FO variables but, that can have free set variables. These set variables will be used in a particular way and will be called *parameters*.

We are interested in on-line checking properties of networks in case of (reported) failures of some nodes (nodes are vertices of the associated graphs). Hence, for each property of interest, defined by a formula  $\varphi(x_1, \dots, x_m)$ , we are not only interested in checking if  $G \models \varphi(a_1, \dots, a_m)$  by using  $J(a_1), \dots, J(a_m)$  for  $a_1, \dots, a_m \in V_G$ , but also in checking if  $G - W \models \varphi(a_1, \dots, a_m)$  by using  $J(a_1), \dots, J(a_m)$  and  $J(W)$  where  $W$  is a subset of  $V_G \setminus \{a_1, \dots, a_m\}$ . We handle this formally as follows. We let  $\perp$  be a new vertex color. If  $W \subseteq V_G$ , we let  $G_W$  be the graph  $G$  equipped with an additional vertex-color  $\perp$ , i.e., as the structure  $G$  expanded with a unary relation  $p_\perp$  such that  $p_\perp(z)$  holds if and only if  $z \in W$ . The property  $G - W \models \varphi(a_1, \dots, a_m)$ , for an FO formula  $\varphi(x_1, \dots, x_m)$ , is equivalent to  $G \models \varphi'(a_1, \dots, a_m, W)$  and to  $G_W \models \varphi''(a_1, \dots, a_m)$  for FO formulas  $\varphi'(x_1, \dots, x_m, Y)$  and  $\varphi''(x_1, \dots, x_m)$  that are easy to write.

**Definition 3.1 (Bounded Formulas)** *An FO formula  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  is a basic bounded formula if, for some positive integer  $p$ , called a bound on the quantification space, we have the following equivalence for all graphs  $G$ , all  $a_1, \dots, a_m \in V_G$  and all  $W_1, \dots, W_q \subseteq V_G$ :*

$$G \models \varphi(a_1, \dots, a_m, W_1, \dots, W_q) \quad \text{if and only if}$$

*there exists  $X \subseteq V_G$  with  $|X| \leq p$  such that  $a_1, \dots, a_m \in X$  and*

$$G[X] \models \varphi(a_1, \dots, a_m, W_1 \cap X, \dots, W_q \cap X).$$

*An FO formula is bounded if it is a Boolean combination of basic bounded*



formulas.

If  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  is a basic bounded formula and  $G[Z] \models \varphi(a_1, \dots, a_m, W_1 \cap Z, \dots, W_q \cap Z)$ , then  $G[Y] \models \varphi(a_1, \dots, a_m, W_1 \cap Y, \dots, W_q \cap Y)$  for every  $Y \supseteq Z$ .

The negation of a basic bounded formula is not (in general) basic bounded, but it is bounded. The property that a graph has a subgraph isomorphic to a fixed graph  $H$  is expressible by a bounded formula.

**Definition 3.2 (Local Formulas)** *An FO formula  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  is  $t$ -local around  $(x_1, \dots, x_m)$  if for every graph  $G$ , for all  $a_1, \dots, a_m$  in  $V_G$ , and for all subsets  $W_1, \dots, W_q$  of  $V_G$ ,*

$G \models \varphi(a_1, \dots, a_m, W_1, \dots, W_q)$  *if and only if*  $G[N] \models \varphi(a_1, \dots, a_m, W_1 \cap N, \dots, W_q \cap N)$

where  $N := N_G^t(\{a_1, \dots, a_m\})$ .

An FO sentence  $\varphi(Y_1, \dots, Y_q)$  is basic  $(t, s)$ -local if it is equivalent to a sentence of the form

$$\exists x_1 \cdots \exists x_s \left( \bigwedge_{1 \leq i < j \leq s} d(x_i, x_j) > 2t \wedge \bigwedge_{1 \leq i \leq s} \psi(x_i, Y_1, \dots, Y_q) \right)$$

where  $\psi(x, Y_1, \dots, Y_q)$  is  $t$ -local around its unique free variable  $x$ .

**Remark 3.3** *The property  $d(x, y) \leq r$  is basic bounded (for  $p := r + 1$ ) and  $t$ -local for  $t := \lfloor r/2 \rfloor$ . Its negation  $d(x, y) > r$  is  $t$ -local and bounded (but not basic bounded).*

We now recall a decomposition of FO formulas into  $t$ -local and basic  $(t', s)$ -local formulas due to Gaifman [18].

**Theorem 3.4 ([18,26])** *Every FO formula  $\varphi(\bar{x}, \bar{Y})$  is logically equivalent to a Boolean combination  $B(\varphi_1(\bar{u}_1, \bar{Y}), \dots, \varphi_p(\bar{u}_p, \bar{Y}), \psi_1(\bar{Y}), \dots, \psi_h(\bar{Y}))$  where:*

- each formula  $\varphi_i$  is a  $t$ -local formula around some sub-sequence  $\bar{u}_i$  of  $\bar{x}$ ,
- each sentence  $\psi_i$  is a basic  $(t', s)$ -local sentence.

Moreover,  $B, \varphi_1, \dots, \varphi_p$  and  $\psi_1, \dots, \psi_q$  can be computed effectively from  $\varphi$ , and the integers  $t, t'$  and  $s$  can be bounded in terms of  $|\bar{x}|$  and of the quantifier-rank of  $\varphi$ .

This theorem is usually stated and proved for FO formulas without free set variables. However, in an FO formula a set variable  $Y_i$  occurs in atomic formulas of the form “ $y \in Y_i$ ”. This is equivalent to “ $R_i(y)$ ” if  $R_i$  is a unary relation representing  $Y_i$ . We denote by  $\varphi'(\bar{x})$  the formula obtained from  $\varphi(\bar{x}, Y_1, \dots, Y_q)$

by replacing every subformula “ $y \in Y_i$ ” by “ $R_i(y)$ ”. In order to prove that two FO formulas  $\varphi(\bar{x}, Y_1, \dots, Y_q)$  and  $\psi(\bar{x}, Y_1, \dots, Y_q)$  are equivalent in every relational structure of a class  $\mathcal{C}$  of  $\mathcal{R}$ -structures, it is enough to prove that the corresponding formulas  $\varphi'(\bar{x})$  and  $\psi'(\bar{x})$  are equivalent in every structure  $S'$  that is an expansion of a structure  $S$  in  $\mathcal{C}$  by unary relations  $R_1, \dots, R_q$ . Hence, Theorem 3.4 follows from its usual formulation for FO formulas without free set variables. The same holds for Theorem 3.5 below.

We will use a stronger form of Theorem 3.4 from [16] that decomposes  $t$ -local formulas. Let  $m, t \geq 1$ . The  $t$ -distance type of an  $m$ -tuple  $\bar{a}$  of elements of a structure  $S$  is the undirected graph  $\Delta(\bar{a}) := ([m], \text{edg}_{\Delta(\bar{a})})$  where  $\text{edg}_{\Delta(\bar{a})}(i, j)$  holds if and only if  $d_S(a_i, a_j) \leq 2t + 1$ . For each undirected graph  $\Delta$ , the property that an  $m$ -tuple  $\bar{a}$  satisfies  $\Delta(\bar{a}) := \Delta$  can be expressed by a  $t$ -local formula  $\rho_{t, \Delta}(x_1, \dots, x_m)$  equivalent to:

$$\bigwedge_{(i,j) \in \text{edg}_{\Delta}} d(x_i, x_j) \leq 2t + 1 \quad \wedge \quad \bigwedge_{(i,j) \notin \text{edg}_{\Delta}} d(x_i, x_j) > 2t + 1.$$

**Theorem 3.5 ([16])** *Let  $\varphi(\bar{x}, \bar{Y})$  be  $t$ -local around the  $m$ -tuple  $\bar{x}$  (where  $m \geq 1$ ), and with  $\bar{Y} := (Y_1, \dots, Y_q)$ . For each undirected graph  $\Delta$  with vertex set equal to  $[m]$  and connected components  $\Delta_1, \dots, \Delta_p$ , one can compute a Boolean combination  $F^{t, \Delta}(\varphi_{1,1}, \dots, \varphi_{1,j_1}, \dots, \varphi_{p,1}, \dots, \varphi_{p,j_p})$  of formulas  $\varphi_{i,j}$  with free variables in  $\bar{x}$  and in  $\bar{Y}$  such that:*

- the free FO variables of each  $\varphi_{i,j}$  belong to  $\bar{x} \upharpoonright \Delta_i$ <sup>6</sup>,
- $\varphi_{i,j}$  is  $t$ -local around  $\bar{x} \upharpoonright \Delta_i$ ,
- for each  $m$ -tuple  $\bar{a}$  and each  $q$ -tuple of sets  $\bar{W}$ ,  $G \models \rho_{t, \Delta}(\bar{a}) \wedge \varphi(\bar{a}, \bar{W})$  if and only if  $G \models \rho_{t, \Delta}(\bar{a}) \wedge F^{t, \Delta}(\dots, \varphi_{i,j}(\bar{a} \upharpoonright \Delta_i, \bar{W}), \dots)$ .

A *query* is defined by a formula  $\varphi(x_1, \dots, x_m)$  without set variables. A *parametrized query* is defined by a formula  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  where  $Y_1, \dots, Y_q$  are the parameters.

## 4 Locally Decomposable Classes

We generalize notions defined by Frick and Grohe and we will use the same notations as in [16,17]. Definition 4.1 is analogous to [17, Definition 5.1].

### Definition 4.1 (Local Clique-Width)

<sup>6</sup> If  $\bar{x} := (x_1, \dots, x_m)$  and  $\{i_1, \dots, i_r\}$  is the vertex set of  $\Delta_i$  with  $i_1 < i_2 < \dots < i_r$ , then  $\bar{x} \upharpoonright \Delta_i := (x_{i_1}, \dots, x_{i_r})$ .

- (1) The local clique-width of a graph  $G$  is the function  $lcwd^G : \mathbb{N} \rightarrow \mathbb{N}$  defined by  $lcwd^G(t) := \max\{cwd(G[N_G^t(a)]) \mid a \in V_G\}$ .
- (2) A class  $\mathcal{C}$  of graphs has bounded local clique-width if there is a function  $f : \mathbb{N} \rightarrow \mathbb{N}$  such that  $lcwd^G(t) \leq f(t)$  for every  $G \in \mathcal{C}$  and  $t \in \mathbb{N}$ .

### Examples of Graphs of Bounded Local Clique-Width

- (1) Every class of graphs of bounded clique-width has also bounded local clique-width since  $cwd(G[A]) \leq cwd(G)$  for every  $A \subseteq V_G$  (Fact 2.2 (i)).
- (2) The classes of graphs of bounded local tree-width have bounded local clique-width since the clique-width of a graph is bounded by a function of its tree-width (Fact 2.2 (iii)). Bounded degree graph classes and minor-closed classes of graphs that exclude some apex-graph<sup>7</sup> as a minor are examples of graph classes of bounded local tree-width (see [16,17]).
- (3) Let  $m$  be a positive integer and  $\mathcal{C}$  a class of graphs of bounded local clique-width. Then  $\mathcal{C}^m := \{G^m \mid G \in \mathcal{C}\}$  has bounded local clique-width. Let us sketch the proof. Let  $G$  be a graph in  $\mathcal{C}$ . For every vertex  $x$  of  $G$  and every positive integer  $r$ , we have  $N_{G^m}^r(x) = N_G^{rm}(x)$ . Let us verify that  $G^m[N_{G^m}^r(x)] = G^m[N_G^{rm}(x)]$  where  $G^m := (G[N_G^{(r+1)m}(x)])^m$ . Since  $G^m \subseteq G^m$ , we have  $G^m[N_{G^m}^r(x)] \supseteq G^m[N_G^{rm}(x)]$ . For the other inclusion, consider  $y$  and  $z$  in  $N_{G^m}^r(x)$  such that  $d_G(y, z) \leq m$ . Any path of length at most  $m$  that links them is in  $G[N_G^{(r+1)m}(x)]$ , hence  $y$  and  $z$  are adjacent in  $G^m := (G[N_G^{(r+1)m}(x)])^m$ , which gives the desired equality. It is proved in [31]<sup>8</sup> that if a graph  $H$  has clique-width  $k$ , then  $H^m$  has clique-width at most  $4 \cdot (m+1)^k$ . Hence, for every graph  $G$  in  $\mathcal{C}$  and every positive integer  $r$ ,  $lcwd^{G^m}(r) \leq 4 \cdot (m+1)^{f(r(m+1))}$  where  $f$  is the function that bounds the local clique-width of graphs in  $\mathcal{C}$ .
- (4) If  $G$  is a graph, we denote by  $K := Line(G)$  the *line graph* of  $G$  defined as the undirected graph with vertex set equal to the set of edges of  $G$  and such that  $edg_K(e, e')$  holds if and only if the edges  $e$  and  $e'$  are incident in  $G$ . We claim that if a class  $\mathcal{C}$  of graphs has bounded local tree-width, then  $Line(\mathcal{C}) := \{Line(G) \mid G \in \mathcal{C}\}$  has bounded local clique-width. Let  $G$  be a graph in  $\mathcal{C}$  and let  $K$  be its line graph. For every  $e$  and  $e'$  in  $E_G = V_K$ , we have  $d_G(x, y) \leq d_K(e, e') + 1$  if  $x$  is any end vertex of  $e$  and  $y$  is any end vertex of  $e'$ .

Let  $e$  be an edge of  $G$  with end vertex  $x$  and  $H := Line(G[N_G^{r+1}(x)])$ . Since  $H \subseteq K$ , we have  $d_K(e, e') \leq d_H(e, e')$  for every  $e' \in V_H$ , hence  $N_H^r(e) \subseteq N_K^r(e)$ . Conversely, if  $d_K(e, e') \leq r$  and  $(e, e_1, e_2, \dots, e_r)$  is a

<sup>7</sup> An *apex-graph* is a graph  $G$  such that  $G \setminus u$  is planar for some vertex  $u$ .

<sup>8</sup> The proof is done in terms of the parameter  $nlcw(G)$  such that  $nlcw(G) \leq cwd(G) \leq 2 \cdot nlcw(G)$ , see [21].

path in  $G$  with  $r' \leq r$  and  $e_{r'} = e'$ , then  $d_G(y, z) \leq r + 1$  for any two distinct vertices  $y$  and  $z$  on this path. Hence,  $e_1, \dots, e_{r'}$  are edges of  $G[N_G^{r+1}(x)]$ ,  $d_H(e, e') \leq r$  and  $e' \in N_H^r(e)$ , hence  $N_H^r(e) = N_K^r(e)$ . It follows that  $H[N_H^r(e)] \subseteq K[N_K^r(e)]$ . Now, if  $e'$  and  $e''$  belong to  $N_K^r(e)$  and share a vertex, then they are edges of  $G[N_G^{r+1}(x)]$ , hence are adjacent vertices of  $H$ . Hence,  $H[N_H^r(e)] = K[N_K^r(e)]$ .

If  $\mathcal{C}$  has bounded local tree-width, then  $twd(G[N_G^{r+1}(x)]) \leq f(r + 1)$  where  $f$  is the function that bounds the local tree-width of the graphs in  $\mathcal{C}$ . It is proved in [21] that if a graph has tree-width  $k$ , then its line graph has clique-width at most  $g(k)$  for some function  $g$ . Hence, the clique-width of  $H$  is bounded by  $g(f(r + 1))$ . Since  $K[N_K^r(e)]$  is an induced subgraph of  $H$ , its clique-width is bounded by  $g(f(r + 1))$  (Fact 2.2 (i)). Therefore,  $Line(\mathcal{C})$  has bounded local clique-width.

- (5) Cliques have clique-width 2. Their line graphs have unbounded clique-width ([21]) and diameter 2, hence unbounded local clique-width.
- (6) An *interval graph* is a graph that has an intersection model consisting of intervals on a straight line. The class of interval graphs has not bounded local clique-width. Otherwise, interval graphs would have bounded clique-width, because if we add to an interval graph a new vertex adjacent to all, we obtain an interval graph of diameter 2.

In order to obtain a log-labeling scheme for certain graph classes of bounded local clique-width, we will cover their graphs, as in [16,17], by graphs of bounded clique-width. In [16] a notion of *nicely locally tree-decomposable* class of structures was introduced. We will define a slightly more general notion.

**Definition 4.2 (Intersection Graph)** *Let  $V$  be a finite set and  $\mathcal{T}$  a subset of  $\mathcal{P}(V)$ . The intersection graph of  $\mathcal{T}$  is the undirected graph  $\mathcal{I}(\mathcal{T})$  where  $V_{\mathcal{I}(\mathcal{T})} := \mathcal{T}$ , and  $U$  and  $V$  in  $\mathcal{T}$  are adjacent in  $\mathcal{I}(\mathcal{T})$  if and only if  $U \neq V$  and  $U \cap V \neq \emptyset$ .*

**Definition 4.3 (Clique-Width Covers)** *Let  $r, \ell$  and  $g$  be positive integers. An  $(r, \ell, g)$ -cwd cover of a graph  $G$  is a family  $\mathcal{T}$  of subsets of  $V_G$  such that:*

- (1) *for every  $x \in V_G$ , there exists a set  $U \in \mathcal{T}$  such that  $N_G^r(x) \subseteq U$ ,*
- (2) *the graph  $\mathcal{I}(\mathcal{T})$  has maximum degree at most  $\ell$ ,*
- (3) *for each  $U \in \mathcal{T}$ , we have  $cwd(G[U]) \leq g$ .*

*Let now  $g : \mathbb{N} \rightarrow \mathbb{N}$  be a mapping. A nice  $(r, \ell, g)$ -cwd cover is a family  $\mathcal{T}$  as above such that Condition (3) is replaced by Condition (3') below:*

(3') for all  $q \geq 1$  and all  $U_1, \dots, U_q \in \mathcal{T}$ ,  $\text{cwd}(G[U_1 \cup \dots \cup U_q]) \leq g(q)$ .

A class  $\mathcal{C}$  of graphs is (nicely) locally cwd-decomposable if every graph  $G$  in  $\mathcal{C}$  has, for each  $r \geq 1$ , a (nice)  $(r, \ell, g)$ -cwd cover for some  $\ell$  and  $g$  depending on  $r$  (but not on  $G$ ).

The notion of (nicely) locally cwd-decomposable is the same as in [17,16] where we substitute clique-width to tree-width, except that our definition requires nothing about the time necessary to compute covers.

### Examples of (Nicely) Locally Cwd-Decomposable Graph Classes

- (1) Every nicely locally cwd-decomposable class is locally cwd-decomposable and the converse does not seem to be true. But, we do not have a counterexample.
- (2) Each class of nicely locally tree-decomposable graphs is nicely locally cwd-decomposable.
- (3) We do not know if every graph class of bounded local clique-width is locally cwd-decomposable. We conjecture that there exists a graph class of bounded local clique-width which is not locally cwd-decomposable.
- (4) Figure 1 shows inclusion relations between the many classes defined in Sections 3 and 4.

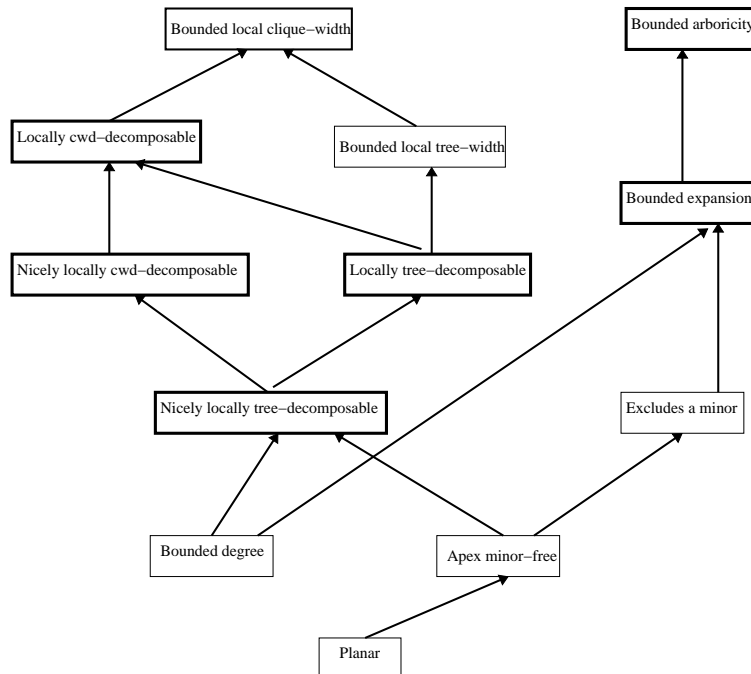


Fig. 1. Inclusion diagram indicating which results apply to which classes. An arrow means an inclusion of classes. Bold boxes are used in this paper.

We give an example of a nicely locally cwd-decomposable class of graphs which is not locally tree-decomposable. A *unit-interval* graph is an interval graph that has an intersection model in which every interval has unit length.

**Fact 4.4** *The class of unit-interval graphs is nicely locally cwd-decomposable.*

**Proof.** We will use a result by Lozin in [27]. We let  $H_{n,m}$  be the graph  $\langle V_1 \cup \dots \cup V_n, E^1 \cup E^2 \rangle$  with  $nm$  vertices such that:

$$\begin{aligned} V_i &:= \{v_{i,1}, \dots, v_{i,m}\}, \\ E^1 &:= \bigcup_{1 \leq i \leq n} \{v_{i,j}v_{i,\ell} \mid j, \ell \leq m, j \neq \ell\}, \\ E^2 &:= \bigcup_{1 \leq i \leq n-1} \{v_{i,j}v_{i+1,\ell} \mid j \leq \ell \leq m\}. \end{aligned}$$

Each subgraph induced by  $V_i$  is a complete graph. Figure 2 shows the graph  $H_{4,4}$ . It is proved in [27] that the clique-width of  $H_{n,m}$  is at most  $3n$ . Moreover, every unit-interval graph with  $n$  vertices is an induced subgraph of  $H_{n,n}$  [27]. These two properties will be used.

We first prove that unit-interval graphs have bounded local clique-width. Let  $G$  be a unit-interval graph with  $n$  vertices. Then for every positive integer  $r$  and every vertex  $x$  of  $G$ , the subgraph  $G[N_G^r(x)]$  is isomorphic to an induced subgraph of  $H_{r,n}$ . Thus, for every vertex  $x$  of  $G$  and every positive integer  $r$ ,  $G[N_G^r(x)]$  has clique-width at most  $3r$ . (Bagan gives in [2] another proof stating that unit-interval graphs have bounded local clique-width.)

We now prove that the class of unit-interval graphs is nicely locally cwd-decomposable. Let  $G$  be a unit-interval graph with  $n$  vertices. Hence, it is a subgraph of  $H_{n,n} := \langle V_1 \cup \dots \cup V_n, E^1, E^2 \rangle$ . Without loss of generality, we may assume  $G$  connected. We can also assume that  $V_G = \bigcup_{1 \leq i \leq n} V'_i$  where  $V'_i = \{v_{i,i_1}, \dots, v_{i,i_\ell}\}$  with  $1 \leq i_1 \leq i_2 \leq \dots \leq i_\ell \leq n$ . For each  $1 \leq i \leq n$ , we let  $U_i := N_G^{r+1}(v_{i,i_1})$ . We let  $g : \mathbb{N} \rightarrow \mathbb{N}$  be defined by  $g(q) := 3 \cdot q \cdot (r+1)$ . We claim that  $\{U_i \mid 1 \leq i \leq n\}$  is a nice  $(r, 2r+3, g)$ -cwd cover. It is clear by construction that for every  $1 \leq i \leq n$  and every vertex  $v$  in  $V'_i$ , the set  $N_G^r(v)$  is a subset of  $U_i$ .

We now prove that for every positive integer  $q$ , if we take  $q$  subsets  $U_{j_1}, \dots, U_{j_q}$ , then the subgraph  $G[U_{j_1} \cup \dots \cup U_{j_q}]$  has clique-width at most  $3 \cdot q \cdot (r+1)$ . Assume that  $j_1 \leq j_2 \leq \dots \leq j_q$  and let  $G_1, \dots, G_p$  be the connected components of  $G[U_{j_1} \cup \dots \cup U_{j_q}]$ . By Fact 2.2 (ii), we need only prove the claim for each connected component. Let  $G_1$  be one of them. It is of the form, without loss of generality,  $G[U_{j_1} \cup \dots \cup U_{j_{\ell_1}}]$  with the property that  $j_1 \leq \dots \leq j_{\ell_1}$ . Thus,  $G_1$  is an induced subgraph of  $H_{j_{\ell_1} \cdot (r+1), n}$ , hence has clique-width at most  $3 \cdot \ell_1 \cdot (r+1)$ . Hence, the clique-width of  $G[U_{j_1} \cup \dots \cup U_{j_q}]$  is at most  $3 \cdot q \cdot (r+1)$ .

Let  $v$  be a vertex in  $V'_i$  for  $1 \leq i \leq n$ . By construction,  $v$  can only be in  $U_i, U_{i-1}, \dots, U_{i-(r+1)}, U_{i+1}, \dots, U_{i+(r+1)}$ . Thus,  $v$  is in at most  $2(r+1) + 1$  sets  $U_i$ . This concludes the proof.  $\square$

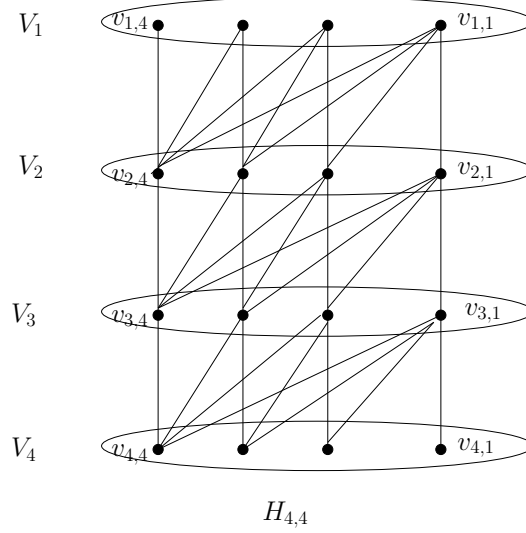


Fig. 2. The graph  $H_{4,4}$ . Each  $V_i$  for  $1 \leq i \leq 4$ , induces a clique.

The lemma below is an easy adaptation of the results in [17].

**Lemma 4.5** *Let  $\mathcal{C}$  be a class of graphs of bounded local clique-width and let  $\varphi$  be a basic  $(t, s)$ -local sentence without set variables. For every graph  $G$  in  $\mathcal{C}$ , we can check in time  $O(|V_G|^4)$  whether  $G$  satisfies  $\varphi$ .*

**Proof Sketch.** Let  $G$  be in a class  $\mathcal{C}$  of graphs of bounded local clique-width and let  $f$  be the function that bounds the local clique-width of the graphs in  $\mathcal{C}$ . Let  $\varphi$  be a basic  $(t, s)$ -local sentence, equivalent to

$$\exists x_1 \cdots \exists x_s \left( \bigwedge_{1 \leq i < j \leq s} d(x_i, x_j) > 2t \wedge \bigwedge_{1 \leq i \leq s} \psi(x_i) \right)$$

where  $\psi(x)$  is  $t$ -local around its unique free variable  $x$ .

For each vertex  $a$  in  $G$ , we can compute the set  $N_G^t(a)$  of size at most  $|V_G|$ , in total time  $O(|V_G|^2)$ . Since  $cwd(G[N_G^t(a)]) \leq f(t)$ , we can verify in time  $O(|V_G|^3)$  if  $G$  satisfies  $\psi(a)$  by combining the results of [22] and of [7]. We can thus compute in time  $O(|V_G|^4)$  the set  $P := \{a \in V_G \mid G \models \psi(a)\}$ . The formula  $\varphi$  is valid in  $G$  if and only if there exist  $a_1, \dots, a_s$  in  $P$  such that  $d(a_i, a_j) > 2t$  for any  $1 \leq i < j \leq s$ . It is proved in [19] that we can verify their existence in time  $O(|V_G|^3)$ .  $\square$

## 5 Labeling Schemes for First-Order Queries

Our results concern four types of graph classes (see Figure 1) and five types of FO queries. We now state the main theorem of the paper. We denote by  $n$  the number of vertices of each input graph, and each query is denoted by  $\varphi(\bar{x}, \bar{Y})$ . We denote by  $\bar{a}$  an  $|\bar{x}|$ -tuple of vertices and by  $\bar{W}$  a  $|\bar{Y}|$ -tuple of sets of vertices of the considered graph.

**Theorem 5.1 (First Main Theorem)** *There exist log-labeling schemes  $(\mathcal{A}, \mathcal{B})$  for the following queries and graph classes.*

- (1) *Quantifier-free parametrized queries in graphs of bounded arboricity: the algorithm  $\mathcal{A}$  constructs a labeling in time  $O(n)$ , and the algorithm  $\mathcal{B}$  gives the answer in time  $O(\log(n) \cdot (|\bar{W}| + |\bar{a}| + 1))$  for all tuples  $\bar{a}$  and  $\bar{W}$ ; the same labeling can be used to check any quantifier-free query<sup>9</sup>.*
- (2) *Bounded parametrized FO queries for each class of graphs of bounded expansion: the algorithm  $\mathcal{A}$  constructs a labeling in time  $O(n \cdot \log(n))$ , and the algorithm  $\mathcal{B}$  gives the answer in time  $O(\log(n) \cdot (|\bar{W}| + 1))$  for all tuples  $\bar{a}$  and  $\bar{W}$ .*
- (3) *FO queries in locally cwd-decomposable classes: the algorithm  $\mathcal{A}$  constructs a labeling in time  $O(f(n) + n^4)$  where  $f(n)$  is the time taken to construct a cwd-cover for given  $r$ , and the algorithm  $\mathcal{B}$  gives the answer in time  $O(\log(n))$  for all tuples  $\bar{a}$ .*
- (4) *FO parametrized queries in nicely locally cwd-decomposable classes: the algorithm  $\mathcal{A}$  constructs a labeling in time  $O(f(n) + n^4)$  where  $f(n)$  is the time taken to construct a nice cwd-cover for given  $r$ , and the algorithm  $\mathcal{B}$  gives the answer in time  $O(\log(n) \cdot (|\bar{W}| + 1))$  for all tuples  $\bar{a}$  and  $\bar{W}$ .*

Each statement is proved separately.

**Proof of Theorem 5.1 (1).** Let  $G$  be a colored graph with  $n$  vertices, represented by the relational structure  $\langle V_G, (edg_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1} \rangle$  and of arboricity at most  $k$ .

Assume first that  $und(G)$ , the simple loop-free graph obtained from  $G$  by forgetting loops, edge directions, colors of vertices and of edges, and fusing the parallel undirected edges, is a forest. Let  $R$  be a subset of  $V_G$  that contains one and only one vertex of each connected component, which is a tree, of  $G$ . For each color  $c$  in  $C_2$ , we let  $f_c^+, f_c^- : V_G \rightarrow V_G$  be partial functions such that:

- $f_c^+(u) := v$  if  $edg_{cG}(u, v)$  holds and  $v$  is on the unique undirected path between  $u$  and some vertex of  $R$ ,

<sup>9</sup> The time bound is valid even if  $\bar{W} = (\emptyset, \dots, \emptyset)$  because  $|\bar{W}| + 1 \neq 0$ .



- $f_c^-(u) := v$  if  $edg_{cG}(v, u)$  holds and  $v$  is on the unique undirected path between  $u$  and some vertex of  $R$ .

For every two distinct vertices  $u$  and  $v$  of  $G$ ,

$$edg_{cG}(u, v) \iff v = f_c^+(u) \vee u = f_c^-(v). \quad (1)$$

If  $und(G)$  is the union of  $k$  edge-disjoint forests  $F_1, \dots, F_k$ , we take the pairs  $(f_{i,c}^+, f_{i,c}^-)$ , for each forest  $und(F_i)$ . For every two distinct vertices  $u$  and  $v$  of  $G$ ,  $edg_{cG}(u, v)$  is defined in a similar way as in (1) with  $2k$  unary functions by letting

$$edg_{cG}(u, v) \iff \bigvee_{i \in [k]} (v = f_{i,c}^+(u) \vee u = f_{i,c}^-(v)). \quad (2)$$

We let  $C_1 := \{d_1, \dots, d_p\}$  and  $C_2 := \{c_1, \dots, c_\ell\}$ . For each vertex  $x$  of  $G$ , we let  $\vec{b}_x$  be the Boolean vector  $(b_1, \dots, b_p, b'_1, \dots, b'_\ell)$  such that  $b_i := 1$  if and only if  $p_{d_i G}(x)$  holds, and  $b'_j := 1$  if and only if  $edg_{c_j G}(x, x)$  holds. If vertices are numbered from 1 to  $n$  and  $\ulcorner x \urcorner$  is the bit representation of the index of  $x$ , then we let

$$J(x) := (\ulcorner x \urcorner, \ulcorner f_{1,c_1}^+(x) \urcorner, \ulcorner f_{1,c_1}^-(x) \urcorner, \dots, \ulcorner f_{k,c_\ell}^+(x) \urcorner, \ulcorner f_{k,c_\ell}^-(x) \urcorner, \vec{b}_x).$$

(Since the vertices are numbered from 1 to  $n$ , the value 0 can be used for  $f_c^+(u)$  if it is undefined.) It is clear that  $|J(x)| = O(\log(n))$ .

We now explain how to check any quantifier-free formula by using this labeling. Let  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  be a quantifier-free formula. For all  $m$ -tuples  $\vec{a} := (a_1, \dots, a_m)$  of  $V_G$  and all  $q$ -tuples  $\vec{W} := (W_1, \dots, W_q)$  of subsets of  $V_G$ , we can determine the induced subgraph  $G[\{a_1, \dots, a_m\} \cup W_1 \cup \dots \cup W_q]$  from  $J(a_1), \dots, J(a_m)$  and  $J(W_1), \dots, J(W_q)$ , and check whether  $\varphi(\vec{a}, \vec{W})$  holds, because  $G \models \varphi(\vec{a}, \vec{W})$  if and only if  $G[\{a_1, \dots, a_m\} \cup W_1 \cup \dots \cup W_q] \models \varphi(\vec{a}, \vec{W})$ .

If the input graph has  $n$  vertices and  $n'$  edges, then it is clear that our algorithm constructs the labels in time  $O(n + n')$ <sup>10</sup>. But, if  $G$  has arboricity at most  $k$ , the number of edges is at most  $2k \cdot |C_2| \cdot n$ . Therefore, the labels are constructed in linear-time. We now examine the time taken to check whether  $G$  satisfies  $\varphi(a_1, \dots, a_m, W_1, \dots, W_q)$ . For each  $z \in \{a_1, \dots, a_m\}$ , it takes constant time to check whether  $p_{d_i G}(z)$  or  $edg_{c_j G}(z, z)$  holds by using the  $\vec{b}_z$  part of  $J(z)$ . For every  $z$  and  $t$  in  $W_1 \cup \dots \cup W_q \cup \{a_1, \dots, a_m\}$  and every  $c$  in  $C_2$ , it takes time  $O(\log(n))$  to check whether  $edg_{cG}(z, t)$  holds and it takes time  $O(|W_i| \cdot \log(n))$  to check if  $z$  is in  $W_i$ . Therefore, we can check the validity of

<sup>10</sup> If no covering of  $G$  by  $k$  forests is known, one can construct a covering by  $2k - 1$  forests in time  $O(n + n')$  by Eppstein [14].

$\varphi(a_1, \dots, a_m, W_1, \dots, W_q)$  in time  $O(\log(n) \cdot (|\overline{W}| + m + 1))$  since a quantifier-free formula is a Boolean combination of atomic formulas. Note that  $m$  and  $q$  are not fixed because the same labeling can be used for all quantifier-free formulas.  $\square$

**Proof of Theorem 5.1 (2).** Let  $\mathcal{C}$  be a class of graphs of bounded expansion and let  $G$  in  $\mathcal{C}$  be a graph with  $n$  vertices, represented by the relational structure  $\langle V_G, (edg_{aG})_{a \in C_2}, (paG)_{a \in C_1} \rangle$ . Let  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  with  $m \geq 1$  be a basic bounded formula with bound  $p$  on the quantification space (see Definition 3.1). We partition  $V_G$  into  $V_1 \uplus V_2 \uplus \dots \uplus V_N$  as in Definition 2.3 with each  $V_i$  nonempty and  $N \leq N(\mathcal{C}, p)$ . (We denote by  $\uplus$  the disjoint union of sets.)

For every  $\alpha \subseteq [N]$  of cardinality  $p$ , we let  $V_\alpha := \bigcup_{i \in \alpha} V_i$  so that the tree-width of  $G[V_\alpha]$  is at most  $p - 1$ . Each vertex  $x$  belongs to less than  $(N - 1)^{p-1}$  sets  $V_\alpha$ . Hence, the basic bounded formula  $\varphi(\bar{a}, \overline{W})$  is true in  $G$  if and only if it is true in some subgraph  $G[X]$  with  $|X| \leq p$ , hence in some  $G[V_\alpha]$  such that  $a_1, \dots, a_m \in V_\alpha$  where  $\bar{a} := (a_1, \dots, a_m)$ . For each  $\alpha$ , we construct a labeling  $J_\alpha$  of  $G[V_\alpha]$  (this graph has tree-width at most  $p - 1$ ) supporting query  $\varphi$  by using Theorem 2.5. We let  $J(x) := (\ulcorner x \urcorner, ((\ulcorner \alpha \urcorner, J_\alpha(x)) \mid x \in V_\alpha))$ . We have  $|J(x)| = O(\log(n))$ .

Given  $\bar{a} \in (V_G)^m$  and  $\overline{W} \in \mathcal{P}(V_G)^q$ , we now explain how to decide the validity of  $\varphi(\bar{a}, \overline{W})$  by using  $J(a_1), \dots, J(a_m)$  and  $J(W_1), \dots, J(W_q)$ . From  $J(a_1), \dots, J(a_m)$ , we can determine those sets  $\alpha$  such that  $V_\alpha$  contains  $a_1, \dots, a_m$ . Using the components  $J_\alpha(\cdot)$  of  $J(a_1), \dots, J(a_m)$  and the labels in  $J(W_1), \dots, J(W_q)$ , we can determine whether for some  $\alpha$ , we have  $G[V_\alpha] \models \varphi(a_1, \dots, a_m, W_1 \cap V_\alpha, \dots, W_q \cap V_\alpha)$ , hence whether  $G \models \varphi(a_1, \dots, a_m, W_1, \dots, W_q)$ .

It remains to consider the case of a basic bounded formula of the form  $\varphi(Y_1, \dots, Y_q)$ , i.e., the case where  $m = 0$ . We define the labelings  $J_\alpha$  from  $\varphi$  as in the first case. For each  $\alpha$ , we determine the truth value  $b_\alpha$  of  $\varphi(\emptyset, \dots, \emptyset)$  in  $G[V_\alpha]$ . The family of pairs  $(\alpha, b_\alpha)$  is of fixed size (depending on  $p$ ) and is appended (as a sequence of bits) to  $J(x)$  defined as above.

From  $J(W_1), \dots, J(W_q)$ , we can determine  $D := \{\alpha \mid V_\alpha \cap (W_1 \cup \dots \cup W_q) \neq \emptyset\}$ . By using the  $J_\alpha(\cdot)$  components of the labels in  $J(W_1) \cup \dots \cup J(W_q)$ , we can determine if for some  $\alpha \in D$ , we have  $G[V_\alpha] \models \varphi(W_1 \cap V_\alpha, \dots, W_q \cap V_\alpha)$ . If one is found, we can conclude positively. Otherwise, we look for some  $\beta \notin D$  such that  $b_\beta = \text{TRUE}$ . The final answer is positive if such  $\beta$  is found.

For a Boolean combination of basic bounded formulas  $\varphi_1, \dots, \varphi_t$  with associated labelings  $J_1, \dots, J_t$ , we take as label of  $x$  the concatenation of the labels  $J_1(x), J_2(x), \dots, J_t(x)$ . This label is of size  $O(\log(n))$  and gives the desired

result.

In [28] Nešetřil and Ossona de Mendez describe a linear-time algorithm that computes the partition  $\{V_1, \dots, V_N\}$ . The number of sets  $V_\alpha$  where  $\alpha$  is a subset of  $[N]$  of size  $p$  is bounded by  $N^p$ , hence so is the number of graphs  $G[V_\alpha]$ . Then the labeling  $J$  is constructed in time  $O(n \cdot \log(n))$  since each labeling  $J_\alpha$  is constructed in time  $O(n \cdot \log(n))$  by Theorem 2.5.

We now examine the time taken to check whether  $G$  satisfies  $\varphi(a_1, \dots, a_m, W_1, \dots, W_q)$ . Each vertex  $x$  is in less than  $(N-1)^{p-1}$  sets  $V_\alpha$ . By comparing the sets that contain all the  $a_i$ 's with the sets that contain  $a_1$ , we can determine in time  $O(\log(n))$  the sets  $V_\alpha$  that contain  $a_1, \dots, a_m$ . For each  $V_\alpha$  and each  $W_i$ , we can determine in time  $O(\log(n) \cdot |W_i|)$  the set  $W_i \cap V_\alpha$ . By Theorem 2.5, we can verify in time  $O(\log(n) \cdot (|\overline{W}| + 1))$ , for each  $\alpha$ , whether  $G[V_\alpha]$  satisfies  $\varphi(a_1, \dots, a_m, W_1 \cap V_\alpha, \dots, W_q \cap V_\alpha)$ . Therefore,  $\mathcal{B}$  checks the validity of  $\varphi(a_1, \dots, a_m, W_1, \dots, W_q)$  in time  $O(\log(n) \cdot (|\overline{W}| + 1))$ .  $\square$

Before continuing, we prove a lemma necessary for the proofs of Theorem 5.1 (3)-(4), that concerns parametrized local queries.

**Lemma 5.2** *For every parametrized local query, there exists a log-labeling scheme  $(\mathcal{A}, \mathcal{B})$  on locally cwd-decomposable classes of graphs. Moreover, the algorithm  $\mathcal{A}$  constructs a labeling in time  $O(f(n) + n^4)$  where  $f(n)$  is the time taken to construct a cwd-cover for given  $r$ , and the algorithm  $\mathcal{B}$  gives the answer in time  $O(\log(n) \cdot (|\overline{W}| + 1))$  for all tuples  $\bar{a}$  and  $\overline{W}$ .*

**Proof of Lemma 5.2.** Let  $\mathcal{C}$  be a locally cwd-decomposable class of graphs and let  $G = \langle V_G, (edg_{aG})_{a \in \mathcal{C}_2}, (paG)_{a \in \mathcal{C}_1} \rangle$  in  $\mathcal{C}$  be a graph with  $n$  vertices. Let  $\varphi(\bar{x}, Y_1, \dots, Y_q)$  be a  $t$ -local formula around  $\bar{x} := (x_1, \dots, x_m)$ ,  $m \geq 1$ . Then  $G \models \varphi(\bar{a}, W_1, \dots, W_q)$  if and only if  $G[N_G^t(\bar{a})] \models \varphi(\bar{a}, W_1 \cap N_G^t(\bar{a}), \dots, W_q \cap N_G^t(\bar{a}))$ . Let  $\Delta$  be a  $t$ -distance type with connected components  $\Delta_1, \dots, \Delta_p$ . By Theorem 3.5,  $G \models \rho_{t,\Delta}(\bar{a}) \wedge \varphi(\bar{a}, W_1, \dots, W_q)$  if and only if  $G \models \rho_{t,\Delta}(\bar{a}) \wedge F^{t,\Delta}(\varphi_{1,1}(\bar{a} \upharpoonright \Delta_1, W_1, \dots, W_q), \dots, \varphi_{p,p}(\bar{a} \upharpoonright \Delta_p, W_1, \dots, W_q))$ .

We let  $\mathcal{T}$  be an  $(r, \ell, g)$ -cwd cover of  $G$  where  $r := m(2t+1)$ . We use this value of  $r$  to guarantee that if  $\Delta = \Delta(a_1, \dots, a_m)$  and  $i_1, \dots, i_k$  in  $[m]$  belong to a connected component of  $\Delta$ , then  $N_G^t(\{a_{i_1}, \dots, a_{i_k}\}) \subseteq U$  for some  $U$  in  $\mathcal{T}$ . This is so because  $d_G(a_{i_1}, a_{i_{k'}}) \leq (m-1) \cdot (2t+1)$  for every  $k' = 2, \dots, k$ , hence if  $a \in N_G^t(\{a_{i_1}, \dots, a_{i_k}\})$ , we have  $d_G(a_{i_1}, a) \leq t + (m-1) \cdot (2t+1) < r$ . Hence,  $N_G^t(\{a_{i_1}, \dots, a_{i_k}\}) \subseteq N_G^r(a_{i_1}) \subseteq U$  for some  $U$  in  $\mathcal{T}$ . For each vertex  $x$ , there exist less than  $\ell$  many sets  $V$  in  $\mathcal{T}$  such that  $x \in V$ . We assume that each set  $U$  in  $\mathcal{T}$  has an index encoded as a bit string denoted by  $\lceil U \rceil$ . There are at most  $n \cdot \ell$  sets in  $\mathcal{T}$ . Hence,  $\lceil U \rceil$  has length  $O(\log(n))$ .

For each set  $U$  in  $\mathcal{T}$ , we label each vertex in  $G[U]$  by  $L_U(x)$  of length  $O(\log(n))$  in order to decide from  $L_U(x)$  and  $L_U(y)$  whether  $d_{G[U]}(x, y) \leq 2t+1$  (Theorem 2.5)<sup>11</sup>. For each vertex  $x$  of  $G$ , we let

$$L(x) := \left( \ulcorner x \urcorner, \left\{ \left( \ulcorner U \urcorner, L_U(x) \right) \mid N_G^{2t+1}(x) \subseteq U \right\}, \left\{ \left( \ulcorner U \urcorner, L_U(x) \right) \mid x \in U, N_G^{2t+1}(x) \not\subseteq U \right\} \right).$$

(Recall that  $x \in N_G^{2t+1}(x)$ .) It is clear that  $|L(x)| = O(\log(n))$ .

By Theorem 2.5, for each formula  $\varphi_{i,j}(\bar{x} \upharpoonright \Delta_i, Y_1, \dots, Y_q)$ , arising from Theorem 3.5, and each  $U \in \mathcal{T}$ , we can label each vertex  $x \in U$  by some label  $J_{i,j,U}^\Delta(x)$  of length  $O(\log(n))$  so that we can decide if  $\varphi_{i,j}(\bar{a} \upharpoonright \Delta_i, W_1, \dots, W_q)$  holds in  $G[U]$  by using  $\left( J_{i,j,U}^\Delta(b) \right)_{b \in \bar{a} \upharpoonright \Delta_i}$  and  $J_{i,j,U}^\Delta(W_1 \cap U), \dots, J_{i,j,U}^\Delta(W_q \cap U)$ . For each vertex  $x$  of  $G$ , we let  $J_\Delta(x)$  be the concatenation of the tuples  $\left( \ulcorner U \urcorner, J_{1,1,U}^\Delta(x), \dots, J_{1,j_1,U}^\Delta(x), \dots, J_{p,1,U}^\Delta(x), \dots, J_{p,j_p,U}^\Delta(x) \right)$  for all  $U \in \mathcal{T}$  such that  $N_G^t(x) \subseteq U$ .

It is clear that  $|J_\Delta(x)| = O(\log(n))$  since each vertex  $x$  is in less than  $\ell$  many sets  $U$  in  $\mathcal{T}$ . There exist at most  $2^{m(m-1)/2}$   $t$ -distance types; we enumerate them as  $\Delta^1, \dots, \Delta^{k'}$  for some  $k' \leq 2^{m(m-1)/2}$ . For each vertex  $x$  of  $G$ , we let  $J(x) := (L(x), J_{\Delta^1}(x), \dots, J_{\Delta^{k'}}(x))$ . It is clear that  $J(x)$  is of length  $O(\log(n))$ .

By hypothesis, the cover  $\mathcal{T}$  is computed in time  $f(n)$  for  $G$  in  $\mathcal{C}$  with  $n$  vertices. By Theorem 2.5, the labelings  $L_U$  and  $J_{i,j,U}^\Delta$  can be constructed in cubic-time. Therefore, the labeling  $J$  can be constructed in total time  $O(f(n) + n^4)$  since there are less than  $\ell \cdot n$  sets  $U$  in  $\mathcal{T}$ .

We now explain how to decide whether  $G \models \varphi(a_1, \dots, a_m, W_1, \dots, W_q)$  by using  $J(a_1), \dots, J(a_m)$  and  $J(W_1), \dots, J(W_q)$ .

From the labels  $L(x)$ , we can determine the set  $\{\ulcorner U \urcorner \mid U \in \mathcal{T}, x \in U\}$ , hence the family of sets  $U \in \mathcal{T}$  such that  $W \cap U \neq \emptyset$ , where  $W \subseteq V_G$  is a set argument.

Since for each vertex  $x$  of  $G$ , there exists a set  $U$  in  $\mathcal{T}$  such that  $N_G^r(x) \subseteq U$ , for each pair of vertices  $\{x, y\}$ , we have  $d_G(x, y) \leq 2t+1$  if and only if  $d_{G[U]}(x, y) \leq 2t+1$ . Hence, by using the components  $L(a_1), \dots, L(a_m)$  of  $J(a_1), \dots, J(a_m)$ , we can construct the  $t$ -distance type  $\Delta$  of  $(a_1, \dots, a_m)$ ; let  $\Delta_1, \dots, \Delta_p$  be the connected components of  $\Delta$ . From each  $J(a_i)$ , we can recover  $J_\Delta(a_i)$ . For each tuple  $\bar{a} \upharpoonright \Delta_i$ , there exists at least one  $U \in \mathcal{T}$  such that  $N_G^t(\bar{a} \upharpoonright \Delta_i) \subseteq U$ . We let  $U_i$  be any of them. It can be determined by using the component  $L(b)$  of the label  $J(b)$  for each  $b \in \bar{a} \upharpoonright \Delta_i$ . One can determine (by using  $J(W_1), \dots, J(W_q)$ ) the sets  $W_i \cap U_j$ . Hence, one can check if  $G[U_i] \models \varphi_{i,j}(\bar{a} \upharpoonright \Delta_i, W_1 \cap U_i, \dots, W_q \cap U_p)$  for each  $j \in [j_i]$ . We can now decide whether  $G \models$

<sup>11</sup> For checking if  $d_G(x, y) \leq 2t+1$ , an  $(r', \ell', g')$ -cwd cover suffices with  $r' := 2t+1$ .

$F^{t,\Delta}(\varphi_{1,1}(\bar{a} \upharpoonright \Delta_1, W_1 \cap U_1, \dots, W_q \cap U_1), \dots, \varphi_{p,j_p}(\bar{a} \upharpoonright \Delta_p, W_1 \cap U_p, \dots, W_q \cap U_p))$  for the chosen  $(U_1, \dots, U_p)$ . This is sufficient by Theorem 3.5.

We now examine the time taken to check  $\varphi(\bar{a}, \bar{W})$ . For each couple  $(a_i, a_j)$ , it takes time  $O(\log(n))$  to check if  $d(a_i, a_j) \leq 2t + 1$ . Since there are at most  $m^2$  couples, we construct the graph  $\Delta$  in time  $O(\log(n))$ . For each connected component  $\bar{a} \upharpoonright \Delta$ , we can determine the sets  $U$  that contain it in time  $O(\log(n))$  (because there are less than  $\ell$  such sets). By Theorem 2.5, we can check each  $\varphi_{i,j}$  in time  $O(\log(n) \cdot (|\bar{W}| + 1))$ . Therefore,  $\mathcal{B}$  checks the validity of  $\varphi(\bar{a}, \bar{W})$  in time  $O(\log(n) \cdot (|\bar{W}| + 1))$ .  $\square$

We can now prove Theorem 5.1(3)-(4).

**Proof of Theorem 5.1 (3).** Let  $\mathcal{C}$  be a locally cwd-decomposable class of graphs and let  $G = \langle V_G, (edg_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1} \rangle$  in  $\mathcal{C}$  be a graph with  $n$  vertices. Let  $\varphi(x_1, \dots, x_m)$  be an FO formula without set arguments. By Theorem 3.4, this formula is equivalent to a Boolean combination  $B(\varphi_1(\bar{x}), \dots, \varphi_p(\bar{x}), \psi_1, \dots, \psi_h)$  where each  $\varphi_i$  is  $t$ -local and each  $\psi_i$  is a basic  $(t', s)$ -local sentence without set variables, for some  $t, t'$  and  $s$  depending only on  $\varphi$ .

By Lemma 4.5, one can decide in time  $O(n^4)$  the validity of each sentence  $\psi_i$ . Let  $\vec{b} := (b_1, \dots, b_h)$  where  $b_i := 1$  if  $G$  satisfies  $\psi_i$  and 0 otherwise. For each  $1 \leq i \leq p$ , we construct a labeling  $J_i$  supporting the query  $\varphi_i$  by Lemma 5.2 ( $G$  belongs to a locally cwd-decomposable class and  $\varphi_i$  is a  $t$ -local formula around  $\bar{x}$ ). For each vertex  $x$  of  $G$ , we let  $J(x) := (\ulcorner x \urcorner, J_1(x), \dots, J_p(x), \vec{b})$ <sup>12</sup>. It is clear that  $|J(x)| = O(\log(n))$  since  $|J_i(x)| = O(\log(n))$ . We now explain how to decide whether  $G \models \varphi(a_1, \dots, a_m)$  by using  $J(a_1), \dots, J(a_m)$ .

From the Boolean vector  $\vec{b}$ , we can recover the truth value of each sentence  $\psi_i$ . By using  $J_i(\bar{a})$ , we can check if  $\varphi_i(\bar{a})$  holds. Then we can check if  $B(\varphi_1(\bar{x}), \dots, \varphi_p(\bar{x}), \psi_1, \dots, \psi_h)$  holds, hence if  $\varphi(\bar{a})$  holds.

By Lemma 4.5, the validity of each sentence  $\psi_i$  is checked in time  $O(n^4)$ . And by Lemma 5.2, each labeling  $J_i$  is constructed in time  $O(f(n) + n^4)$  where  $f(n)$  is the time taken for constructing an  $(r, \ell, g)$ -cwd cover. Hence, the labeling  $J$  can be constructed in time  $O(f(n) + n^4)$ . The time taken to check the validity of  $\varphi(a_1, \dots, a_m)$  is  $O(\log(n))$  by Lemma 5.2.  $\square$

<sup>12</sup>We must put  $\vec{b}$  in each label because the algorithm  $\mathcal{B}$  (cf. Definition 2.4) does not know anything about the considered graph  $G$ . All informations about it must be in the vertex labels.

Before proving Theorem 5.1 (4), we introduce some definitions and facts. Let  $m$  be a positive integer. A *distance- $m$  coloring* of a graph  $H$  is a proper coloring of  $\text{und}(H)^m$  (see Section 2 for the definition of  $\text{und}(H)^m$ ). Then in a distance- $m$  coloring, vertices at distance at most  $m$  have different colors. A graph admits a proper  $(d+1)$ -coloring if  $d$  is its maximum degree. If  $\mathcal{T}$  is an  $(r, \ell, g)$ -cwd cover of a graph  $G$ , then  $\mathcal{I}(\mathcal{T})$  has maximum degree at most  $\ell$ . Hence,  $\mathcal{I}(\mathcal{T})$  has a distance- $m$  coloring with  $\ell^{O(m)}$  colors since  $\mathcal{I}(\mathcal{T})^m$  has maximum degree at most  $\ell \cdot (1 + (\ell - 1) + \dots + (\ell - 1)^{m-1})$ . If  $U$  is a subset of  $V_G$  and  $t \geq 1$ , we let  $K_G^t(U)$  be the set  $\{x \in U \mid N_G^t(x) \subseteq U\}$ . We call it the  *$t$ -kernel* of  $U$ .

We say that two sets of vertices  $W$  and  $W'$  of a graph  $G$  *touch* if  $W \cap W' \neq \emptyset$  or there exists an edge between a vertex of  $W$  and one of  $W'$ . It is clear that if  $W := \bigcup_{1 \leq i \leq p} W_i \subseteq V_G$  and  $W_i, W_j$  pairwise do not touch, then  $G[W]$  is the disjoint union of the graphs  $G[W_i]$ . It follows that  $\text{cwd}(G[W]) = \max\{\text{cwd}(G[W_i]) \mid 1 \leq i \leq p\}$ , by Fact 2.2 (ii).

**Proof of Theorem 5.1 (4).** Let  $\mathcal{C}$  be a nicely locally cwd-decomposable class of graphs and let  $G = \langle V_G, (\text{edg}_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1} \rangle$  in  $\mathcal{C}$  be a graph with  $n$  vertices. We want a labeling for a parametrized FO query. By Theorem 3.4 and Lemma 5.2, it is sufficient to define a labeling for FO formulas  $\varphi(Y_1, \dots, Y_q)$  of the form (they are the sentences  $\psi_1, \dots, \psi_h$  of Theorem 3.4):

$$\exists x_1 \cdots \exists x_m \left( \bigwedge_{1 \leq i < j \leq m} d(x_i, x_j) > 2t \wedge \bigwedge_{1 \leq i \leq m} \psi(x_i, Y_1, \dots, Y_q) \right)$$

where  $\psi(x, Y_1, \dots, Y_q)$  is  $t$ -local around  $x$ . We show how to check such formulas by means of log-labelings.

We first consider for the sake of clarity the particular case where  $m = 2$ . Let  $\mathcal{T}$  be a nice  $(r, \ell, g)$ -cwd cover of  $G$  where  $r := 2t + 1$  and let  $\gamma$  be a distance-3 coloring of  $\mathcal{I}(\mathcal{T})$ , the intersection graph of  $\mathcal{T}$ . For every two colors  $i$  and  $j$ , we let  $G_{i,j}$  be the graph induced by the union of the sets  $U$  in  $\mathcal{T}$  that are colored by  $i$  or  $j$  (we may have  $i = j$ ).

**Claim 5.3**  $\text{cwd}(G_{i,j}) \leq g(2)$ .

**Proof of Claim 5.3.** Let  $\mathcal{T}^2 := \{U \cup U' \mid U, U' \in \mathcal{T}, U \cap U' \neq \emptyset\}$ . The vertex set of the graph  $G_{i,j}$  is a union of sets in  $\mathcal{T} \cup \mathcal{T}^2$ . No two sets of this union touch: if a set  $U \cup U'$  is such that  $U \cap U' \neq \emptyset$  and meets some  $U'' \in \mathcal{T}$  with  $U'' \neq U$  and  $U'' \neq U'$ , then we have  $\gamma(U) = i$ ,  $\gamma(U') = j \neq i$  and  $U''$  meets  $U$  or  $U'$ . It can have neither color  $i$  nor color  $j$  because  $\gamma$  is a distance-3 coloring and  $U, U'$  and  $U''$  are pairwise at distance at most 2. Now, if there exists an

edge between a vertex  $x$  in  $U \cup U'$  and a vertex  $y$  in  $U'' \in \mathcal{T}$ , then there exists a set  $W \in \mathcal{T}$  such that  $x$  and  $y$  are in  $W$ . Hence,  $U''$  and  $U$  are at distance at most 3, similarly for  $U''$  and  $U'$ . Thus,  $U''$  can have neither color  $i$  nor color  $j$ . We can then conclude that  $G_{i,j}$  is a disjoint union of graphs  $G[U \cup U']$  with  $U \cup U' \in \mathcal{T}^2$  and of graphs  $G[U]$  for  $U \in \mathcal{T}$  that do not touch pairwise. Since  $\text{cwd}(G[U \cup U']) \leq g(2)$ , we are done (by Fact 2.2 (ii)).  $\square$

**Claim 5.4** *Let  $x \in K_G^{2t}(U)$  and  $y \in K_G^{2t}(U')$  for some sets  $U$  and  $U'$  in  $\mathcal{T}$ . Then  $d_G(x, y) > 2t$  if and only if  $d_{G[U \cup U']}(x, y) > 2t$ .*

**Proof of Claim 5.4.** It is clear that if  $d_G(x, y) > 2t$ , then  $d_{G[U \cup U']}(x, y) > 2t$  since  $d_G(x, y) \leq d_{G[U \cup U']}(x, y)$ . For proving the converse direction, assume that  $d_G(x, y) \leq 2t$ . Then there exists in  $G$  a path of length at most  $2t$  from  $x$  to  $y$ . This path is also in  $G[U]$  since  $x \in K_G^{2t}(U)$ . Hence, it is also in  $G[U \cup U']$ . Therefore,  $d_{G[U \cup U']}(x, y) \leq 2t$ .  $\square$

Let us now give to each vertex  $x$  of  $G$  the smallest color  $i$  such that  $x \in K_G^{2t}(U)$  and  $\gamma(U) = i$ . Hence, each vertex has one and only one color. We express this by the validity of  $p_i(x)$  where  $p_i$  is a new unary relation. The number of relations  $p_i$  does not depend on the graph  $G$ . For each pair  $(i, j)$  (possibly  $i = j$ ), we consider the formula

$$\psi_{i,j} := \exists x, y \left( d(x, y) > 2t \wedge \psi(x, Y_1, \dots, Y_q) \wedge \psi(y, Y_1, \dots, Y_q) \wedge p_i(x) \wedge p_j(y) \right).$$

By Theorem 2.5, we can construct a *log*-labeling  $J_{i,j}$  for the formula  $\psi_{i,j}$  in the graph  $G_{i,j}$  (because vertex colors, i.e., additional unary relations, do not modify the clique-width). We compute the truth value  $b_{i,j}$  of  $\psi_{i,j}(\emptyset, \dots, \emptyset)$  in  $G_{i,j}$ ; we get a Boolean vector  $\vec{b}$  of fixed length. We also label each vertex  $x$  by its color  $\gamma(x)$ . We concatenate that  $\vec{b}$  and the labels  $J_{i,j}(x)$ , for each  $x$  in  $V_{G_{i,j}}$ , giving  $J(x)$ . The coloring  $\gamma$  uses  $O(\ell^3)$  colors. Then the number of graphs  $G_{i,j}$  is bounded by  $O(\ell^6)$ . Therefore,  $|J(x)| = O(\log(n))$ .

From  $J(W_1), \dots, J(W_q)$ , we can determine those graphs  $G_{i,j}$  such that  $V_{G_{i,j}} \cap (W_1 \cup \dots \cup W_q) \neq \emptyset$  and check if for one of them,  $G_{i,j} \models \psi_{i,j}(W_1, \dots, W_q)$ . If one is found, we are done. Otherwise, we use the Booleans  $b_{i,j}$  to look for graphs  $G_{i,j}$  such that  $G_{i,j} \models \psi_{i,j}(\emptyset, \dots, \emptyset)$  and  $(W_1 \cup \dots \cup W_q) \cap V_{G_{i,j}} = \emptyset$ . This gives the correct answers because of the following facts:

- If  $x$  and  $y$  satisfy the formula  $\varphi$ , then  $x \in K_G^{2t}(U)$ ,  $y \in K_G^{2t}(U')$  (possibly  $U = U'$ ) and  $d_G(x, y) > 2t$  implies  $d_{G_{i,j}}(x, y) > 2t$ , hence  $G_{i,j} \models \psi_{i,j}(W_1, \dots, W_q)$  where  $i = \gamma(U)$  and  $j = \gamma(U')$ .

- If  $G_{i,j} \models \psi_{i,j}(W_1, \dots, W_q)$ , then we get  $G \models \varphi(W_1, \dots, W_q)$  by a similar argument (in particular  $d_{G_{i,j}}(x, y) > 2t$  implies  $d_{G[U \cup U']}(x, y) > 2t$  which implies that  $d_G(x, y) > 2t$  by Claim 5.4).

For  $m = 1$ , the proof is similar by using a distance-2 coloring  $\gamma$  and the graphs  $G_{i,i}$  instead of the graphs  $G_{i,j}$ .

For the case  $m > 2$ , the proof is similar. One takes for  $\gamma$  a distance- $(m + 1)$  coloring of the intersection graph  $\mathcal{I}(\mathcal{T})$  and considers the graphs  $G_{i_1, \dots, i_m}$  defined as the unions of pairwise non touching sets  $U_1 \cup \dots \cup U_{m'}$  for  $m' \leq m$  and  $U_1, \dots, U_{m'} \in \mathcal{T}$ , of respective colors  $i_1, \dots, i_{m'}$ . Thus, we have  $cwd(G[U_1 \cup \dots \cup U_{m'}]) \leq g(m)$  and  $cwd(G_{i_1, \dots, i_m}) \leq g(m)$ .

By hypothesis, the cover  $\mathcal{T}$  is computed in time  $f(n)$  for an  $n$ -vertex graph  $G$  in  $\mathcal{C}$ . In each graph  $G_{i_1, \dots, i_m}$ , the labeling  $J_{i_1, \dots, i_m}$  is constructed in cubic-time by Theorem 2.5. The coloring  $\gamma$  uses  $\ell^{O(m)}$  colors. Then the number of graphs  $G_{i_1, \dots, i_m}$  is bounded by  $\ell^{O(m^2)}$ . Hence, the labeling  $J$  can be computed in time  $O(f(n) + n^3)$ .

We now examine the time taken to check the validity of  $\varphi(\overline{W})$ . For each  $G_{i_1, \dots, i_m}$  and each  $W_i$ , it takes time  $O(\log(n) \cdot |W_i|)$  to determine  $W_i \cap V_{G_{i_1, \dots, i_m}}$ . By Theorem 2.5, it takes time  $O(\log(n) \cdot (|\overline{W}| + 1))$  to check in  $G_{i_1, \dots, i_m}$  the validity of  $\varphi(\overline{W})$ . This concludes the proof of Theorem 5.1.  $\square$

Let us ask a very general question: what can be done with labels of size  $O(\log(n))$ ? Here is a fact that limits the extension of these results.

Let  $\varphi_0(x, y)$  be the  $t$ -local and bounded FO formula telling us whether two distinct vertices  $x$  and  $y$  are connected by a path of length 2:

$$x \neq y \wedge \exists z (z \neq x \wedge z \neq y \wedge \text{edg}(x, z) \wedge \text{edg}(z, y)).$$

The *adjacency query* is the query defined by  $\text{edg}(x, y) \vee \text{edg}(y, x)$  for a graph without edge colors. The following proposition uses a reduction from the case of adjacency queries in arbitrary graphs. We recall that the adjacency query has a log-labeling scheme in graphs of bounded arboricity (Theorem 5.1 (1)).

**Proposition 5.5** *Every labeling scheme supporting  $\varphi_0$  on graphs with  $n$  vertices and of arboricity at most 2 requires labels of length at least  $\sqrt{\frac{n}{2}} - 1$  for some graphs.*

**Proof.** We first define a construction to be used in a reduction between problems. With every simple, loop-free and undirected graph  $G$ , we associate the



graph  $\tilde{G}$  obtained by inserting a vertex  $z_{xy}$  on each edge  $xy$ .

$$\begin{aligned} V_{\tilde{G}} &:= V_G \cup \{z_{x,y} \mid x, y \in V_G \text{ and } xy \in E_G\}, \\ E_{\tilde{G}} &:= \{xz_{x,y} \mid xy \in E_G\}. \end{aligned}$$

The following properties hold.

- (1)  $V_G \subseteq V_{\tilde{G}}$  and  $|V_{\tilde{G}}| = |V_G| + |E_G|$ .
- (2) For all  $x$  and  $y$  in  $V_G$ , the edge  $xy$  is in  $E_G$  if and only if  $\tilde{G} \models \varphi_0(x, y)$ .
- (3)  $\tilde{G}$  has arboricity at most 2.

The first two points are clear. For the third one, we orient each edge  $e$  of  $G$  and we get a directed graph, that we denote by  $\vec{G}$ . We let:

$$\begin{aligned} F_1 &:= \{xz_{x,y} \mid (x, y) \in E_{\vec{G}}\}, \\ F_2 &:= \{z_{x,y}y \mid (x, y) \in E_{\vec{G}}\}. \end{aligned}$$

Neither  $F_1$  nor  $F_2$  has a cycle in  $\tilde{G}$ . Then  $\tilde{G}$  has arboricity at most 2 since  $(F_1, F_2)$  is a bipartition of  $E_{\tilde{G}}$ .

Let  $k : \mathbb{N} \rightarrow \mathbb{N}$  be a mapping and  $\mathcal{D}$  a class of undirected graphs. If for every graph  $H \in \mathcal{D}$  with  $n$  vertices, we can label its vertices with a bit sequence of size  $k(n)$  and check the adjacency query by using these bit sequences, then the number of non isomorphic graphs of size  $n$  in  $\mathcal{D}$  is bounded by  $2^{n \cdot k(n)}$ . Since the number of simple undirected graphs with  $n$  vertices is  $2^{n \cdot (n-1)/2}$ , any labeling scheme supporting the adjacency query in simple undirected graphs with  $n$  vertices requires some labels of size at least  $(n-1)/2$ . Hence, the adjacency query requires labels of size  $\lfloor n/2 \rfloor$  in some graphs. Using (2) above, we conclude that any labeling scheme for  $\varphi_0$  on the graph family  $\mathcal{F}_n := \{\tilde{G} \mid G \text{ has } n \text{ vertices}\}$  requires labels of size at least  $\lfloor \frac{n}{2} \rfloor$ . Let  $\tilde{G}$  be in  $\mathcal{F}_n$  and let  $\tilde{n} := |V_{\tilde{G}}|$ . Using (1), we have  $\tilde{n} = n + |E_G| \leq \frac{n(n+1)}{2}$ , i.e.,  $n \geq \sqrt{2\tilde{n}} - 1$ . Hence, any labeling scheme for  $\varphi_0$  on  $\mathcal{F}_n$  requires for some graphs with  $\tilde{n}$  vertices labels of size at least  $\lfloor \frac{\sqrt{2\tilde{n}}-1}{2} \rfloor > \sqrt{\frac{\tilde{n}}{2}} - 1$ .  $\square$

## 6 Extension to Counting Queries

By a graph, we still mean a  $(C_1, C_2)$ -graph, i.e., a binary relational structure. The results do not depend on  $(C_1, C_2)$ . We will not specify this pair at each time. We now consider an extension to counting queries.

**Definition 6.1 (Counting Query)** *Let  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  be an MSO*

formula and let  $G$  be a (colored) graph. For  $W_1, \dots, W_q \subseteq V_G$ , we let

$$\#_G \varphi(W_1, \dots, W_q) := \left| \left\{ (a_1, \dots, a_m) \in V_G^m \mid G \models \varphi(a_1, \dots, a_m, W_1, \dots, W_q) \right\} \right|.$$

The counting query of  $\varphi$  consists in determining  $\#_G \varphi(W_1, \dots, W_q)$  for given  $(W_1, \dots, W_q)$ <sup>13</sup>. If  $s \geq 2$ , the counting query of  $\varphi$  modulo  $s$  consists in determining  $\#_G \varphi(W_1, \dots, W_q)$  modulo  $s$  for given  $(W_1, \dots, W_q)$ .

The following theorem is an easy extension of Theorem 2.5.

**Theorem 6.2** *Let  $k$  be a positive integer,  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  an MSO formula over graphs and  $s \geq 2$ . There exists a  $\log^2$ -labeling scheme (resp. a  $\log$ -labeling scheme)  $(\mathcal{A}, \mathcal{B})$  on the class of graphs of clique-width at most  $k$  for the counting query of  $\varphi$  (resp. the counting query of  $\varphi$  modulo  $s$ ). Moreover, if the input graph has  $n$  vertices, the algorithm  $\mathcal{A}$  constructs the labels in time  $O(n^3)$  or in time  $O(n \cdot \log(n))$  if the clique-width expression is given, and the algorithm  $\mathcal{B}$  computes  $\#_G \varphi(W_1, \dots, W_q)$  in time  $O(\log^2(n) \cdot (|\overline{W}| + 1))$  (resp.  $O(\log(n) \cdot (|\overline{W}| + 1))$ ).*

**Proof Sketch.** The proof of Theorem 2.5 given in [10] builds labels that are sequences of states of an automaton on terms representing graphs, constructed from the MSO formula  $\varphi$ . An extension of this construction, that is given in [10], makes it possible to compute

$$\max \left\{ |U| \mid U \subseteq V_G \text{ and } G \models \varphi(U, W_1, \dots, W_q) \right\}$$

where  $\varphi(X, Y_1, \dots, Y_q)$  is an MSO formula and  $W_1, \dots, W_q$  are given sets of vertices. Instead of sequences of states, the labels consist of sequences of mappings from the set of states to  $[[V_G]]$ . It follows that the size of these labels is  $O(\log^2(n))$  instead of  $O(\log(n))$ . For a counting query, a similar technique can be used, with mappings from the set of states to  $[[V_G]]^m$ . Since  $m$  is fixed, labels have still size  $O(\log^2(n))$ .

If we wish to count answers modulo an integer  $s$ , we need only mappings from the set of states to  $\{0, \dots, s-1\}^m$  and the labels have size  $O(\log(n))$ .  $\square$

We will prove a similar theorem for nicely locally cwd-decomposable classes of graphs and FO formulas.

<sup>13</sup>In order to simplify the notation, we let  $Y_1, \dots, Y_q$  be set variables. This is not a loss of generality because a first-order variable  $y$  can be replaced by a set variable  $Y$  subject to the condition that its value is a singleton.

**Theorem 6.3 (Second Main Theorem)** *Let  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  be an FO formula and let  $s \geq 2$ . There exists a  $\log^2$ -labeling scheme (resp. a log-labeling scheme)  $(\mathcal{A}, \mathcal{B})$  for the counting query of  $\varphi$  (resp. the counting query of  $\varphi$  modulo  $s$ ) on nicely locally cwd-decomposable classes. Moreover, if the input graph has  $n$  vertices, the algorithm  $\mathcal{A}$  constructs the labels in time  $O(f(n) + n^3)$  where  $f(n)$  is the time taken to construct a nice cwd-cover, and the algorithm  $\mathcal{B}$  computes  $\#_G \varphi(W_1, \dots, W_q)$  in time  $O(\log^2(n) \cdot (|\overline{W}| + 1))$  (resp.  $O(\log(n) \cdot (|\overline{W}| + 1))$ ).*

We will first prove Theorem 6.3 for particular  $t$ -local formulas on locally cwd-decomposable classes.

**Definition 6.4 ( $t$ -Connected Formulas)** *A formula  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  is  $t$ -connected if for all graphs  $G$ , all  $a_1, \dots, a_m \in V_G$  and all  $W_1, \dots, W_q \subseteq V_G$ ,*

$$G \models \varphi(a_1, \dots, a_m, W_1, \dots, W_q) \quad \text{if and only if} \quad \begin{cases} \bigwedge_{1 \leq i < j \leq m} d(a_i, a_j) \leq t \text{ and} \\ G[N] \models \varphi(a_1, \dots, a_m, W_1 \cap N, \dots, W_q \cap N) \end{cases}$$

where  $N := N_G^t(\{a_1, \dots, a_m\})$ .

**Remark 6.5** *Let  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  be a  $t$ -connected formula. Then for all  $W \supseteq N_G^t(\{a_1, \dots, a_m\})$ , we have*

$$G \models \varphi(a_1, \dots, a_m, W_1, \dots, W_q) \quad \text{if and only if} \quad G[W] \models \varphi(a_1, \dots, a_m, W_1 \cap W, \dots, W_q \cap W)$$

and, since  $N_G^t(\{a_1, \dots, a_m\}) \subseteq N_G^{2t}(a_1)$  we have

$$G \models \varphi(a_1, \dots, a_m, W_1, \dots, W_q) \quad \text{if and only if} \quad G[N_G^{2t}(a_1)] \models \varphi(a_1, \dots, a_m, W_1, \dots, W_q).$$

**Lemma 6.6** *Let  $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$  be a  $t$ -connected formula and let  $s \geq 2$ . There exists a  $\log^2$ -labeling scheme (resp. a log-labeling scheme)  $(\mathcal{A}, \mathcal{B})$  for the counting query of  $\varphi$  (resp. the counting query of  $\varphi$  modulo  $s$ ) on locally cwd-decomposable classes of graphs. Moreover, if the input graph has  $n$  vertices, the algorithm  $\mathcal{A}$  constructs the labels in time  $O(f(n) + n^3)$  where  $f(n)$  is the time taken to construct a cwd-cover, and the algorithm  $\mathcal{B}$  computes  $\#_G \varphi(W_1, \dots, W_q)$  in time  $O(\log^2(n) \cdot (|\overline{W}| + 1))$  (resp.  $O(\log(n) \cdot (|\overline{W}| + 1))$ ).*

**Proof.** Let  $\mathcal{C}$  be a class of locally cwd-decomposable graphs and let  $G$  in  $\mathcal{C}$  be a graph with  $n$  vertices. Let  $\mathcal{T}$  be a  $(2t, \ell, g)$ -cwd cover of  $G$  and  $\gamma$  a distance-2 coloring of  $\mathcal{I}(\mathcal{T})$  with colors in  $[\ell^2 + 1]$ .

**Claim 6.7** *Let  $x \in K_G^{2t}(U)$  and  $y \in K_G^{2t}(U')$  with  $\gamma(U) = \gamma(U')$ ,  $U \neq U'$ . Then  $d_G(x, y) > 2t$ .*

**Proof of Claim 6.7.** If this is not the case, then  $y \in U$ . Hence,  $U$  and  $U'$  are adjacent in  $\mathcal{I}(\mathcal{T})$ . This is impossible since they have the same color.  $\square$

We color each vertex  $x$  of  $G$  by  $i$ , the smallest  $\gamma(U)$  such that  $x \in K_G^{2t}(U)$ . We represent this by the validity of  $p_i(x)$  as in the proof of Theorem 5.1 (4). For each  $i \in [\ell^2 + 1]$ , we let  $\varphi_i$  be the formula:

$$\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q) \wedge p_i(x_1).$$

Then the following is clear.

**Claim 6.8**  $\#_G \varphi(Y_1, \dots, Y_q) = \sum_i \#_G \varphi_i(Y_1, \dots, Y_q)$ .

We now show that the counting query of  $\varphi$  admits a  $\log^2$ -labeling scheme on  $G$ . We let  $V_i := \bigcup_{\gamma(U)=i} \{U \mid U \in \mathcal{T}\}$ .

**Claim 6.9**  $\text{cwd}(G[V_i]) \leq g$ .

**Proof of Claim 6.9.** Let  $U$  and  $U'$  in  $\mathcal{T}$  be subsets of  $V_i$ . Since  $\gamma$  is a distance-2 coloring, the sets  $U$  and  $U'$  are disjoint, otherwise they will have different colors. They do not touch, otherwise there exist a vertex  $x \in U$  and a vertex  $y \in U'$  such that  $xy \in E_G$ . Then by construction of  $\mathcal{T}$ , there exists a set  $W$  in  $\mathcal{T}$  such that  $x \in K_G^{2t}(W)$ . Therefore,  $U$  and  $U'$  are at distance at most 2 in  $\mathcal{I}(\mathcal{T})$  and thus have different colors. A contradiction because  $\gamma(U) = \gamma(U') = i$ . Therefore,  $G[V_i]$  is a union of graphs that pairwise do not touch and have clique-width at most  $g$ . We can thus conclude that  $\text{cwd}(G[V_i]) \leq g$ .  $\square$

**Claim 6.10**  $\#_G \varphi_i(Y_1, \dots, Y_q) = \#_{G[V_i]} \varphi_i(Y_1, \dots, Y_q)$ .

**Proof of Claim 6.10.** If  $\varphi(a_1, \dots, a_m, W_1, \dots, W_q)$  holds and  $p_i(a_1)$  holds, then  $a_1 \in K_G^{2t}(U)$  for some  $U$  such that  $\gamma(U) = i$ . Hence,  $a_2, \dots, a_m \in N_G^{2t}(a_1)$  and  $G[N_G^{2t}(a_1)] \models \varphi_i(a_1, \dots, a_m, W_1, \dots, W_q)$ . Thus,  $G[V_i] \models \varphi_i(a_1, \dots, a_m, W_1, \dots, W_q)$ .

If  $G[V_i] \models \varphi_i(a_1, \dots, a_m, W_1, \dots, W_q)$ , then  $p_i(a_1)$  holds and  $d_{G[V_i]}(a_l, a_s) \leq t$  for all  $l$  and  $s$  in  $[m]$ . But,  $d_G(a_l, a_s) = d_{G[V_i]}(a_l, a_s) = d_{G[U]}(a_l, a_s)$  where  $a_1 \in U$  and  $\gamma(U) = i$ . And since  $N_G^t(\{a_1, \dots, a_m\}) \subseteq V_i$ , we have  $G \models \varphi_i(a_1, \dots, a_m, W_1, \dots, W_q)$ .  $\square$

By Theorem 6.2, and Claims 6.9 and 6.10, there exists a  $\log^2$ -labeling  $J_i$  for the counting query of each  $\varphi_i$ . For each  $x \in V_G$ , we let  $J(x) := (J_1(x), \dots, J_{\ell^2+1}(x))$ .

Hence,  $J$  is a  $\log^2$ -labeling for the counting query of  $\varphi$  by Claim 6.8. By Theorem 6.2, labels of size  $O(\log(n))$  are sufficient for the counting query of each  $\varphi_i$  modulo  $s$ .

By Theorem 6.2, each labeling  $J_i$  is constructed in cubic-time. Therefore, the labeling  $J$  is constructed in time  $O(f(n) + n^3)$  where  $f(n)$  is the time taken for constructing the  $(2t, \ell, g)$ -cwd cover  $\mathcal{T}$  of  $G$ . By Claim 6.8 and Theorem 6.2,  $\mathcal{B}$  computes  $\#_G \varphi(W_1, \dots, W_q)$  in time  $O(\log^2(n) \cdot (|\overline{W}| + 1))$  (resp.  $O(\log(n) \cdot (|\overline{W}| + 1))$ ).  $\square$

We now prove Theorem 6.3.

**Proof of Theorem 6.3.** Let  $\varphi(\bar{x}, \overline{Y})$  be an FO formula with  $\bar{x} := (x_1, \dots, x_m)$  and  $\overline{Y} := (Y_1, \dots, Y_q)$ . By Theorem 3.4,  $\varphi$  is logically equivalent to a Boolean combination  $B$  of  $t$ -local formulas around  $\bar{x}$  and of basic  $(t', s)$ -local formulas. The Boolean combination  $B$  can be written as a disjunction of mutually exclusive conjunctions of formulas. Hence, we can reduce the counting query of  $\varphi$  to that of finitely many formulas of the form:

$$\varphi'(\bar{x}, \overline{Y}) \wedge \psi(\overline{Y})$$

where  $\varphi'(\bar{x}, \overline{Y})$  is a  $t$ -local formula around  $\bar{x}$ , and  $\psi(\overline{Y})$  is a conjunction of basic  $(t', s)$ -local formulas and of negations of basic  $(t', s)$ -local formulas. We have proved that each basic  $(t', s)$ -local formula admits a log-labeling scheme on each nicely locally cwd-decomposable class of graphs (Theorem 5.1 (4)). It remains then to prove that the counting query of a  $t$ -local formula admits a  $\log^2$ -labeling scheme on each nicely locally cwd-decomposable class of graphs  $\mathcal{C}$ .

Let  $\psi(\bar{x}, Y_1, \dots, Y_q)$  be a  $t$ -local formula around  $\bar{x} := (x_1, \dots, x_m)$ . By Theorem 3.5, we can reduce the counting query of  $\psi$  to the counting query of finitely many formulas of the form  $\rho_{t, \Delta}(\bar{x}) \wedge \varphi'(\bar{x}, Y_1, \dots, Y_q)$  that can be expressed as

$$\varphi'(\bar{x}, Y_1, \dots, Y_q) := \bigwedge_{1 \leq i < j \leq p} d(\bar{x} \upharpoonright \Delta_i, \bar{x} \upharpoonright \Delta_j) > 2t + 1 \wedge \bigwedge_{1 \leq i \leq p} \varphi_i(\bar{x} \upharpoonright \Delta_i, Y_1, \dots, Y_q)$$

where each  $\varphi_i$  is  $t$ -local and  $(m \cdot (2t + 1))$ -connected. To simplify the writing, we will assume that  $\psi$  is of the form  $\varphi'(\bar{x}, Y_1, \dots, Y_q)$ .

Let  $G$  with  $n$  vertices be in  $\mathcal{C}$ . Let  $\mathcal{T}$  be a nice  $(r, \ell, g)$ -cwd cover of  $G$  where  $r := m \cdot (2t + 1)$  and let  $\gamma$  be a distance- $(m + 1)$  coloring of  $\mathcal{I}(\mathcal{T})$ , the intersection graph of  $\mathcal{T}$ . For every  $m$ -tuple of colors  $(i_1, \dots, i_m)$ , we let  $G_{i_1, \dots, i_m}$  be the graph  $G[V]$  where  $V$  is the union of all sets  $U \in \mathcal{T}$  such that  $\gamma(U) \in \{i_1, \dots, i_m\}$ . We have  $\text{cwd}(G[V]) \leq g(m)$  (by the same arguments as in Claim 5.3 and the

proof of Theorem 5.1 (4)). We color each vertex with the smallest color  $i$  such that  $x \in K_G^r(U)$  and  $\gamma(U) = i$  and we express this by the validity of  $p_i(x)$ . We let  $\varphi'_{i_1, \dots, i_m}$  be

$$\bigwedge_{1 \leq i < j \leq p} d(\bar{x} \upharpoonright \Delta_i, \bar{x} \upharpoonright \Delta_j) > 2t + 1 \wedge \bigwedge_{1 \leq \ell \leq p} (\varphi_\ell(\bar{x} \upharpoonright \Delta_\ell, Y_1, \dots, Y_q) \wedge p_{i_\ell}(z_\ell))$$

where  $z_\ell$  is the first variable of each tuple  $\bar{x} \upharpoonright \Delta_\ell$ . We have:

**Claim 6.11**  $\#_G \psi(Y_1, \dots, Y_q) = \sum_{(i_1, \dots, i_m)} \#_G \varphi'_{i_1, \dots, i_m}(Y_1, \dots, Y_m)$ .

We let  $H := G_{i_1, \dots, i_m}$ . By the same arguments as in the proof of Claim 5.4 we have:

**Claim 6.12**  $d_G(\bar{x} \upharpoonright \Delta_i, \bar{x} \upharpoonright \Delta_j) > 2t + 1$  if and only if  $d_H(\bar{x} \upharpoonright \Delta_i, \bar{x} \upharpoonright \Delta_j) > 2t + 1$ .

It follows that:

**Claim 6.13**  $\#_G \varphi'_{i_1, \dots, i_m}(Y_1, \dots, Y_q) = \#_H \varphi'_{i_1, \dots, i_m}(Y_1, \dots, Y_q)$ .

By Theorem 6.2 and Claims 6.11, 6.12 and 6.13, there exists a  $\log^2$ -labeling scheme for the counting query of each  $t$ -local formula, and a  $\log$ -labeling scheme is enough for modulo counting.

By hypothesis, a nice  $(r, \ell, g)$ -cwd cover  $\mathcal{T}$  of  $G$  can be constructed in time  $f(n)$ . For each formula  $\varphi_{i_1, \dots, i_m}$ , the associated labeling  $J_{i_1, \dots, i_m}$  is constructed in time  $O(n^3)$  by Theorem 6.2. The coloring  $\gamma$  uses  $\ell^{O(m)}$  colors. The number of graphs  $G_{i_1, \dots, i_m}$  is bounded by  $\ell^{O(m^2)}$ . Hence, the labeling  $J$  is computed in time  $O(f(n) + n^3)$ . By Claim 6.11 and Theorem 6.2, the algorithm  $\mathcal{B}$  computes  $\#_G \varphi(W_1, \dots, W_q)$  in time  $O(\log^2(n) \cdot (|\overline{W}| + 1))$  (resp.  $O(\log(n) \cdot (|\overline{W}| + 1))$ ). This finishes the proof.  $\square$

The constructions of Theorems 6.2 and 6.3 can be adapted so as to produce, for given sets  $W_1, \dots, W_q$  (given as sets of labels) an  $m$ -tuple  $(\ulcorner a_1 \urcorner, \dots, \ulcorner a_m \urcorner)$  for one answer  $\bar{a}$  to the parametrized query  $\varphi$ , and  $\ulcorner a \urcorner$  is the coding of vertex  $a$ . Labels of length  $O(\log(n))$  suffice for that.

## 7 Conclusion

We conjecture that the results of Theorem 5.1 (3-5) extend to classes of graphs that exclude, or locally exclude a minor (definitions are in [11,19]).

**Question 1** *Does there exist a log-labeling scheme for parametrized FO queries on locally cwd or tree-decomposable classes?*

## Acknowledgements

We thank the referees for many helpful comments.

## References

- [1] S. Arnborg, J. Lagergren and D. Seese. Easy Problems for Tree-Decomposable Graphs. *Journal of Algorithms* 12(2):308-340, 1991.
- [2] G. Bagan. *Algorithmes et Complexité des Problèmes d'Énumération pour l'Évaluation de Requêtes Logiques*. PhD Thesis, Université de Caen/Basse Normandie, Caen, 2009.
- [3] A. Blumensath and B. Courcelle. Recognizability, Hypergraph Operations and Logical Types. *Information and Computation* 204(6):853-919, 2006.
- [4] H.L. Bodlaender. A Linear-Time Algorithm for Finding Tree-Decompositions of Small Tree-width. *SIAM Journal on Computing* 25(6):1305-1317, 1996.
- [5] H.L. Bodlaender. Tree-Width: Structure and Algorithms. In G. Principe and S. Zaks eds., *Structural Information and Communication Complexity (SIROCCO)*, volume 4474 of *LNCS*, pages 11-25. Springer, 2007.
- [6] B. Courcelle, C. Gavaille, M.M. Kanté and A. Twigg. Optimal Labeling for Connectivity Checking in Planar Networks with Obstacles. Manuscript, 2008. An extended abstract about 3-connected planar graphs is published in *Electronic Notes in Discrete Mathematics* 31:151-155, 2008, the proceedings of the first Conference *Topological and Geometric Graph Theory (TGGT)*, Paris, 2008.
- [7] B. Courcelle, J.A. Makowsky and U. Rotics. Linear-Time Solvable Optimization Problems on Graphs of Bounded Clique-Width. *Theory of Computing Systems* 33(2):125-150, 2000.
- [8] B. Courcelle and S. Olariu. Upper Bounds to the Clique-Width of Graphs. *Discrete Applied Mathematics* 101(1-3):77-114, 2000.
- [9] B. Courcelle and A. Twigg. Constraint-Path Labelings on Graphs of Bounded Clique-Width. To appear in *Theory of Computing Systems*, 2009.
- [10] B. Courcelle and R. Vanicat. Query Efficient Implementation of Graphs of Bounded Clique-Width. *Discrete Applied Mathematics* 131(1):129-150, 2003.

- [11] A. Dawar, M. Grohe and S. Kreutzer. Locally Excluding a Minor. In *22<sup>nd</sup> IEEE Symposium on Logic in Computer Science (LICS)*, pages 270-279. IEEE Computer Society, 2007.
- [12] R. Diestel. *Graph Theory*. Springer, 3<sup>rd</sup> edition, 2005.
- [13] A. Durand and E. Grandjean. First-Order Queries on Structures of Bounded Degree are Computable with Constant Delay. *ACM Transactions on Computational Logic* 8(4), 2007.
- [14] D. Eppstein. Arboricity and Bipartite Subgraph Listing Algorithms. *Information Processing Letters* 51(4):207-211, 1994.
- [15] E. Fisher, J.A. Makowsky and E.V. Ravve. Counting Truth Assignments of Formulas of Bounded Tree-Width or Clique-Width. *Discrete Applied Mathematics* 156(4):511-529, 2008.
- [16] M. Frick. Generalized Model-Checking over Locally Tree-Decomposable Classes. *Theory of Computing Systems* 37(1):157-191, 2004.
- [17] M. Frick and M. Grohe. Deciding First-Order Properties of Locally Tree-Decomposable Structures. *Journal of the ACM* 48(1):1184-1206, 2001.
- [18] H. Gaifman. On Local and Non-Local Properties. In J. Stern ed, *Proceedings of the Herbrand Symposium, Logic Colloquium'81*, pages 105-135. North-Holland Publishing Company, 1982.
- [19] M. Grohe. Logic, Graphs and Algorithms. In Flum, Grädel and Wilke eds., *Logic, Automata, History and Perspectives*, pages 357-422. Amsterdam University Press, 2007.
- [20] C. Gavaille and D. Peleg. Compact and Localized Distributed Data Structures. *Distributed Computing* 16(2-3):111-120, 2003.
- [21] F. Gurski and E. Wanke. Line Graphs of Bounded Clique-Width. *Discrete Mathematics* 307(22):2734-2754, 2007.
- [22] P. Hliněný and S. Oum. Finding Branch-Decompositions and Rank-Decompositions. *SIAM Journal on Computing* 38(3):1012-1032, 2008.
- [23] M. Kamiński, V. Lozin and M. Milanič. Recent Developments on Graphs of Bounded Clique-Width. *Discrete Applied Mathematics*, in press, 2008.
- [24] M. M. Kanté. *Graph Structurings: Some Algorithmic Applications*. PhD thesis, Université Bordeaux 1, Bordeaux, 2008.
- [25] S. Kreutzer. Algorithmic Meta-Theorems. In M. Grohe and R. Neidermeier eds, *International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 5018 of *LNCS*, pages 10-12. Springer, 2008. A full version is available at <http://arxiv.org/abs/0902.3616>.
- [26] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.



- [27] V. Lozin. From Tree-Width to Clique-Width: Excluding a Unit-Interval Graph. In S. Hong, H. Nagamochi and T. Fukunaga eds, *International Symposium on Algorithms and Computation (ISAAC)*, volume 5369 of *LNCS*, pages 871-882. Springer, 2008.
- [28] J. Nešetřil and P. Ossona de Mendez. Linear Time Low Tree-Width Partitions and Algorithmic Consequences. In J.M. Kleinberg ed., *38<sup>th</sup> Annual ACM Symposium on Theory of Computing (STOC)*, pages 391-400. ACM, 2006.
- [29] S. Oum and P. Seymour. Approximating Clique-Width and Branch-Width, *Journal of Combinatorial Theory, Series B*, 96(4):514-528, 2006.
- [30] D. Seese. Linear Time Computable Problems and First-Order Descriptions. *Mathematical Structures in Computer Science* 6(6):505-526, 1996.
- [31] K. Suchan and I. Todinca. On powers of graphs of bounded NLC-Width (Clique-Width). *Discrete Applied Mathematics* 155(14):1885-1893, 2007.