Learnability and Definability in Trees and Similar Structures

M. Grohe* Gy. Turán†

May 29, 2003

Abstract

We prove upper bounds for combinatorial parameters of finite relational structures, related to the complexity of learning a definable set. We show that monadic second order (MSO) formulas with parameters have bounded Vapnik-Chervonenkis dimension over structures of bounded clique-width, and first-order formulas with parameters have bounded Vapnik-Chervonenkis dimension over structures of bounded local clique-width (this includes planar graphs). We also show that MSO formulas of a fixed size have bounded strong consistency dimension over MSO formulas of a fixed larger size, for labeled trees. These bounds imply positive learnability results for the PAC and equivalence query learnability of a definable set over these structures. The proofs are based on bounds for related definability problems for tree automata.

1. Introduction

The general problem of *concept learning* is to identify an unknown set from a given family of sets. The given family, referred to as the *concept class*, represents the possible classifications of the elements of the universe, and the unknown set, also called the *target concept*, is the actual classification. The identification can be exact or approximate (for example, in a probabilistic sense, as in the model of Probably Approximately Correct (PAC) learning). In order to specify a formal model of learning, one also has to determine what type of information is available to the learner (for example, random examples or certain types of queries), and what complexity measures are considered (for example, sample size, number of queries or computation time).

It is an interesting fact that several notions of learning complexity turn out to be closely related to certain *combinatorial parameters* of the concept class. The best known example is sample size for PAC learning, which is of the same order of magnitude as the *Vapnik-Chervonenkis dimension* or *VC-dimension* of the concept class (see, e.g., [23]). Other examples include the query complexity of learning with equivalence and membership queries, closely related to *certificate size* [21], and the query complexity of learning with equivalence queries, closely related to the *strong consistency dimension* [6]. Determining these combinatorial parameters for specific concept classes thus gives useful information about learning complexity.

The measures mentioned above are related to the *informational complexity* of learning and not to its *computational complexity*. There are also results relating learning complexity to the complexity of *computational problems* associated with the concept class, such as the *hypothesis finding problem* for PAC learning (see [23]) and the *representation problem* for learning with equivalence and membership queries [1].

In this paper we consider some of the informational complexity measures for learning in the context of *predicate logic*, which is a frequently used framework in machine learning, besides, for example, propositional logic and neural networks.

A general setup for predicate logic learning is to assume that examples are elements (or, more generally, tuples of elements) of a *finite structure*, and concepts are represented by a *class of predicate logic formulas*. This framework can be appropriate when the data for learning are provided by a relational database, but some other models can also be represented in this way. For instance, a standard setup for *inductive logic programming* is to learn a single non-recursive Horn clause with ground background knowledge (see, e.g., [28]). This is

*Laboratory for Foundations of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, Scotland, UK. Email: grohe@inf.ed.ac.uk

†Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 851 S. Morgan Street, Chicago, IL 60607-7045, USA and Research Group on AI of the Hungarian Academy of Sciences, Szeged, Hungary. Email: gyt@uic.edu.
Partially supported by NSF grant CCR-9800070 and CCR-0100336 and OTKA grant T-025271.

equivalent to learning an existential sentence having a conjunction of atoms as its quantifier free part (also called a conjunctive query), over the finite structure formed by the ground atoms in the background knowledge.

In the predicate logic framework, questions about the learning-related combinatorial parameters lead to combinatorial questions about classes of *definable sets*, which are much studied in model theory, mostly for infinite structures [22]. In fact, one of the three simultaneous sources for the notion of VC-dimension is [35] in model theory (besides [33] and [40]). The relationship between our results and the related results in model theory is discussed at the end of Section 6. A model theoretic approach to learnability was proposed in [29], and some results for finite models are given in [27, 38]. In particular, [27] gives an explicit upper bound for the VC-dimension of first-order formulas over structures with unary predicates and a single unary function.

The *model checking* problem for finite models is to decide, given a finite structure and a formula with a variable assignment, whether the formula holds in the structure. There are several results for this problem, putting together a picture of the borderline between tractable and intractable cases [19]. These results establish connections between the *expressiveness* of a logic and the *computational complexity* of computational problems associated with the logic. Our results show that for the learning complexity we get a very similar borderline between tractable and intractable cases. On a more technical level, our results show that the techniques developed for the model checking problem, in particular the use of tree automata and the notions of graph width, are also useful in the learning context.

We show that the class of sets definable by *monadic second-order (MSO)* formulas with parameters has bounded VC-dimension for classes of structures of bounded *clique-width*. As a tool for proving this result, we introduce tree automata versions of definability in a structure. These notions appear to be new, and may be of some interest in themselves. On the other hand, we note that MSO-formulas can have unbounded VC-dimension on *grids*. We also show that *first-order (FO)* formulas with parameters have bounded VC-dimension for classes of structures of bounded *local clique-width*. This includes, for example, the class of *planar graphs* and all classes of graphs of *bounded degree*. The main step in proving these results is a lemma that may be of interest in itself; it states that bounded VC-dimension of first-order formulas is a *local* property of structures, that is, it only depends on bounded radius neighborhoods of elements of the structures. All these bounded VC-dimension results imply bounded sample complexity for PAC-learnability using the standard results of computational learning theory (see [23]).

Besides discussing the VC-dimension, we also give some initial results on the strong consistency dimension [6], a much less studied and understood notion. The potential relevance of the strong consistency dimension for logic is discussed in Balcázar [4]. We show that on labeled trees, fixed size MSO formulas with one free variable and several parameters have a bounded strong consistency dimension with respect to MSO formulas of some fixed, larger size. Using the characterization given by [6], this implies that on trees, such MSO formulas can be learned using *logarithmically many* (in the size of the underlying structure) *equivalence queries* that are larger, but fixed size MSO formulas. This, perhaps somewhat surprising, result is among the first applications of the strong consistency dimension for proving positive learnability results. We note that the result is not practical, for two reasons: it does not provide an efficient algorithm to compute the queries, and it involves huge constants. [27] gives a computationally efficient query learning algorithm for quantifier-free formulas over structures with unary predicates and functions.

The paper is organized as follows. Sections 2 and 3 give background for definability and the VC-dimension. Sections 4 and 5 contain the automaton and MSO definability results for trees, respectively, for graphs of bounded clique-width and tree-width. First-order definability results for structures of bounded local clique-width and tree-width are given in Section 6. The strong consistency dimension results are presented in Section 7. Finally, Section 8 contains some further remarks and open problems. The Appendix gives a schematic overview of all the classes of structures considered in this paper.

2. Definability

A *vocabulary* is a finite set of relation symbols. In the following, τ always denotes a vocabulary. A *τ-structure* $\mathcal{A}$ consists of a non-empty set A, called the *universe* of $\mathcal{A}$, and a relation $R^{\mathcal{A}} \subseteq A^r$ for every r-ary relation symbol $R \in \tau$. For a vocabulary $\tau' \supseteq \tau$, a *τ'-expansion* of a τ-structure $\mathcal{A}$ is a τ'-structure $\mathcal{A}'$ with universe $A' = A$ and $R^{\mathcal{A}'} = R^{\mathcal{A}}$ for all $R \in \tau$. *Unless explictly mentioned otherwise, in this paper, we will only consider structures whose universe is finite.*

An *atomic formula*, or *atom*, is a formula of the form $x = y$ or $Rx_1 \dots x_r$, where R is an r-ary relation symbol and $x, y, x_1, \dots, x_r$ are *(individual) variables*. The formulas of *first-order logic* are built up from atomic formulas using the usual Boolean connectives and existential and universal quantification over the elements of the universe of a structure. The class of all formulas of first-order logic is denoted by FO.

Monadic second-order logic is the extension of first-order logic allowing quantification not only over elements of the universe of a structure, but also over subsets of the universe. Formally, we have two types of variables — *individual variables*, which are interpreted by elements of the universe of a structure, and *set variables*, which are interpreted by subsets of the universe of a structure. In addition to the first-order atoms, in monadic second-order logic we also have atoms Xx saying that the element interpreting the individual variable x is contained in the set interpreting the set variable X. Furthermore, we have existential and universal quantification over both individual and set variables. MSO denotes the set of all formulas of monadic second-order logic.

We always use lowercase letters $x, y, \dots$ to denote individual variables and uppercase letters $X, Y, \dots$ to denote set variables. A *free variable* of a formula φ is an (individual or set) variable v that does not occur in the scope of a quantifier $\exists v$ or $\forall v$. The set of free variables of a formula φ is denoted by free(φ). A *sentence* is a formula without free variables. We write $\varphi(X_1, \dots, X_k, x_1, \dots, x_l)$ to indicate that free$(\varphi) \subseteq \{X_1, \dots, X_k, x_1, \dots, x_l\}$. For a structure $\mathcal{A}$, subsets $A_1, \dots, A_k \subseteq A$, and elements $a_1, \dots, a_l \in A$ we write $\mathcal{A} \models \varphi(A_1, \dots, A_k, a_1, \dots, a_l)$ to denote that $\mathcal{A}$ satisfies φ if the set variables $X_1, \dots, X_k$ are interpreted by $A_1, \dots, A_k$, respectively, and the individual variables $x_1, \dots, x_l$ are interpreted by $a_1, \dots, a_l$, respectively. We only consider formulas that only have free individual variables.

For a formula $\varphi(x_1, \dots, x_k, y_1, \dots, y_l)$, a structure $\mathcal{A}$, and elements $b_1, \dots, b_l \in A$ we let

$$\varphi(\mathcal{A}, b_1, \dots, b_l) = \big\{(a_1, \dots, a_k) \in A^k \mid \mathcal{A} \models \varphi(a_1, \dots, a_k, b_1, \dots, b_l)\big\}.$$

We call $\varphi(\mathcal{A}, b_1, \dots, b_l)$ the set defined by φ in $\mathcal{A}$ with *parameters* $b_1, \dots, b_l$. We let

$$\mathcal{C}(\varphi, \mathcal{A}) = \{\varphi(\mathcal{A}, b_1, \dots, b_l) \mid b_1, \dots, b_l \in A\}.$$

We often denote tuples $(a_1, \dots, a_k)$ of elements of a set A by $\bar{a}$, and we write $\bar{a} \in A$ instead of $\bar{a} \in A^k$. Similarly, we denote tuples of individual variables by $\bar{x}$. For tuples $\bar{a}$ and $\bar{b}$, we write $\bar{a}\bar{b}$ to denote their concatenation.

The *quantifier-rank* of a first-order or monadic second-order formula φ, denoted by qr(φ), is the maximum number of nested quantifiers in φ. It is easy to see that for all $q, \ell \geq 0$, up to logical equivalence there are only finitely many first-order or monadic second-order formulas φ with qr$(\varphi) \leq q$ and $|\text{free}(\varphi)| \leq \ell$. The *size* of a formula φ is denoted by $||\varphi||$.

3. Vapnik–Chervonenkis dimension

Let V be a set and $\mathcal{C} \subseteq 2^V$ a family of subsets of V, also referred to as a *concept class*. For a subset $U \subseteq V$, we let $\mathcal{C} \cap U = \{C \cap U \mid C \in \mathcal{C}\}$. The set U is *shattered* by $\mathcal{C}$ if $\mathcal{C} \cap U = 2^U$.

The *Vapnik-Chervonenkis dimension*, or *VC-dimension*, of $\mathcal{C}$, denoted by VC$(\mathcal{C})$, is the maximum of the sizes of the shattered subsets of V, or ∞ if this maximum does not exist.

The Vapnik-Chervonenkis dimension characterizes the sample complexity needed for learning the concept class $\mathcal{C}$ in the PAC model of learning. For completeness, we give a brief description of this model.

For a function $m(\varepsilon, \delta)$, the concept class $\mathcal{C}$ is *PAC-learnable* with sample size $m(\varepsilon, \delta)$ if there is an algorithm which, given ε and δ, draws $m(\varepsilon, \delta)$ many random examples of an unknown target concept $C \in \mathcal{C}$ from a distribution P on X, and produces a hypothesis H from $\mathcal{C}$, such that $P(P(C \oplus H) \geq \varepsilon) \leq \delta$ for every C and P (where $C \oplus H$ denotes the symmetric difference of C and H). As we said in the introduction, we are only concerned with the informational complexity of learning in this paper and not with the computational complexity; this is why we ignore issues of algorithmic efficiency in the PAC-model.

Theorem 1 (Blumer, Ehrenfeucht, Haussler, Warmuth [8], Vapnik, Chervonenkis [40])
Every concept class $\mathcal{C}$ is PAC-learnable with sample size

$$m(\varepsilon, \delta) = O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta} + \frac{VC(\mathcal{C})}{\varepsilon}\log\frac{1}{\varepsilon}\right).$$

If $\mathcal{C}$ is PAC-learnable with sample size $m(\varepsilon, \delta)$ then

$$m(\varepsilon, \delta) = \Omega\left(\frac{1}{\varepsilon}\log\frac{1}{\delta} + \frac{VC(\mathcal{C})}{\varepsilon}\right).$$

The following combinatorial result is an important ingredient in the proof of Theorem 1, and we also use it in Section 6.

Theorem 2 (Sauer [33], Shelah [35], Vapnik and Chervonenkis [40]) *Let V be a set, $d \geq 1$, and $\mathcal{C} \subseteq 2^V$ such that $\mathrm{VC}(\mathcal{C}) \leq d$. Then for every set $U \subseteq V$ we have*

$$|\mathcal{C} \cap U| \leq \sum_{i=0}^{d} \binom{|U|}{i} = O(|U|^d).$$

For a formula $\varphi(\bar{x}, \bar{y})$ and a structure $\mathcal{A}$ we let $\mathrm{VC}(\varphi, \mathcal{A}) = \mathrm{VC}(\mathcal{C}(\varphi, \mathcal{A}))$. We say that a formula $\varphi(\bar{x}, \bar{y})$ has *bounded VC-dimension* on a class K of structures if there is a c such that for every $\mathcal{A} \in K$ we have $\mathrm{VC}(\varphi, \mathcal{A}) \leq c$.

The following standard example shows that even very simple formulas can have unbounded VC-dimension if we do not put any restriction on the structures considered.

Example 3 Let τ consist of a single binary relation E for graph adjacency, and consider the formula $\varphi = E(x, y)$. Let $\mathcal{G}_n$ be the $(n + 2^n)$-vertex graph, where for each subset of the first n vertices, there is a distinct vertex which is connected to just the vertices in this subset. Then clearly $VC(\varphi, \mathcal{G}_n) \geq n$.

In this paper, we shall show that MSO-formulas and FO-formulas have bounded VC-dimension on a variety of classes of structures. This following general result shows that in order to prove the boundedness of the VC-dimension for a class of structures, one may restrict attention to formulas with a single free variable.

Lemma 4 (Shelah [34]) *Let K be a class of structures such that every first-order formula $\varphi(x, \bar{y})$ has bounded VC-dimension on K. Then every first-order formula $\varphi(\bar{x}, \bar{y})$ has bounded VC-dimension on K.*

The analogous statement holds for formulas of monadic second-order logic.

Laskowski [24] gave a purely combinatorial proof of this result, which yields an explicit upper bound for the VC-dimension of $\varphi(\bar{x}, \bar{y})$. This is important for us, because in our results we also provide explicit bounds.

Usually, the lemma is only stated for FO, but it is easy to verify that Laskowski's proof goes through for MSO, and actually for any class of formulas that is closed under renaming of variables, Boolean combinations, and existential quantification (see [39]). Recall here that we only consider MSO-formulas with only free individual variables.

Remark 5 We will phrase our main results in the form: *Every formula of first-order logic or monadic second-order logic has bounded VC-dimension on a certain class of structures.* In a typical learning application, we may not be interested in the concept class $\mathcal{C}(\varphi, \mathcal{A})$ defined by a single formula φ, but actually in the class of all concepts defined by formulas of a certain size or quantifier rank, or more generally in the class $\bigcup_{\varphi \in \Phi} \mathcal{C}(\varphi, \mathcal{A})$ for some finite class Φ of formulas. However, it is a consequence of Theorem 2 that *for all m, c there is a d such that if concept classes $\mathcal{C}_1, \ldots, \mathcal{C}_m \subseteq 2^V$ have VC-dimension at most c then their union $\bigcup_{i=1}^{m} \mathcal{C}_i$ has VC-dimension at most d.* Thus whenever we prove that every formula of first-order or monadic second-order logic has bounded VC-dimension on some class K of structures, all finite unions $\bigcup_{\varphi} \mathcal{C}(\varphi, \mathcal{A})$ also have bounded VC-dimension on K.

4. Definability in trees

4.1. Trees. The trees we consider are finite ordered binary trees. We view such trees as $\{S_1, S_2, \preceq\}$-structures, where S_1, S_2, and $\preceq$ are binary relation symbols. In a tree $\mathcal{T} = (T, S_1^{\mathcal{T}}, S_2^{\mathcal{T}}, \preceq^{\mathcal{T}})$, $S_1^{\mathcal{T}}$ is the left child relation and $S_2^{\mathcal{T}}$ the right child relation. Moreover, $\preceq^{\mathcal{T}}$ is the tree-order, that is, the transitive closure of $S_1^{\mathcal{T}} \cup S_2^{\mathcal{T}}$. We do

not require each inner node of a tree to have exactly two children. Thus, in particular, we may also view strings as trees, in which every inner node only has a left child.

We are mainly interested in trees whose vertices are labeled with letters from some finite alphabet. For a finite alphabet Σ, we let $\tau(\Sigma) = \{S_1, S_2, \preceq\} \cup \{P_s \mid s \in \Sigma\}$, where for all $s \in \Sigma$, P_s is a unary relation symbol. A Σ-tree is a $\tau(\Sigma)$-structure

$$\mathcal{T} = \big(T, S_1^{\mathcal{T}}, S_2^{\mathcal{T}}, \preceq^{\mathcal{T}}, (P_s^{\mathcal{T}})_{s\in\Sigma}\big)$$

such that $\big(T, S_1^{\mathcal{T}}, S_2^{\mathcal{T}}, \preceq^{\mathcal{T}}\big)$ is an ordered binary tree and for each $a \in T$ there exists exactly one $s \in \Sigma$ such that $a \in P_s^{\mathcal{T}}$. We denote this a by $\sigma^{\mathcal{T}}(a)$

In order to be able to study subsets of trees defined by formulas with k free variables, for some $k \geq 1$, we let $\Sigma_k = \Sigma \times \{0,1\}^k$. For a Σ-tree $\mathcal{T}$ and a tuple $\bar{a} = (a_1, \ldots, a_k)$ of vertices of $\mathcal{T}$, we let $\mathcal{T}_{\bar{a}}$ be the Σ_k-tree with the same underlying tree as $\mathcal{T}$ and

$$\sigma^{\mathcal{T}_{\bar{a}}}(a) = (\sigma^{\mathcal{T}}(a), \varepsilon_1, \ldots, \varepsilon_k)$$

where $\varepsilon_i = 1$ if, and only if, $a = a_i$.

We may view $\mathcal{T}_{\bar{a}}$ as a Σ-tree with k distinguishable pebbles placed on $a_1, \ldots, a_k$, respectively.

4.2. Tree automata. Let Σ be a finite alphabet. A Σ-*tree automaton* is a tuple $\mathfrak{A} = (Q, \delta, F)$, where Q is a finite set of states, $F \subseteq Q$ is the set of accepting states, and $\delta : (Q \cup \{*\})^2 \times \Sigma \to Q$ is the transition function. Here $*$ is a special symbol not contained in Q.

A run $\rho : T \to Q$ of $\mathfrak{A}$ on a Σ-tree $\mathcal{T}$ is defined in a bottom-up manner. If a is a leaf then $\rho(a) = \delta(*, *, \sigma^{\mathcal{T}}(a))$. If a has two children b_1, b_2, then $\rho(a) = \delta(\rho(b_1), \rho(b_2), \sigma^{\mathcal{T}}(a))$. If a only has a left child b then $\rho(a) = \delta(\rho(b), *, \sigma^{\mathcal{T}}(a))$, and similarly if a only has a right child b then $\rho(a) = \delta(*, \rho(b), \sigma^{\mathcal{T}}(a))$. The automaton accepts $\mathcal{T}$ if $\rho(r) \in F$ for the root r of $\mathcal{T}$.

We are mainly interested in automata running on trees $\mathcal{T}_{\bar{a}}$, for some Σ-tree $\mathcal{T}$ and tuple $\bar{a} = (a_1, \ldots, a_k) \in T^k$. Recall that we view $\mathcal{T}_{\bar{a}}$ as $\mathcal{T}$ with pebbles placed on $a_1, \ldots, a_k$. Then a Σ_k-tree automaton running on $\mathcal{T}_{\bar{a}}$ is not only controlled by the labels $\sigma^{\mathcal{T}}(a)$ of the vertices a, but also by the pebbles placed on a. Instead of asking which trees the automaton accepts, in the following we will ask which pebble tuples on a fixed tree the automaton accepts. For a Σ_k-automaton $\mathfrak{A}$ and a Σ-tree $\mathcal{T}$, we let

$$\mathfrak{A}(\mathcal{T}) = \{\bar{a} \in T^k \mid \mathfrak{A} \text{ accepts } \mathcal{T}_{\bar{a}}\}.$$

In this sense, a Σ_k-tree automaton defines a k-ary relation on each Σ-tree.

4.3. The VC-dimension of automaton definable families. We need to extend the definition just made to definability with parameters. Let Σ be a finite alphabet, $\mathcal{T}$ a Σ-tree, $k, \ell \geq 1$, and $\mathfrak{A}$ a $\Sigma_{k+\ell}$-tree automaton. Then for every tuple $\bar{b} \in T^\ell$ we let

$$\mathfrak{A}(\mathcal{T}, \bar{b}) = \{\bar{a} \in T^k \mid \mathfrak{A} \text{ accepts } \mathcal{T}_{\bar{a}\bar{b}}\}.$$

Furthermore, we let

$$\mathcal{C}(\mathfrak{A}, \mathcal{T}) = \{\mathfrak{A}(\mathcal{T}, \bar{b}) \mid \bar{b} \in T^\ell\}$$

and

$$\mathrm{VC}(\mathfrak{A}, \mathcal{T}) = \mathrm{VC}(\mathcal{C}(\mathfrak{A}, \mathcal{T})).$$

The notations $\mathcal{C}(\mathfrak{A}, \mathcal{T})$ and $\mathrm{VC}(\mathfrak{A}, \mathcal{T})$ are slightly ambiguous, because they do not explicitly mention ℓ, but ℓ will always be clear from the context.

Theorem 6 *Let $\ell, m \geq 1$, and let $\mathfrak{A}$ be a $\Sigma_{1+\ell}$-tree automaton with m states. Then for every Σ-tree $\mathcal{T}$ we have:*

$$\mathrm{VC}(\mathfrak{A}, \mathcal{T}) < 8m(\ell + 1).$$

The proof depends on the following lemma:

Lemma 7 *Let $\ell, m, \mathfrak{A}, \mathcal{T}$ be as in the statement of the theorem and let U be a subset of T of size $p = 8m(\ell+1)$.*

Then there is a subset $Y \subseteq U$ such that for every $\bar{b} = (b_1, \ldots, b_\ell)$ there exist $c \in Y$ and $d \in U \setminus Y$ for which $\mathfrak{A}$ accepts $\mathcal{T}_{c\bar{b}}$ if, and only if, $\mathfrak{A}$ accepts $\mathcal{T}_{d\bar{b}}$.

Proof (of Lemma 7): The strategy of the proof is to find subsets $V_1, \ldots, V_{\ell+1}$ of T that contain many elements of U, but have 'little communication with each other', and then to form Y by a cut-and-paste argument based on these subsets.

We start by with a few remarks concerning our terminology: A *subtree* of a tree is a substructure that is itself a tree and that is upward closed with respect to the tree-order, which means that children of vertices in the subtree also belong to the subtree.

For a set Z of vertices in the tree, let the *largest common ancestor* of Z be the unique vertex $\mathrm{lca}(Z)$ such that $\mathrm{lca}(Z) \preceq^{\mathcal{T}} z$ for all $z \in Z$, and there is no y such that $\mathrm{lca}(Z) \prec^{\mathcal{T}} y$ and $y \preceq^{\mathcal{T}} z$ for all $z \in Z$. We let $G(Z)$, the *subgraph generated by* Z, be the union of all paths connecting the elements of Z with $\mathrm{lca}(Z)$. Thus $G(Z)$ is a tree, but it is in general different from the subtree rooted at $\mathrm{lca}(Z)$.

From the bottom up, take a minimal subtree of $\mathcal{T}$ (minimal with respect to inclusion) that contains at least $2m$ elements of U. Let U_1 be the elements of U in this subtree; then the root of this subtree is $\mathrm{lca}(U_1)$. As the tree is binary, it holds that $|U_1| < 4m$. Remove this subtree, and repeat the same procedure $2(\ell+1)$ times, to obtains sets $U_1, \ldots, U_{2(\ell+1)}$. Since $p = 4m \cdot 2(\ell+1)$, U is sufficiently large to form all the sets $U_1, \ldots, U_{2(\ell+1)}$. By construction, the subgraphs $G(U_i)$ are pairwise vertex disjoint and the vertices $\mathrm{lca}(U_i)$ are all distinct.

We define a binary relation F on $H = \{U_1, \ldots, U_{2(\ell+1)}\}$ to be the set of all pairs (U_i, U_j) such that $\mathrm{lca}(U_i) \prec^{\mathcal{T}} \mathrm{lca}(U_j)$ and there is no k with $\mathrm{lca}(U_i) \prec^{\mathcal{T}} \mathrm{lca}(U_k) \prec^{\mathcal{T}} \mathrm{lca}(U_j)$. Then (H, F) is a forest with $2(\ell+1)$ vertices and thus with at most $2\ell+1$ edges. Therefore, at most ℓ vertices of this forest have more than one child. Without loss of generality we can assume that $U_1, \ldots, U_{\ell+1}$ have at most 1 child.

If U_i has no children, we let $V_i = \{v \in T \mid \mathrm{lca}(U_i) \preceq^{\mathcal{T}} v\}$, i.e. the set of vertices of the subtree of $\mathcal{T}$ rooted at $\mathrm{lca}(U_i)$. If U_i has one child U_j, then we let $V_i = \{v \in T \mid \mathrm{lca}(U_i) \preceq^{\mathcal{T}} v, \mathrm{lca}(U_j) \not\preceq^{\mathcal{T}} v\}$, i.e. the set of all vertices of the subtree of $\mathcal{T}$ rooted at $\mathrm{lca}(U_i)$ that are not in the subtree rooted at $\mathrm{lca}(U_j)$.

Now we turn to the second part of the proof, the construction of Y based on the V_is. Let $1 \leq i \leq \ell+1$. If U_i has no children (in the forest (H, F)), then since $|U_i| > m$ we can find distinct vertices c_i, d_i of U_i such that:

(1) There is a state q^i of $\mathfrak{A}$ such that if none of the parameters $b_1, \ldots, b_\ell$ is contained in V_i and $\mathfrak{A}$ is running on either $\mathcal{T}_{c_i \bar{b}}$ or $\mathcal{T}_{d_i \bar{b}}$, then it reaches $\mathrm{lca}(U_i)$ in state q^i.

Now let us assume that U_i has a child U_j. Let the states of $\mathfrak{A}$ be $q_1, \ldots, q_m$. We define elements $c_{i,k}, d_{i,k}$ for $k = 1, \ldots, m$ by induction on k. Suppose that $1 \leq k \leq m$ and we have already defined $c_{i,k'}$ and $d_{i,k'}$ for $1 \leq k' < k$. Since $|U_i| \geq 2m$ we have $U_i \setminus \{d_{i,1}, \ldots, d_{i,k-1}\} > m$, therefore there are distinct elements $c_{i,k}, d_{i,k} \in U_i \setminus \{d_{i,1}, \ldots, d_{i,k-1}\}$ such that

(2) There is a state $q^{i,k}$ of $\mathfrak{A}$ such that if none of the parameters $b_1, \ldots, b_\ell$ is contained in V_i, the automaton $\mathfrak{A}$ is running on either $\mathcal{T}_{c_{i,k} \bar{b}}$ or $\mathcal{T}_{d_{i,k} \bar{b}}$, and it leaves $\mathrm{lca}(U_j)$ in state q_k, then it reaches $\mathrm{lca}(U_i)$ in state $q^{i,k}$

Note that some of the $c_{i,k}$s may be identical, but all the $c_{i,k}$s differ from all the $d_{i',k'}$s.

We let $Y = \{c_i : U_i \text{ has no children}\} \cup \{c_{i,k} : U_i \text{ has one child}, 1 \leq k \leq m\}$ and claim that Y satisfies the requirements of the lemma. Consider any choice of the parameters $\bar{b} = (b_1, \ldots, b_\ell)$. Then for some $i, 1 \leq i \leq \ell+1$, the set V_i contains no b_j. If U_i has no children, then $\mathfrak{A}$ accepts $\mathcal{T}_{c_i \bar{b}}$ if, and only if $\mathcal{A}$ accepts $\mathcal{T}_{d_i \bar{b}}$. If U_i has one child U_j and the automaton leaves $\mathrm{lca}(U_j)$ in state q_j then $\mathfrak{A}$ accepts $\mathcal{T}_{c_{i,j} \bar{b}}$ if, and only if, $\mathfrak{A}$ accepts $\mathcal{T}_{d_{i,j} \bar{b}}$. □

Proof of Theorem 6: Let $\ell, m, \mathfrak{A}, \mathcal{T}$ be as in the statement of the theorem and let U, p as in the statement of Lemma 7. Let Y be the subset of U provided by the lemma. Then there is no parameter setting $\bar{b}$ such that $\mathfrak{A}(\mathcal{T}, \bar{b}) \cap U = Y$, so U cannot be shattered by $\mathcal{C}(\mathfrak{A}, \mathcal{T})$. □

As an important special case we obtain that the VC-dimension of sets defined by string automata is bounded. Our proof simplifies in the string case and yields a better bound.

The following example shows that up to a constant factor, the upper bound of the theorem is optimal:

Example 8 Let $\Sigma = \{0, 1\}$. We show that for every $m \geq 1$ there is a Σ_2-tree automaton $\mathfrak{A}$ with $3m$-states and a Σ-tree $\mathcal{S}$ such that $\mathrm{VC}(\mathfrak{A}, \mathcal{S}) \geq m$. Actually, $\mathcal{S}$ is just a Σ-string. The universe of $\mathcal{S}$ is $\{1, \ldots, m \cdot (2^m + 1)\}$,

with $\preceq^{\mathcal{S}}$ being the natural ordering. The labeling $\sigma^{\mathcal{S}}$ is defined by

$$\sigma^{\mathcal{S}}(i) = \begin{cases} 0 & \text{if } 1 \leq i \leq m, \\ \varepsilon & \text{if } (j+1)m < i \leq (j+2)m \text{ and } \varepsilon \text{ is the } (i-(j+1)m)\text{th bit} \\ & \text{in the binary representation of } j. \end{cases}$$

The automaton $\mathfrak{A}$ is constructed in a such a way that for each $b = (j+1)m$ for some $j, 0 \leq j \leq 2^m - 1$ it accepts precisely those $\mathcal{S}_{ab}$ for which the ath bit of the binary representation of j is 1. We leave it as an exercise for the reader to construct the automaton.

Similarly, one can can show that there is a Σ_ℓ-tree automaton $\mathfrak{A}$ with $O(m)$ states and a string $\mathcal{S}$ such that $\mathrm{VC}(\mathfrak{A}, \mathcal{S}) \in \Omega(m \cdot \ell)$.

4.4. VC-dimension of MSO-definable families. There is a well-known correspondence between tree-automata and sentences of monadic second-order logic: A class K of trees is definable by a sentence of monadic second-order logic if, and only if, there is a tree-automaton that accepts precisely the trees in K [37]. We define the function tower : $\mathbb{N} \to \mathbb{N}$ by letting $\mathrm{tower}(0) = 1$ and, for $i \geq 1$, $\mathrm{tower}(i) = 2^{\mathrm{tower}(i-1)}$. It is known that in the worst case, the size of an automaton equivalent to an MSO-formula of length n is in $\mathrm{tower}(\Theta(n))$ [36].

We need the following straightforward extension of the equivalence between MSO-sentences and tree automata to formulas with free variables. A Σ_k-tree automaton $\mathfrak{A}$ is *equivalent* to an MSO-formula $\varphi(x_1, \ldots, x_k)$ of vocabulary $\tau(\Sigma)$ if for all Σ-trees $\mathcal{T}$ we have

$$\mathfrak{A}(\mathcal{T}) = \varphi(\mathcal{T}).$$

Lemma 9 *For every MSO-formula $\varphi(x_1, \ldots, x_k)$ of vocabulary $\tau(\Sigma)$ there is a Σ_k-tree automaton $\mathfrak{A}$ that is equivalent to φ. Furthermore, the size of the automaton $\mathfrak{A}$ is in* $\mathrm{tower}(O(n))$*, where n is the length of the formula $\varphi(x_1, \ldots, x_k)$.*

Theorem 10 *Every formula of monadic second-order logic (and thus every formula of first-order logic) has bounded VC-dimension on the class of all trees.*

Furthermore, the VC-dimension of an MSO-formula $\varphi(x, \bar{y})$ of length n is in $\mathrm{tower}(O(n))$.

Proof: Let $\varphi(x, y_1, \ldots, y_\ell)$ be an MSO-formula of vocabulary $\tau(\Sigma)$, for some finite alphabet Σ, and let n be the length of φ. Let $\mathfrak{A}$ be an automaton with $m \leq \mathrm{tower}(O(n))$ states that is equivalent to φ. Then by Theorem 6, for every Σ-tree $\mathcal{T}$ we have

$$\mathrm{VC}(\varphi, \mathcal{T}) = \mathrm{VC}(\mathfrak{A}, \mathcal{T}) < 8 \cdot m(\ell + 1) \leq \mathrm{tower}(O(n)).$$

To bound the VC-dimension of formulas $\varphi(x_1, \ldots, x_k, y_1, \ldots, y_\ell)$, we apply Lemma 4. □

We will now show that the upper bound $\mathrm{tower}(O(n))$ stated in the theorem is close to optimal, even the case of FO-formulas on strings. The basic idea of our lower bound proof is the same as the one used in Example 8, but it requires some technical machinery, which is provided by [15]. There, a family μ_h, for $h \geq 1$, of encodings of natural numbers by strings over certain finite alphabets has been introduced. They have the property that they can be decoded by shorter and shorter first-order formulas.

For all $n, i \in \mathbb{N}$ we let $\mathrm{bit}(i, n)$ denote the ith bit in the binary representation of n. (Here we count the lowest priority bit as the 0th bit.) For all $h \geq 1$ we let $\Sigma(h) = \{\texttt{0}, \texttt{1}, \texttt{<1>}, \texttt{</1>}, \ldots, \texttt{<h>}, \texttt{</h>}\}$. Note that the "tags" `<i>` and `</i>` are single symbols of the alphabet that are just used to improve readability. We define $L : \mathbb{N} \to \mathbb{N}$ by $L(0) = 0$, $L(1) = 1$, $L(n) = \lfloor \lg(n-1) \rfloor + 1$ for $n \geq 2$. Note that for $n \geq 1$, $L(n)$ is precisely the length of the binary representation of $n - 1$.

We are now ready to define the encodings $\mu_h : \mathbb{N} \to \Sigma(h)^*$, for $h \geq 1$. We let $\mu_1(0) = \texttt{<1></1>}$ and for $n \geq 1$

$$\mu_1(n) = \texttt{<1>}\,\mathrm{bit}(0, n-1)\,\mathrm{bit}(1, n-1)\,\ldots\,\mathrm{bit}(L(n)-1, n-1)\,\texttt{</1>}$$

for $n \geq 1$. For $h \geq 2$, we let $\mu_h(0) = \texttt{<h></h>}$ and

$$\mu_h(n) = \texttt{<h>}\,\mu_{h-1}(0)\,\mathrm{bit}(0, n-1)\,\mu_{h-1}(1)\,\mathrm{bit}(1, n-1)\,\ldots\,\mu_{h-1}(L(n)-1)\,\mathrm{bit}(L(n)-1, n-1)\,\texttt{</h>}.$$

The key result is the following lemma:

Lemma 11 ((Frick and Grohe [15])) *Let $h \geq 1$ and let $\Sigma \supseteq \Sigma(h)$. There is a first-order formula $\chi_h(x_1, x_2)$ of vocabulary $\tau(\Sigma(h))$ and size $O(h)$ such that for all strings $\mathcal{S} \in \Sigma^*$, $a_1, a_2 \in S$, and $n_1, n_2 \in \{0, \ldots, \text{tower}(h)\}$ the following holds:*

If a_1 is the first position of a substring $\mathcal{S}_1$ of $\mathcal{S}$ that is isomorphic to $\mu_h(n_1)$ and a_2 is the first position of a substring $\mathcal{S}_2$ of $\mathcal{S}$ that is isomorphic to $\mu_h(n_2)$, then

$$\mathcal{S} \models \chi_h(a_1, a_2) \iff n_1 = n_2.$$

Using this lemma we can easily prove our lower bound:

Theorem 12 *There is a family of MSO-formulas $\varphi_n(x, y)$ and strings $\mathcal{S}_n$, for $n \geq 1$, such that the length of φ_n is $O(n)$, and*

$$\text{VC}(\varphi_n, \mathcal{S}_n) \geq \text{tower}(n).$$

Proof: Let $n \geq 1$. We let $\mathcal{S}_n$ be the string

$$\mu_n(0)\mu_n(1)\ldots\mu_n(\text{tower}(n)-1)\#\mu_{n+1}(0)\mu_{n+1}(1)\ldots\mu_{n+1}(\text{tower}(n+1)),$$

over the alphabet $\Sigma(n+1) \cup \{\#\}$.

The formula $\varphi_n(x, y)$ says

- $\sigma(x) =$ `<n>` *and x appears before #.*
  Thus in $\mathcal{S}_n$, x is the first position of a substring of the form $\mu_n(r)$ for some $r \leq \text{tower}(n) - 1$, and this substring appears before #.
- $\sigma(y) =$ `<n+1>`
  Thus in $\mathcal{S}_n$, y is the first position of a substring $\mathcal{S}_y$ of the form $\mu_{n+1}(s)$ for some $s \leq \text{tower}(n+1)$
- *There exists an x' between y and the next closing* `</n+1>` *such that $\sigma(x') =$* `<n>`*, the bit after the next closing* `</n>` *is '1', and $\chi_n(x, x')$.*
  Here χ_n is the formula provided by Lemma 11. Thus in $\mathcal{S}_n$, x' is the first position of a substring of $\mathcal{S}_y$ of the form $\mu_n(r')$ for some $r' \leq L(\text{tower}(n+1)) - 1$, and the r'th bit of s is 1, and $r' = r$.

We can easily write this in first-order logic:

$$\begin{aligned}
\varphi_n(x, y) = \quad & P_{\texttt{<n>}}(x) \wedge \exists z(P_{\#}(z) \wedge x < z) \\
\wedge \quad & P_{\texttt{<n+1>}}(y) \\
\wedge \quad & \exists x'\Big(P_{\texttt{<n>}}(x') \wedge y < x' \wedge \forall z\big((y < z \wedge z < x) \rightarrow \neg P_{\texttt{</n+1>}}\big) \\
& \quad \wedge \exists v \exists w\big(P_{\texttt{</n>}}(v) \wedge \forall z\big((x' < z \wedge z < v) \rightarrow \neg P_{\texttt{</n>}}\big) \wedge S(v, w) \wedge P_{\texttt{1}}(w)\big) \\
& \quad \wedge \chi_n(x, x')\Big).
\end{aligned}$$

Recall that S is the successor relation in a string.

Since $L(\text{tower}(n+1)) = \text{tower}(n)$, the string $\mu_{n+1}(\text{tower}(n+1))$ contains precisely $\text{tower}(n)$ substrings $\mu_n(r)$ whose first positions may serve as x'. Thus $\mathcal{C}(\varphi_n(x, y), \mathcal{S}_n)$ shatters the set of all first positions of subswords $\mu_n(r)$ appearing before #. This set has size $\text{tower}(n)$, thus $\text{VC}(\varphi_n(x, y), \mathcal{S}_n) \geq \text{tower}(n)$. □

Remark 13 The alphabet of the formulas φ_n and strings $\mathcal{S}_n$ in Theorem 12 depends on n; its size is $O(n)$. We do not know if the statement remains true for a fixed alphabet. Of course we can simply encode the symbols in $\Sigma(n)$ by words over $\{0, 1\}$ and translate the formula φ_n and the string $\mathcal{S}_n$ to the alphabet $\{0, 1\}$, but this would increase the formula size by a logarithmic factor.

Remark 14 Theorem 10 is closely related to recent results of Benedikt, Libkin, Schwentick, and Segoufin [7] on the VC-dimension of first-order formulas and monadic second-order formulas on infinite trees. (Note that our results speak about finite trees.) Their setting and methods are quite different; they study algebras of strings, which can be considered as infinite trees, and then use model-theoretic techniques to prove bounded VC-dimension.

It seems that the fact that the VC-dimension of MSO-formulas on trees is bounded can also be derived by their techniques.

5. Tree-like structures

There are different ways of defining classes of structures that are similar to trees. The best-known notion measuring the similarity of a graph to a tree is *tree-width* [31]. It is well-known that structures of *bounded tree-width* inherit many of the nice properties of trees; we shall see that bounded VC-dimension of MSO-definable families of sets is among them.

However, instead of tree-width we shall measure the similarity of structures to trees by their *clique-width* [9]. It is well-known that all classes of structures of bounded tree-width have bounded clique-width. There are natural classes of structures of bounded clique-width that have unbounded tree-width, maybe the simplest example is the class of all linear orders. Another example is the class of trees. If the partial order $\preceq$ is present, then trees do not have bounded tree-width, but it is easy to see that they do have bounded clique width.

A *k-colored τ-structure* is a pair $(\mathcal{A}, \gamma)$ consisting of a τ-structure $\mathcal{A}$ and a mapping $\gamma : A \to \{1, \dots, k\}$. A *basic k-colored τ-structure* is a k-colored τ-structure $(\mathcal{A}, \gamma)$ where $|A| = 1$ and $R^{\mathcal{A}} = \emptyset$ for all $R \in \tau$.

We let $\Gamma_k[\tau]$ be the smallest class of k-colored τ-structures that contains all basic k-colored τ-structures and is closed under the following operations:

- *Union:* Take two k-colored τ-structures on disjoint vertex sets and form their union.
- *$(i \to j)$-recoloring, for $1 \leq i, j \leq k$:* Take a k-colored τ-structure and recolor all vertices colored i to j.
- *$(R, i_1, \dots, i_r)$-connecting, for every $r \geq 1$, every r-ary $R \in \tau$ and $1 \leq i_1, \dots, i_r \leq k$:* Take a k-colored τ-structure $(\mathcal{A}, \gamma)$ and add all tuples $(a_1, \dots, a_r) \in A^r$ with $\gamma(a_j) = i_j$ for $1 \leq j \leq r$ to $\mathcal{R}^{\mathcal{A}}$.

Definition 15 The *clique-width* of a τ-structure $\mathcal{A}$, denoted by $\mathrm{cw}(\mathcal{A})$, is the minimum k such that there exists a k-coloring $\gamma : A \to \{1, \dots, k\}$ such that $(\mathcal{A}, \gamma) \in \Gamma_k[\tau]$.

For every k-colored structure $(\mathcal{A}, \gamma) \in \Gamma_k[\tau]$ we can define a binary, labeled *parse-tree* in a straightforward way. The leaves of this tree are the elements of A labeled by their color, and each inner node is labeled by the operation it corresponds to. A *parse-tree* of a structure $\mathcal{A}$ of clique-width k is a parse tree of some $(\mathcal{A}, \gamma) \in \Gamma_k[\tau]$. For the next lemma, it is important to note that if $\mathcal{T}$ is a parse-tree for a structure $\mathcal{A}$, then $A \subseteq T$.

Lemma 16 *Let $k \geq 1$. For every MSO-formula $\varphi(\bar{x})$ there is a formula $\tilde{\varphi}(\bar{x})$ such that for every structure $\mathcal{A}$ of clique-width k and for every parse-tree $\mathcal{T}$ of $\mathcal{A}$ we have $\varphi(\mathcal{A}) = \tilde{\varphi}(\mathcal{T})$.*

Furthermore, there are constants c, d (only depending on k and the vocabulary) such that $||\tilde{\varphi}|| \leq c||\varphi||$ and $\mathrm{qr}(\tilde{\varphi}) \leq \mathrm{qr}(\varphi) + d$.

Proof: The proof is a straightforward induction, the only non-trivial case being $\varphi(\bar{x}) = Rx_1 \dots x_r$ for some r-ary relation symbol R. For a vertex $t \in T$, we let $\mathcal{T}_t$ denote the subtree of $\mathcal{T}$ with root t. Note that for each $t \in T$ there exists a substructure $\mathcal{A}_t \subseteq \mathcal{A}$ and a labeling $\gamma_t : A_t \to \{1, \dots, k\}$ such that $\mathcal{T}_t$ is a parse-tree for $(\mathcal{A}_t, \gamma_t)$. For every tuple $\bar{a} = (a_1, \dots, a_r) \in A^r$ we have $\bar{a} \in R^{\mathcal{A}}$ if, and only if, there exists a node t of $\mathcal{T}$ and indices $i_1, \dots, i_r \in \{1, \dots, k\}$ such that t is labeled by the operation "$(R, i_1, \dots, i_r)$-connecting", and $\gamma_t(a_j) = i_j$ for $1 \leq j \leq r$.

We claim that for all $i \in \{1, \dots, k\}$ there is an MSO-formula $\psi_i(x, y)$ such that for all $a \in A$, $t \in T$ we have:

$$\mathcal{T} \models \psi_i(a, t) \iff a \in A_t \text{ and } \gamma_t(a) = i.$$

To see this, consider the path from t to a as a string and observe that the language of all such strings with $\gamma_t(a) = i$ is regular. Indeed, a finite automaton moving along this path can keep track of all the relabelings, and thus it can compute the label of a at t. Thus by Büchi's Theorem, the language is MSO-definable.

Let $P_{R,i_1,\dots,i_r}$ be the predicate symbol indicating that a node t is labeled by the operation

$$\text{"}(R, i_1, \dots, i_r)\text{-connecting"},$$

and let $\zeta(x)$ be a formula saying that x is a leaf of the tree. We let

$$\tilde{\varphi}(x_1, \dots, x_r) = \exists y \big(P_{R,i_1,\dots,i_r} y \wedge \bigwedge_{j=1}^{r} (\zeta(x_i) \wedge \psi_{i_j}(x_i, y)) \big).$$

Then, recalling that $\varphi(\bar{x}) = Rx_1 \dots x_r$, for all $a_1, \dots, a_r \in A$ we have

$$\mathcal{A} \models \varphi(a_1, \dots, a_r) \iff \mathcal{T} \models \tilde{\varphi}(a_1, \dots, a_r).$$

□

Theorem 17 *Let $w \geq 1$. Then every formula of monadic second order logic has bounded VC-dimension on the class of all structures of clique-width at most w.*

Proof: This follows immediately from Theorem 10 and Lemma 16. □

Note that the proof of this result does not involve any large constants, so the bounds on the VC-dimension we obtain are essentially the same as those of Theorem 10.

As we have mentioned before, the clique-width of a structure is bounded in terms of its *tree-width*; more precisely, a structure of tree-width at most k has clique-width at most 2^k [10].

Corollary 18 *Let $w \geq 1$. Then every formula of monadic second-order logic has bounded VC-dimension on the class of all structures of tree-width width at most w.*

We now show that in some weak sense, our previous results for the VC-dimension of MSO-formulas are optimal. As a first step, in the following example we observe that MSO-formulas have unbounded VC-dimension on *grids*.

Example 19 Let $n, m \geq 0$. The $(n \times m)$-*grid* is the graph $\mathcal{G}_{n \times m}$ with vertex set $\{0, \dots, m-1\} \times \{0, \dots, n-1\}$ and an edge between (i, j) and (i', j') if, and only if, either $i = i'$ and $|j - j'| = 1$ or $j = j'$ and $|i - i'| = 1$. We think of the vertices of a grid as being numbered as a matrix, i.e., $(0, 0)$ is the upper left corner, and (i, j) is the vertex in the ith row and jth column.

It is not hard to see that there is an MSO-formula $\varphi(x, y)$ such that for all $n \geq 1$,

$$\mathrm{VC}(\mathcal{C}(\varphi, \mathcal{G}_{n \times n})) \geq \log(n).$$

Let us sketch a proof of this result: We show that there is an MSO-formula $\psi(x, y_1, y_2)$ such that for $1 \leq i \leq n$ we have

$$(0, j) \in \varphi(\mathcal{G}_{n \times n}, (0, 0), (i, 0)) \iff \text{the } j\text{th bit in the binary representation of } i \text{ is } 1.$$

This shows that $\mathcal{C}(\psi, \mathcal{G}_{n \times n})$ shatters the set $\{(0, j) \mid 0 \leq j < \log(n)\}$. It is easy to get rid of the additional parameter on $(0, 0)$, which is only used to fix our coordinate system (together with the second parameter).

ψ says that:

- There exists a set X such that for $0 \leq p, q \leq n$ we have $(p, q) \in X$ if, and only if, the qth bit in the binary representation of p is 1. To express this in MSO, we say that for $1 \leq p \leq n - 2$, the $(p + 1)$st row is one plus the pth row if we read the rows as binary numbers with elements of X being ones, starting with the least significant bit.
- There is a path from y_2 to x that goes horizontally to the right from y_2 to an element of X, then vertically up to x.

It is easy to formalize this in MSO.

The *excluded grid theorem* due to Robertson and Seymour [32] says that a class K of graphs has bounded tree-width if, and only if, there is an $n \geq 1$ such that $\mathcal{G}_{n \times n}$ is not a minor of any graph in K.

Corollary 20 *Let K be a class of graphs that is closed under taking subgraphs. Then every MSO-formula has bounded VC-dimension on K if, and only if, K has bounded tree-width.*

Proof: For classes K that are closed under taking minors, the statement of the corollary follows immediately from Corollary 18, Example 19, and the excluded grid theorem.

To obtain the stronger statement for classes that are merely closed under taking subgraphs, we work with *walls* (or *hexagonal grids*) instead of the grids of Example 19 and use a variant of the excluded grid theorem stating that a class K of graphs has bounded tree-width if, and only if, there is an $n \geq 1$ such that no subdivision of a wall of height and width n is a subgraph of any graph in K [32]. Then we note that Example 19 can easily be extended to subdivisions of walls. □

Note that Corollary 20 does not contradict Theorem 17, because the class of all graphs of clique-width at most k is not closed under taking subgraphs for any $k \geq 2$. Bounded clique-width is *not* a necessary condition for bounded VC-dimension of MSO-formulas on arbitrary classes of graphs, as the following example shows.

Example 21 Let $\mathcal{H}_n$ be the graph obtained from the complete graph $\mathcal{K}_n$ by subdividing each edge by a new vertex. Then it follows directly from symmetry considerations that every MSO-formula has bounded VC-dimension on this class. On the other hand, the clique-width of $\mathcal{H}_n$ is $\Omega(\sqrt{n})$, as $\mathcal{H}_n$ contains an $m \times m$ grid with $m = \Omega(\sqrt{n})$ as an induced subgraph, and the class of graphs of clique-width w is closed under taking induced subgraphs [10].

6. Locally tree-like structures

While Example 19 and Corollary 20 indicate that we cannot extend the range of structures where formulas of monadic second-order logic have bounded VC-dimension much further, we shall show in this section that formulas of *first-order logic* have bounded VC-dimension on many other interesting classes of structures, most notably the class of planar graphs and classes of graphs of bounded degree.

Our main technical tool is the *locality* of first-order logic. We need some new notation: The *Gaifman graph* of a τ-structure $\mathcal{A}$ is the graph $\mathcal{G}_\mathcal{A}$ with vertex set $G_\mathcal{A} = A$ and an edge between two distinct vertices $a, b \in A$ if there exists an $R \in \tau$ and a tuple $(a_1, \ldots, a_k) \in R^\mathcal{A}$ such that $a, b \in \{a_1, \ldots, a_k\}$. The *distance* $d^\mathcal{A}(a, b)$ between two elements $a, b \in A$ of a structure $\mathcal{A}$ is the length of the shortest path in $\mathcal{G}_\mathcal{A}$ connecting a and b. The distance between two tuples $\bar{a} = (a_1, \ldots, a_k) \in A^k, \bar{b} = (b_1, \ldots, b_l) \in A^l$ of elements of a structure $\mathcal{A}$ is the minimum distance between their elements, i.e., the number $d^\mathcal{A}(\bar{a}, \bar{b}) = \min\{d^\mathcal{A}(a_i, b_j) \mid 1 \leq i \leq k, 1 \leq j \leq l\}$. For $r \geq 1$ and $a \in A$ we define the *r-neighborhood* of a in $\mathcal{A}$ to be $N_r^\mathcal{A}(a) = \{b \in A \mid d^\mathcal{A}(a, b) \leq r\}$. For a tuple $\bar{a} = (a_1, \ldots, a_k) \in A^k$ we let $N_r^\mathcal{A}(\bar{a}) = \bigcup_{i=1}^k N_r^\mathcal{A}(a_i)$. By $\mathcal{N}_r^\mathcal{A}(\bar{a})$ we denote the induced substructure of $\mathcal{A}$ with universe $N_r^\mathcal{A}(\bar{a})$.

One of the features that distinguish first-order logic from second-order logic is the *locality* of first-order logic, as it is described in *Gaifman's locality theorem* [16]. The following lemma, which is the main technical result of this section, states that bounded VC-dimension of first-order formulas is a local property. For an $r \geq 1$ and a class K of structures, we let $N(r, K) = \{\mathcal{N}_r^\mathcal{A}(a) \mid \mathcal{A} \in K, a \in A\}$.

Lemma 22 *Let K be a class of structures such that for every $r \geq 1$, every first-order formula has bounded VC-dimension on the class $N(r, K)$. Then every first-order formula has bounded VC-dimension on K.*

The proof of this lemma requires some additional terminology and a few model-theoretic facts. We fix a vocabulary τ. Let $q, k \geq 0$. Let $\mathcal{A}$ be a τ-structure and $\bar{a} \in A^k$. The *(q, k)-type of $\bar{a}$ in $\mathcal{A}$*, denoted by $\mathrm{tp}_q^\mathcal{A}(\bar{a})$, is the set of all first-order formulas $\varphi(x_1, \ldots, x_k)$ of quantifier rank at most q such that $\mathcal{A} \models \varphi(\bar{a})$. A *$(q, k)$-type* is a maximal consistent set of first-order formulas $\varphi(x_1, \ldots, x_k)$ of quantifier rank at most q. Equivalently, a (q, k)-type is the (q, k)-type of some k-tuple $\bar{a}$ in some structure $\mathcal{A}$. For all q, k there are only finitely many (q, k)-types; we denote the number of (q, k)-types by $t(q, k)$.

Since up to logical equivalence there are only finitely many first-order formulas $\varphi(x_1, \ldots, x_k)$ of quantifier rank at most q, for every (q, k)-type Θ there is a first-order formula $\theta(\bar{x})$ of quantifier rank q such that for all structures $\mathcal{A}$ and tuples $\bar{a} \in A^k$ we have $\mathrm{tp}_q^\mathcal{A}(\bar{a}) = \Theta \iff \mathcal{A} \models \theta(\bar{a})$. We say that *$\theta$ isolates* the type Θ.

The *union* of two τ-structures $\mathcal{A}, \mathcal{B}$ is the structure $\mathcal{A} \cup \mathcal{B}$ with universe $A \cup B$ and relations $R^{\mathcal{A}\cup\mathcal{B}} = R^\mathcal{A} \cup R^\mathcal{B}$ for $R \in \tau$. The following lemma is a simple consequence of the well-known Feferman-Vaught theorem; it can easily be proved directly using the Ehrenfeucht-Fraïssé game for first-order logic (see e.g. [12]).

Lemma 23 *Let $k, l, q \geq 0$, and let $\mathcal{A}, \mathcal{B}, \mathcal{A}', \mathcal{B}'$ be structures with $A \cap B = \emptyset$, $A' \cap B' = \emptyset$ and $\bar{a} \in A^k, \bar{a}' \in (A')^k, \bar{b} \in B^l, \bar{b}' \in (B')^l$ such that $\mathrm{tp}_q^{\mathcal{A}}(\bar{a}) = \mathrm{tp}_q^{\mathcal{A}'}(\bar{a}')$, $\mathrm{tp}_q^{\mathcal{B}}(\bar{b}) = \mathrm{tp}_q^{\mathcal{B}'}(\bar{b}')$.*

Then

$$\mathrm{tp}_q^{\mathcal{A}\cup\mathcal{B}}(\bar{a}\bar{b}) = \mathrm{tp}_q^{\mathcal{A}'\cup\mathcal{B}'}(\bar{a}'\bar{b}').$$

We exploit the locality of first-order logic using the following lemma. Let us remark that for a larger value of r, it is an immediate consequence of Gaifman's locality theorem [16]:

Lemma 24 (Libkin [25]) *Let $q \geq 0$ and $r = 2^q - 1$. Then for all structures $\mathcal{A}$ and all k-tuples $\bar{a}, \bar{b} \in A^k$ we have*

$$\mathrm{tp}_q^{\mathcal{A}}(\bar{a}) = \mathrm{tp}_q^{\mathcal{A}}(\bar{b}) \iff \mathrm{tp}_q^{\mathcal{N}_r^{\mathcal{A}}(\bar{a})}(\bar{a}) = \mathrm{tp}_q^{\mathcal{N}_r^{\mathcal{A}}(\bar{b})}(\bar{b})$$

Combining the previous two lemmas, we obtain the following:

Corollary 25 *Let $q \geq 0$ and $r = 2^q - 1$. Let $\mathcal{A}$ be a structure and $\bar{a}, \bar{a}' \in A^k, \bar{b}, \bar{b}' \in A^l$ such that $\mathrm{tp}_q^{\mathcal{A}}(\bar{a}) = \mathrm{tp}_q^{\mathcal{A}}(\bar{a}')$, $\mathrm{tp}_q^{\mathcal{A}}(\bar{b}) = \mathrm{tp}_q^{\mathcal{A}}(\bar{b}')$ and $d^{\mathcal{A}}(\bar{a}, \bar{b}) > 2r + 1$, $d^{\mathcal{A}}(\bar{a}', \bar{b}') > 2r + 1$.*

Then $\mathrm{tp}_q^{\mathcal{A}}(\bar{a}\bar{b}) = \mathrm{tp}_q^{\mathcal{A}}(\bar{a}'\bar{b}')$.

Proof (of Lemma 22): We only prove that every first-order formula $\varphi(x, \bar{y})$ has bounded VC-dimension on K; the extension to formulas $\varphi(\bar{x}, \bar{y})$ then follows from Lemma 4.

So let $\varphi(x, y_1, \ldots, y_\ell)$ be a first-order formula of quantifier rank q and $r = 2^q - 1$. Let $\mathcal{A} \in K$, $\mathcal{C} = \mathcal{C}(\varphi, \mathcal{A})$, and $X \subseteq A$ a set that is shattered by $\mathcal{C}$.

Step 1: We prove that X has no subset Y of size $(2\ell + 1) \cdot t(q, 1) + 1$ such that for all $a, b \in Y$ we have $d^{\mathcal{A}}(a, b) > 4r + 2$.

To see this, suppose that Y is such a set. Then there are pairwise distinct $a_1, \ldots, a_{2\ell+2} \in Y$ such that $\mathrm{tp}_q^{\mathcal{A}}(a_i) = \mathrm{tp}_q^{\mathcal{A}}(a_j)$ for $1 \leq i, j \leq 2\ell + 2$. We claim that there is no choice of parameters $b_1, \ldots, b_\ell \in A$ such that

$$\varphi(\mathcal{A}, b_1, \ldots, b_\ell) \cap Y = \{a_1, \ldots, a_{\ell+1}\}.$$

If this claim is correct, then Y and therefore X is not shattered by $\mathcal{C}(\varphi, \mathcal{A})$, which is a contradiction.

So we have to prove the claim. Let $\bar{b} = (b_1, \ldots, b_\ell) \in A^l$. Since $d^{\mathcal{A}}(a_i, a_j) > 4r + 2$, for each b_i there exists at most one a_j such that $d^{\mathcal{A}}(b_i, a_j) \leq 2r + 1$. Thus there are at least one $j \in \{1, \ldots, \ell + 1\}$ and one $j' \in \{\ell + 2, \ldots, 2\ell + 2\}$ such that $d^{\mathcal{A}}(\bar{b}, a_j) > 2r + 1$ and $d^{\mathcal{A}}(\bar{b}, a_{j'}) > 2r + 1$. Since $\mathrm{tp}_q^{\mathcal{A}}(a_j) = \mathrm{tp}_q^{\mathcal{A}}(a_{j'})$, by Corollary 25 we have

$$\mathrm{tp}_q^{\mathcal{A}}(a_j\bar{b}) = \mathrm{tp}_q^{\mathcal{A}}(a_{j'}\bar{b}).$$

But then $a_j \in \varphi(\mathcal{A}, \bar{b})$ if, and only if, $a_{j'} \in \varphi(\mathcal{A}, \bar{b})$. This proves that $\varphi(\mathcal{A}, b_1, \ldots, b_\ell) \cap Y \neq \{a_1, \ldots, a_{\ell+1}\}$ and thus completes Step 1.

Step 2: We prove that there is a number $s(\ell, q)$ such that X has no subset Y of size $s(\ell, q)$ for which there exists an $a_0 \in A$ such that $Y \subseteq N_{4r+2}^{\mathcal{A}}(a_0)$.

To see this suppose that $a_0 \in A$ and $Y \subseteq X$ such that $Y \subseteq N_{4r+2}^{\mathcal{A}}(a_0)$. Since X is shattered by $\mathcal{C}$, for every $Z \subseteq Y$ there is a tuple $\bar{b}^Z = (b_1^Z, \ldots, b_\ell^Z) \in A^\ell$ such that $\varphi(\mathcal{A}, \bar{b}^Z) \cap Y = Z$. Consider the $\ell + 1$ disjoint sets

$$N_{2(2r+1)}^{\mathcal{A}}(a_0), N_{3(2r+1)}^{\mathcal{A}}(a_0) \setminus N_{2(2r+1)}^{\mathcal{A}}(a_0), \ldots, N_{(2+\ell)(2r+1)}^{\mathcal{A}}(a_0) \setminus N_{(2+\ell-1)(2r+1)}^{\mathcal{A}}(a_0).$$

One of these sets will contain no b_i^Z for $1 \leq i \leq \ell$. Thus there exists a set $I_Z \subseteq \{1, \ldots, \ell\}$ and a $p_Z, 0 \leq p_Z \leq \ell$ such that $b_i^Z \in N_{(2+p_Z)(2r+1)}(a_0)$ for all $i \in I_Z$ and $b_i^Z \notin N_{(2+p_Z+1)(2r+1)}(a_0)$ for all $i \in \{1, \ldots, \ell\} \setminus I_Z$. We let $\bar{c}^Z$ be the subtuple of $\bar{b}^Z$ consisting of all b_i^Z with $i \in I_Z$ and $\bar{d}^Z$ the subtuple of $\bar{b}^Z$ consisting of the remaining b_i^Z (it may happen that either $\bar{c}^Z$ or $\bar{d}^Z$ is the empty tuple). Note that if both tuples are non-empty, we have $d^{\mathcal{A}}(\bar{c}^Z, \bar{d}^Z) > 2r + 1$. Moreover, if $\bar{d}^Z$ is non-empty, we have $d^{\mathcal{A}}(a_0, \bar{d}^Z) > 6r + 3$ and thus for all $a \in Y$, $d^{\mathcal{A}}(a, \bar{d}^Z) > 2r + 1$.

Let

$$T = \{(I_Z, p_Z, \mathrm{tp}_q^{\mathcal{A}}(\bar{d}^Z)) \mid Z \subseteq Y\}$$

and $t = 2^\ell \cdot (\ell+1) \cdot \sum_{k=0}^{\ell} t(q,k)$. Recall that $t(q,k)$ denote the number of (q,k)-types. Thus $|T| \leq t$, so there exists an $I \subseteq \{1,\dots,\ell\}$, a $p, 0 \leq p \leq \ell$, a $(q, \ell - |I|)$-type Θ, and a subset $\mathcal{Z} \subseteq 2^Y$ of size at least $2^{|Y|}/t$ such that for all $Z \in \mathcal{Z}$ we have $I_Z = I$, $p_Z = p$, and $\mathrm{tp}_q^{\mathcal{A}}(\bar{d}^Z) = \Theta$. Without loss of generality we assume that $I = \{1,\dots,k\}$ for some $k \leq \ell$.

For all $Z \in \mathcal{Z}$ and $a \in Z$ we let $\theta_{Z,a}(x, y_1, \dots, y_k)$ be a formula isolating the type $\mathrm{tp}_q^{\mathcal{A}}(a, \bar{c}^Z)$. Moreover, we let

$$\psi(x, y_1, \dots, y_k) = \bigvee_{\substack{Z \in \mathcal{Z} \\ a \in Z}} \theta_{Z,a}(x, y_1, \dots, y_k).$$

Claim: For all $Z \in \mathcal{Z}$ we have $\psi(\mathcal{A}, \bar{c}^Z) \cap Y = Z$.

Let $Z \in \mathcal{Z}$. To see that $Z \subseteq \psi(\mathcal{A}, \bar{c}^Z) \cap Y$, note that by the definition of $\theta_{Z,a}(x, \bar{y})$, for all $a \in Z$ we have $\mathcal{A} \models \theta_{Z,a}(a, \bar{c}^Z)$ and thus $\mathcal{A} \models \psi(a, \bar{c}^Z)$.

To prove the converse inclusion, let $a \in \psi(\mathcal{A}, \bar{c}^Z) \cap Y$. Let $Z' \in \mathcal{Z}$ and $a' \in Z'$ such that $a \in \theta_{Z',a'}(\mathcal{A}, \bar{c}^Z)$. Thus

$$\mathrm{tp}_q^{\mathcal{A}}(a\bar{c}^Z) = \mathrm{tp}_q^{\mathcal{A}}(a'\bar{c}^{Z'}). \tag{$*$}$$

Recall that

- $\bar{b}^{Z'} = \bar{c}^{Z'}\bar{d}^{Z'}$ and $d^{\mathcal{A}}(\bar{c}^{Z'}, \bar{d}^{Z'}) > 2r+1$,
- $\bar{b}^{Z} = \bar{c}^{Z}\bar{d}^{Z}$ and $d^{\mathcal{A}}(\bar{c}^{Z}, \bar{d}^{Z}) > 2r+1$,
- $\mathrm{tp}_q^{\mathcal{A}}(\bar{d}^{Z'}) = \mathrm{tp}_q^{\mathcal{A}}(\bar{d}^{Z}) = \Theta$,
- for all $a'' \in Y$, $d^{\mathcal{A}}(a'', \bar{d}^{Z'}) > 2r+1$ and $d^{\mathcal{A}}(a'', \bar{d}^{Z}) > 2r+1$.

By Corollary 25 and $(*)$, this implies that

$$\mathrm{tp}_q^{\mathcal{A}}(a\bar{b}^Z) = \mathrm{tp}_q^{\mathcal{A}}(a'\bar{b}^{Z'}).$$

Thus $a \in \varphi(\mathcal{A}, \bar{b}^Z)$ if, and only if, $a' \in \varphi(\mathcal{A}, \bar{b}^{Z'})$. Since $a' \in Z' = \varphi(\mathcal{A}, \bar{b}^{Z'}) \cap Y$, we have $a \in \varphi(\mathcal{A}, \bar{b}^Z)$ and thus, recalling that $a \in Y$,

$$a \in Z = \varphi(\mathcal{A}, \bar{b}^Z) \cap Y.$$

This proves the claim.

It is an immediate consequence of the claim that for $W, Z \in \mathcal{Z}$ such that $W \neq Z$ we have

$$\psi(\mathcal{A}, \bar{c}^Z) \cap Y \neq \psi(\mathcal{A}, \bar{c}^W) \cap Y. \tag{$**$}$$

Let $\mathcal{N} = \mathcal{N}^{\mathcal{A}}_{(2+p)(2r+1)+r}(a_0)$. By Lemma 24, $(**)$ implies

$$\psi(\mathcal{N}, \bar{c}^Z) \cap Y \neq \psi(\mathcal{N}, \bar{c}^W) \cap Y,$$

because $N_r^{\mathcal{A}}(a\bar{c}^Z) \subseteq N^{\mathcal{A}}_{(2+p)(2r+1)+r}(a_0)$ for every $a \in Y$. Since $|\mathcal{Z}| \geq 2^{|Y|}/t$, this implies

$$\mathcal{C}(\psi, \mathcal{N}) \cap Y \geq \frac{2^{|Y|}}{t}.$$

Let d be an upper bound for the VC-dimension of ψ on $N((2+\ell)(2r+1)+r, K)$; we can find such a bound only depending on q and ℓ. Note that $\mathcal{N} \in N((2+\ell)(2r+1)+r, K)$. Thus by Lemma 2, we have

$$\frac{2^{|Y|}}{t} \leq c \cdot |Y|^d,$$

for some constant c. This puts a bound $s(\ell, q)$ on the size of $|Y|$ and completes Step 2.

Steps 1 and 2 together imply that

$$|X| \leq 2\ell \cdot t(q,1) \cdot s(\ell, q).$$

This proves Lemma 22. □

Remark 26 It is worthwhile noting that the class K of structures in Lemma 22 may contain finite as well as infinite structures.

Combined with the results of the previous section, this lemma shows that first-order formulas have bounded VC-dimension on a number of interesting classes of structures. We say that a class K of structures has *bounded local clique-width*, if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every $r \geq 1$, $\mathcal{A} \in C$, and $a \in A$ we have $\mathrm{cw}(\mathcal{N}_r^{\mathcal{A}}(a)) \leq f(r)$.

Theorem 27 *Let τ be a vocabulary and K be a class of τ-structures of bounded local clique-width. Then every first-order formula has bounded VC-dimension on K.*

Proof: If K has bounded local clique-width, then by Theorem 17, for every $r \geq 0$ every first-order formula has bounded VC-dimension on the class $N(r, K)$. The theorem follows from Lemma 22. □

Surprisingly many natural classes of structures have bounded local clique-width, among them the class of all planar graphs, and more generally all classes of graphs of bounded genus, and all classes of graphs of bounded degree. As a matter of fact, all these classes have *bounded local tree-width* [13, 18].

Corollary 28 *Let K be a class of graphs of bounded genus or bounded degree. Then every first-order formula φ has bounded VC-dimension on K.*

As a matter of fact, the corollary also follows from known results in model-theory. Let us a call a graph $\mathcal{K}_n$*-free*, if it does not contain a subdivision of the complete graph on n-vertices as a subgraph. We call a class K of graphs $\mathcal{K}_n$-free if every graph in K is $\mathcal{K}_n$-free. It is easy to see that for every class K of graphs of bounded genus or bounded degree there exists an n such that K is $\mathcal{K}_n$-free.

Podewski and Ziegler [30] proved that an infinite graph that is $\mathcal{K}_n$-free for some $n \geq 1$ has a *stable* theory. It is known that if a structure $\mathcal{A}$ has a stable theory, then $\mathrm{VC}(\mathcal{C}(\varphi, \mathcal{A})) < \infty$ for every first-order formula φ (in model-theoretic terminology, a stable structure does not have the *independence property*). It is easy to see that if a structure $\mathcal{A}$ is the disjoint union of all structures of a class K of finite structures, and $\mathrm{VC}(\mathcal{C}(\varphi, \mathcal{A})) < \infty$ for every first-order formula φ, then every first-order formula φ has bounded VC-dimension on K. Thus we obtain:

Theorem 29 (Podewski and Ziegler [30]) *Let $n \geq 1$ and K be a $\mathcal{K}_n$-free class of graphs. Then every first-order formula φ has bounded VC-dimension on K.*

In particular, this gives another proof of Corollary 28. Note, however, that unions of classes of finite structures of bounded local clique-width or bounded clique-width are not stable in general. For example, the clique-width of a linear order is just 2.

We close this section with an example of a natural class of structures that has bounded local clique-width, but neither bounded clique-width nor bounded local tree-width:

Example 30 Let $k \geq 0$ and $\tau = \{E_1, E_2, \equiv, P_1, \ldots, P_k\}$, where E_1, E_2, and $\equiv$ are binary and $P_1, \ldots, P_k$ are unary. We let K be the class of all τ-structures $\mathcal{T}$ whose restriction to $\tau \setminus \{\equiv\}$ is a labeled binary tree without the tree-order $\preceq$, and on which $\equiv^{\mathcal{T}}$ is the "equal-height" relation. Thus for $a, b \in T$ we have $a \equiv^{\mathcal{T}} b$ if the paths from a and b to the root of $\mathcal{T}$ have the same length.

To see that the class K has bounded local clique-width, we first observe that for each h the class of all h-labeled (remember the definition of clique-width) forests of height at most h, where all vertices of the same height have the same label, has clique-width at most h. Since neighborhoods of radius r in trees $\mathcal{T} \in K$ are essentially forests of height $2r + 1$ with an "equal-height" relation, it follows that such neighborhoods have clique-width at most $2r + 1$.

To see that the class K does not have bounded local tree-width, we note that the Gaifman graphs of trees $\mathcal{T} \in K$ may contain arbitrarily large cliques, and this is impossible in a class of structures of bounded local tree-width.

Finally, we claim that MSO-formulas may have unbounded VC-dimension on K; by Theorem 17 this implies that K has unbounded clique-width. To prove the claim, consider $\mathcal{T} \in K$ such that the underlying tree of $\mathcal{T}$ is the complete binary tree of height h. We let $X \subseteq T$ be the set of vertices of the leftmost path in $\mathcal{T}$ from the root to a leaf, except the root itself. Then X is shattered by an MSO-formula $\varphi(x, y)$ which says:

There exists a z on the path from the root to y such that x has the same height as z and z is a right child of its parent.

7. Strong Consistency Dimension

Let V be a set, and let $\mathcal{C} \subseteq \mathcal{H} \subseteq 2^V$ be two families of subsets of V. A *partially specified subset* U of V is a mapping $U : V \to \{0, 1, *\}$, where $v \in U$ if $U(v) = 1$, $v \notin U$ if $U(v) = 0$ (in these cases we say that the membership of v in U is *specified*), and the membership of v in U is unspecified otherwise. The *size* of U is the number of elements whose membership in U is specified. A partially specified subset U' is a *restriction* of a partially specified subset U if $U'(v) = U(v)$ for every v such that $U'(v) \in \{0, 1\}$. In this case we also say that U is an *extension* of U'.

The *strong consistency dimension* [6] of $\mathcal{C}$ with respect to $\mathcal{H}$, denoted by $\mathrm{SC}(\mathcal{C}, \mathcal{H})$, is the smallest number d for which the following holds:

> For every partially specified subset U of V, if every size d restriction U' of U has an extension in $\mathcal{C}$, then U has an extension in $\mathcal{H}$.

As we consider only finite sets V, the strong consistency dimension is defined, as $|V|$ is a possible value for d.

Strong consistency dimension turns out to be relevant for *learning with equivalence queries*. In order to present the learnability implications of our result, we give a brief description of this model. The families of sets $\mathcal{C}$ and $\mathcal{H}$ above are called the *concept class*, respectively the *hypothesis class*. The learner has to identify an unknown *target concept* $C \in \mathcal{C}$ by asking *equivalence queries* from $\mathcal{H}$. An equivalence query is a hypothesis $H \in \mathcal{H}$. If $C = H$ then the answer to the query is 'yes', and the learning process terminates. Otherwise, the answer is a *counterexample*, i.e., an element x from $C \oplus H$. The complexity $\mathrm{EQ}(\mathcal{C}, \mathcal{H})$ of learning $\mathcal{C}$ with equivalence queries from $\mathcal{H}$ is the minimal worst-case number of queries asked by any learning algorithm identifying the target, for every choice of the target and the counterexamples.

Learning algorithms using equivalence queries can be turned into PAC learning algorithms that produce hypotheses from $\mathcal{H}$, by replacing every equivalence query with several random examples [2]. If a counterexample is found, the simulation of the query learning algorithm can continue. Otherwise, the final equivalence query is an approximately correct hypothesis, with high probability.

Theorem 31 (Balcázar, Castro, Guijarro, Simon [6])

$$\mathrm{SC}(\mathcal{C}, \mathcal{H}) \leq \mathrm{EQ}(\mathcal{C}, \mathcal{H}) \leq \lceil \mathrm{SC}(\mathcal{C}, \mathcal{H}) \cdot \ln|\mathcal{C}| \rceil + 1.$$

Let $m, \ell \geq 1$ and $\mathcal{T}$ be a Σ-tree. Then $\mathcal{AUT}(m, \ell, \mathcal{T})$ is the class of all subsets of $\mathcal{T}$ definable by m-state $\Sigma_{\ell+1}$-tree automata in $\mathcal{T}$, or more formally,

$$\mathcal{AUT}(m, \ell, \mathcal{T}) = \bigcup_{\mathfrak{A}} \mathcal{C}(\mathfrak{A}, \mathcal{T}),$$

where the union is over all m-state $\Sigma_{\ell+1}$-tree automata $\mathfrak{A}$.

Theorem 32 *Let Σ be a finite alphabet. For every m and ℓ there is an M such that for every Σ-tree $\mathcal{T}$ it holds that*

$$\mathrm{SC}(\mathcal{AUT}(m, \ell, \mathcal{T}), \mathcal{AUT}(M, \ell, \mathcal{T})) \leq 2(\ell + 1).$$

Let us introduce some additional notions used in the proof. A *partial* Σ-tree is defined the same way as a Σ-tree, except that for every leaf, its label can be either an element of Σ or the special symbol $\star$. Given a Σ-tree automaton $\mathfrak{A}$, an $\mathfrak{A}$-*initialized partial* Σ-*tree* is of the form $(\mathcal{T}, \beta)$, where $\mathcal{T}$ is a partial Σ-tree, and β is an assignment of states of $\mathfrak{A}$ to the $\star$-leaves of $\mathcal{T}$. An $\mathfrak{A}$-initialized partial Σ-tree $(\mathcal{T}, \beta)$ determines a run $\rho : V \to Q$ of $\mathfrak{A}$ in the natural way. The state $\rho(r)$ obtained at the root r of $\mathcal{T}$ is denoted by $\rho(\mathcal{T}, \beta, \mathfrak{A})$.

Let a be a vertex of a partial Σ-tree $\mathcal{T}$, such that the label of a is not $\star$. Recall that $\Sigma_1 = \Sigma \times \{0, 1\}$. In accordance with our previous notation, we let $\mathcal{T}_a$ be the partial Σ_1-tree with the same underlying tree as $\mathcal{T}$ where vertex a has label $(\sigma^{\mathcal{T}}(a), 1)$, every vertex $b \neq a$ with $\sigma^{\mathcal{T}}(b) \neq \star$ has label $(\sigma^{\mathcal{T}}(b), 0)$, and every b with $\sigma^{\mathcal{T}}(b) = \star$ has label $\star$.

Vertices c and d of a partial Σ-tree $\mathcal{T}$ are called *indistinguishable*, if for every m-state Σ_1-tree automaton $\mathfrak{A}$ and every initialization β it holds that

$$\rho(\mathcal{T}_c, \beta, \mathfrak{A}) = \rho(\mathcal{T}_d, \beta, \mathfrak{A}).$$

Let $\mathcal{T}$ be a Σ-tree. Consider a partially specified subset U of T such that every size $2(\ell+1)$ restriction of U has an extension in $\mathcal{AUT}(m, \ell, \mathcal{T})$. Remember that this means that for every size $2(\ell+1)$ restriction U' of U there is a $\Sigma_{\ell+1}$-automaton $\mathfrak{A}$ with m states and an ℓ-tuple $\bar{b} \in T^\ell$ of parameters such that $\mathfrak{A}(\mathcal{T}, \bar{b})$ is an extension of U'. We need to show that U has an extension in $\mathcal{AUT}(M, \ell, \mathcal{T})$, for some M only depending on Σ and m. Thus, we have to construct a $\Sigma_{\ell+1}$-tree automaton $\mathfrak{A}^\star$ with M states and a parameter tuple $\bar{b} \in T^\ell$ such that for every $a \in T$, if $U(a) = 1$ then $\mathfrak{A}^\star$ accepts $\mathcal{T}_{a\bar{b}}$, and if $U(a) = 0$ then $\mathfrak{A}^\star$ rejects $\mathcal{T}_{a\bar{b}}$.

We define a sequence of partial Σ-trees $\mathcal{T}_i$ for $i = 0, 1, \ldots, k$ by induction on i, starting with $\mathcal{T}_0 = \mathcal{T}$. If there are no indistinguishable vertices $c, d \in \mathcal{T}_i$ such that $U(c) = 1$, $U(d) = 0$ then the construction terminates. Otherwise, consider a minimal subtree $\mathcal{S}_i$ of $\mathcal{T}_i$ which contains vertices $U(c_i) = 1$ and $U(d_i) = 0$ that are indistinguishable *in the tree* $\mathcal{S}_i$. (Here a *subtree* is considered to be upward closed with respect to the tree-order $\preceq^{\mathcal{T}}$, i.e., its universe is a set $\{v \in T \mid t \preceq^{\mathcal{T}} v\}$ for some $t \in T$.) $\mathcal{T}_{i+1}$ is obtained by removing S_i from $\mathcal{T}_i$, and replacing it with a $\star$-labeled leaf. As we always delete at least two vertices from the current tree and replace them by at most one new leaf, the procedure terminates, and so k is well defined.

Claim: $k \leq \ell$.

To prove this claim, suppose for contradiction $k > \ell$ and consider the size $2(\ell+1)$ restriction of U to $Y = \{c_i, d_i : i = 0, \ldots, \ell\}$. By our assumption that every $2(\ell+1)$-element subset of U has an extension in $\mathcal{AUT}(m, \ell, \mathcal{T})$, there exists a $\Sigma_{\ell+1}$-automaton $\mathfrak{A}$ with at most m states and a tuple $b = (b_1, \ldots, b_\ell) \in T^\ell$ such that $\mathfrak{A}(\mathcal{T}, \bar{b})$ is an extension of Y.

By the pigeonhole principle, there is an $i, 0 \leq i \leq \ell$ such that $\mathcal{S}_i$ does not contain a $b_j, 1 \leq j \leq \ell$. Since c_i and d_i are indistinguishable in $\mathcal{S}_i$, $\mathfrak{A}$ accepts $\mathcal{T}_{c_i\bar{b}}$ if, and only if, it accepts $\mathcal{T}_{d_i\bar{b}}$. However, we have $U(c_i) = 1$ and $U(d_i) = 0$, so $\mathfrak{A}(\mathcal{T}, \bar{b})$ cannot be an extension of U. This is a contradiction, and the claim is proved.

As we mentioned on Page 5, it is useful to think of a $\Sigma_{\ell+1}$-tree automaton as being controlled by the labels of the vertices and $(\ell+1)$ pebbles placed on the tree. We call the first pebble the *variable-pebble* and the remaining pebbles the *parameter-pebbles*. We show how to construct an automaton $\mathfrak{A}^\star$ and place the ℓ parameter-pebbles in such a way that whenever the variable-pebble is on a c with $U(c) = 1$ the automaton will accept and whenever the variable-pebble is on a d with $U(d) = 0$ the automaton will reject.

The automaton $\mathfrak{A}^\star$ essentially simulates all size m automata (actually, multiple copies of all these automata) in parallel, with a certain switching mechanism at the parameter-pebbles. We only need k parameter-pebbles, which will be placed on the roots of the subtrees $\mathcal{S}_i$ constructed above.

Let us assume first that $k = 0$, i.e., there are no indistinguishable vertices $c, d \in \mathcal{T}$ such that $U(c) = 1$, $U(d) = 0$. Let $\mathfrak{A}_1, \ldots, \mathfrak{A}_N$ be a list of all Σ_1-tree automata with m states. The standard product construction, or in other words, the parallel simulation of all these automata, computes a length N vector of states at the root of $\mathcal{T}$. As there are no indistinguishable pairs, the set of vectors (called *good states*) corresponding to trees with the variable-pebble placed on a $c \in T$ (i.e., trees $\mathcal{T}_c$) with $U(c) = 1$ and the set of vectors (called *bad states*) corresponding to trees with the variable-pebble placed on a $d \in T$ with $U(d) = 0$ are disjoint. Hence the required automaton can be constructed with a suitable choice of its set of final states, without using any parameters.

Assume now that $k > 0$, and let us consider a subtree $\mathcal{S}_i$ which does not have any $\star$-leaf. Let r be the root of $\mathcal{S}_i$. Furthermore, let r^1 and r^2 be the two children of r and $\mathcal{S}^1$ and $\mathcal{S}^2$ the subtrees rooted at r^1 and r^2, respectively. By the minimality of $\mathcal{S}$, neither $\mathcal{S}^1$ nor $\mathcal{S}^2$ contains an indistinguishable pair c, d such that $U(c) = 1, U(d) = 0$. So for $j = 1, 2$ the product automaton constructed above will be able to recognize at r whether the variable pebble is placed on a $c \in S^j$ with $U(c) = 1$, or on a $d \in S^j$ with $U(d) = 0$, or whether it is not placed on an element of $\mathcal{S}^j$ at all. The parameter pebble placed on r is used to

1. send the automaton straight to the root of $\mathcal{T}$ in an accepting state if for either $j = 1$ or $j = 2$, the state at r^j is a good state (i.e., it indicates that the variable pebble is placed on a $c \in S^j$ with $U(c) = 1$),

2. send the automaton straight to the root of $\mathcal{T}$ in a rejecting state if for either $j = 1$ or $j = 2$, the state at r^j is a bad state (i.e., it indicates that the variable-pebble is placed on a $d \in S^j$ with $U(d) = 0$),

3. send the automaton straight to the root (it does not matter in which state), if for either $j = 1$ or $j = 2$, the state at r^j indicates that the variable-pebble is placed on an $e \in S^j$ but it is neither a good nor a bad state,

4. reset the automaton and continue otherwise (we will explain below what we mean by 'resetting the automaton').

To implement (1)–(3) our automaton needs two separate states, one accepting and one rejecting, that cause it to proceed upwards no matter what it reads.

Now we turn to the discussion of an $\mathcal{S}_i$ that has $\star$-leaves. By our construction, there can be at most k such leaves; to simplify the notation, we assume that indeed we have k leaves labeled $\star$.

Let r be the root of $\mathcal{S}_i$. Furthermore, let r^1 and r^2 be the two children of r and $\mathcal{S}^1$ and $\mathcal{S}^2$ the subtrees rooted at r^1 and r^2, respectively. By the minimality of $\mathcal{S}_i$, neither $\mathcal{S}^1$ nor $\mathcal{S}^2$ contains an indistinguishable pair c, d such that $U(c) = 1, U(d) = 0$. This means that for $j = 1, 2$ and for all $c, d \in S^j$ with $U(c) = 1, U(d) = 0$ there is a Σ_1-tree automaton $\mathfrak{A}$ with at most m states and an initialization β that assigns a state of $\mathfrak{A}$ to each of the k $\star$-leaves such that

$$\rho(\mathcal{S}^j_c, \beta, \mathfrak{A}) \neq \rho(\mathcal{S}^j_d, \beta, \mathfrak{A}).$$

(Recall that $\rho(\mathcal{S}^j_c, \beta, \mathfrak{A})$ denotes the state of $\mathfrak{A}$ at the root of $\mathcal{S}^j_c$ if it is run on $\mathcal{S}^j_c$ with initialization β.) So now we not only have to simulate all Σ_1-tree automata with at most m states in parallel, but actually all these automata with all m^k possible initializations of states at the k $\star$-leaves. So we actually need an even bigger product automaton. But we can construct such an automaton, and then proceed as in the case without $\star$-leaves.

It remains to explain what it means to *reset* the automaton at the root r_i of some subtree $\mathcal{S}_i$, which happens if the variable-pebble is not in this subtree. Then r_i becomes a $\star$-leaf in $\mathcal{T}_{i+1}$. Recall that $\mathfrak{A}_1, \ldots, \mathfrak{A}_N$ is a list of all Σ_1-tree automata with m states. Without loss of generality we can assume that the state set of all these automata is $\{1, \ldots, m\}$. We are running m^k copies of each of these automata in parallel. We can think of all these m-state automata as being identified by a tuple $(n, s_1, \ldots, s_k)$, where $1 \leq n \leq N$ and $1 \leq s_j \leq m$ for $1 \leq j \leq k$.

What we do at r_i if the variable-pebble is not in the subtree $\mathcal{S}_i$ is start automaton number $(n, s_1, \ldots, s_k)$ in state s_i. This guarantees that in a subtree with $\star$-leaves we indeed run a copy of each possible m-state automaton with each possible initialization. □

Let $q, \ell \geq 1$ and $\mathcal{T}$ be a Σ-tree. We let

$$\mathcal{MSO}(q, \ell, \mathcal{T}) = \bigcup_{\varphi} \mathcal{C}(\varphi, \mathcal{T}),$$

where the union is over all MSO-formulas $\varphi(x, y_1, \ldots, y_\ell)$ (with a single free variable and ℓ parameters) of quantifier-rank at most q. The following result is an immediate consequence of Theorem 32 and Lemma 9.

Corollary 33 *Let Σ be a finite alphabet. For every q and ℓ there is a Q such that for every Σ-tree $\mathcal{T}$ it holds that*

$$\mathrm{SC}(\mathcal{MSO}(q, \ell, \mathcal{T}), \mathcal{MSO}(Q, \ell, \mathcal{T})) \leq 2(\ell + 1).$$

Finally, combining this with Theorem 31, we obtain:

Corollary 34 *Let Σ be a finite alphabet. For every q and ℓ there is a Q such that for every Σ-tree $\mathcal{T}$ it holds that*

$$\mathrm{EQ}(\mathcal{MSO}(q, \ell, \mathcal{T}), \mathcal{MSO}(Q, \ell, \mathcal{T})) = O(\log|\mathcal{T}|).$$

The following simple example shows that without any restriction on the structures considered, no such result holds in general.

Example 35 Let $q = 0$, $\ell = 1$, let Q, L, d be arbitrary, and put $N = 2dL + 2$. Let $G_{L,d}$ be the $N + \binom{N}{d}$ vertex graph where for every size d subset of the first N vertices there is a distinct vertex that is connected to just these vertices. Consider a partially specified subset U that assigns 0, resp. 1, to half of the first N vertices. Then every size d restriction of U is consistent with $E(x, a)$ for some choice of the parameter a, where E is the binary adjacency relation. On the other hand, for symmetry reasons, U is not consistent with any formula having at most L parameters. Thus

$$SC(\mathcal{MSO}(0, 1, G_{L,d}), \mathcal{MSO}(Q, L, G_{L,d})) > d.$$

Note that the bound of Corollary 34 is sharp — to learn a prefix of a string $\mathcal{S}$ (which is defined as $\varphi(\mathcal{S}, b)$ for the formula $x \preceq y$ of quantifier-rank 0 and a suitable parameter b) one needs at least $\Omega(\log|\mathcal{S}|)$ queries, even if equivalence queries with arbitrary sets and membership queries of the form 'Is $x \in C$?' are allowed [26].

8. Conclusions

In this paper we presented upper bounds for the VC-dimension and the strong consistency dimension of the classes of definable sets in finite relational structures for monadic second-order logic and first-order logic. As these quantities characterize the sample complexity of PAC-learnability, respectively, the complexity of learning with equivalence queries, the bounds imply upper bounds for learning complexity in these models.

Finite upper bounds for some of the cases considered follow from previous results in model theory, but even in these cases we obtain explicit bounds, as opposed to the non-constructive previous results.

Although our bounds are explicit, the resulting learnability results are not practical in the sense that they only refer to the informational (sample, resp., query) complexity of the learning algorithms, and they do not provide efficient algorithms to find a consistent hypothesis, resp., to form the next query.

Our results are based on a new view of definability for tree automata. It is an interesting question whether one can develop computationally efficient learning algorithms for the corresponding learning problem.

There are many open problems related to the strong consistency dimension. It would be of interest to extend our results to formulas with more than one free variable and to more general structures than trees. Also, it is not clear, what the relationship between the strong consistency dimension of MSO and FO is (they may be unrelated).

Acknowledgements. We thank Michael Benedikt for clarifying the relation between our Theorem 10 and the results of [7].

References

[1] H. Aizenstein, T. Hegedűs, L. Hellerstein, and L. Pitt. Complexity-theoretic hardness results for query learning. *Computational Complexity*, 7:19–53, 1998.

[2] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[3] L. Babai and Gy. Turán. The complexity of defining a relation on a finite graph. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 33:277–288, 1987.

[4] J.L. Balcázar. The consistency dimension, compactness and query learning. In J. Flum and M. Rodriguez-Artalejo, editors, *Computer Science Logic, 13th International Workshop CSL'99*, volume 1683 of *Lecture Notes in Computer Science*, pages 2–13. Springer-Verlag, 1999.

[5] J.L. Balcázar, J. Castro, and D. Guijarro. A new abstract combinatorial dimension for exact learning via queries. In S.A. Goldman N. Cesa-Bianchi, editor, *Proceedings of the 13th Annual Conference on Computational Learning Theory (COLT 2000)*, pages 248–254. Morgan-Kaufmann, 2000.

[6] J.L. Balcázar, J. Castro, D. Guijarro, and H.U. Simon. The consistency dimension and distribution- dependent learning from queries. In O. Watanabe and T. Yokomori, editors, *Algorithmic Learning Theory, 10th International Conference, ALT '99*, volume 1720 of *Lecture Notes in Computer Science*, pages 77–92. Springer-Verlag, 1999.

[7] M. Benedikt, L. Libkin, T. Schwentick, and L. Segoufin. A model-theoretic approach to regular string relations. In *Proceedings of the 16th IEEE Symposium on Logic in Computer Science*, pages 431–440, 2001.

[8] A. Blumer, A. Ehrenfeucht, D. Haussler, and M.K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36:929–965, 1989.

[9] B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle-rewriting hypergraph grammars. *Journal of Computer and System Sciences*, 46:218–270, 1993.

[10] B. Courcelle and S. Olariu. Upper bounds to the clique-width of graphs. *Discrete Applied Mathematics*, 101:77–114, 2000.

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

[12] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Springer-Verlag, 2nd edition, 1994.

[13] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27:275–291, 2000.

[14] M. Frazier and L. Pitt. CLASSIC learning. *Machine Learning*, 25:151–193, 1996.

[15] M. Frick and M. Grohe. The complexity of first-order and monadic second-order logic revisited. In *Proceedings of the 17th IEEE Symposium on Logic in Computer Science*, pages 215–224, 2002.

[16] H. Gaifman. On local and non-local properties. In J. Stern, editor, *Proceedings of the Herbrand Symposium, Logic Colloquium '81*, pages 105–135. North Holland, 1982.

[17] M.C. Golumbic and U. Rotics. On the clique-width of some perfect graph classes. *International Journal of Foundations of Computer Science*, 11:423–443, 2000.

[18] M. Grohe. Local tree-width, excluded minors, and approximation algorithms. *Combinatorica*. To appear.

[19] M. Grohe. Generalized model-checking problems for first-order logic. In H. Reichel and A. Ferreira, editors, *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science*, volume 2010 of *Lecture Notes in Computer Science*, pages 12–26. Springer-Verlag, 2001.

[20] T. Hegedűs. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the 8th Annual Conference on Computational Learning Theory (COLT 1995)*, pages 108–117, 1995.

[21] L. Hellerstein, K. Pillaipakkamnatt, V.Raghavan, and D.Wilkins. How many queries are needed to learn? *Journal of the ACM*, 43:840–862, 1996.

[22] W. Hodges. *Model Theory*. Cambridge University Press, 1993.

[23] M.J. Kearns and U.V.Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[24] M.C. Laskowski. Vapnik-Chervonenkis classes of definable sets. *Journal of the London Mathematical Society (2)*, 45:377–384, 1992.

[25] L. Libkin. Logics with counting and local properties. *ACM Transactions on Computational Logic*, 1:33–59, 2000.

[26] W. Maass and Gy. Turán. Lower bound methods and separation results for on-line learning models. *Machine Learning*, 9:107–145, 1992.

[27] W. Maass and Gy. Turán. On learnability and predicate logic. In *Proceedings of the Bar-Ilan Symposium on the Foundations of Artificial Intelligence (BISFAI-95)*, pages 75–85, 1995.

[28] S.-H. Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming*, volume 1228 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[29] D.N. Osherson, M. Stob, and S. Weinstein. New directions in automated scientific discovery. *Information Sciences*, 57-58:217–230, 1991.

[30] K.P. Podewski and M. Ziegler. Stable graphs. *Fundamenta Mathematicae*, 100:101–107, 1978.

[31] N. Robertson and P.D. Seymour. Graph minors II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7:309–322, 1986.

[32] N. Robertson and P.D. Seymour. Graph minors V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41:92–114, 1986.

[33] N. Sauer. On the density of families of sets. *Jornal of Combinatorial Theory, Series A*, 13:145–147, 1972.

[34] S. Shelah. Stability, the f.c.p. and superstability. *Annals of Mathematical Logic*, 3:271–362, 1971.

[35] S. Shelah. A combinatorial problem: stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41:241–261, 1972.

[36] L.J. Stockmeyer and A.R. Meyer. Word problems requiring exponential time. In *Proceedings of the 5th ACM Symposium on Theory of Computing*, pages 1–9, 1973.

[37] J.W. Thatcher and J.B. Wright. Generalised finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2:57–81, 1968.

[38] I. Tsapara and Gy.Turán. Learning atomic formulas with prescribed properties. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT'98)*, pages 166–174, 1998.

[39] L. van den Dries. *Tame Topology and O-minimal Structures*. Cambridge University Press, 1998.

[40] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

Appendix: Properties of classes of structures

Figure 1 below gives an overview of the properties of classes of structures considered in this papers, amended by examples separating the different properties.

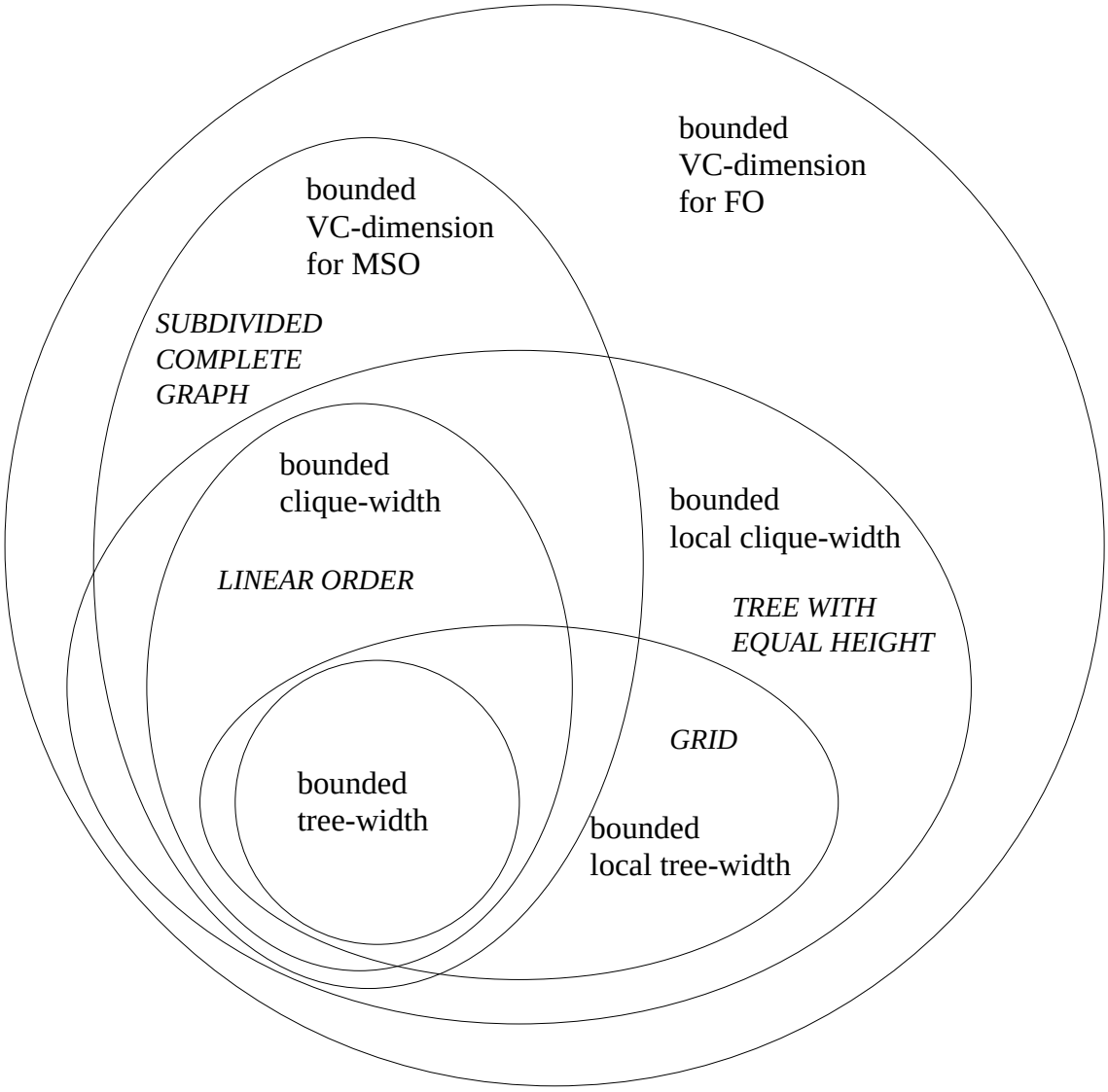

Figure 1.