Property Testing for Bounded Degree Databases

Isolde Adler
University of Leeds, School of Computing, Leeds, UK
i.m.adler@leeds.ac.uk

Frederik Harwath
Institut für Informatik, Goethe-Universität, 60054 Frankfurt, Germany
harwath@cs.uni-frankfurt.de

Abstract

Aiming at extremely efficient algorithms for big data sets, we introduce property testing of relational databases of bounded degree. Our model generalises the bounded degree model for graphs (Goldreich and Ron, STOC 1997). We prove that in this model, if the databases have bounded tree-width, then every query definable in monadic second-order logic with modulo counting is testable with a constant number of oracle queries and polylogarithmic running time. This is the first logical meta-theorem in property testing of sparse models. Furthermore, we discuss conditions for the existence of uniform and non-uniform testers.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms, Theory of computation → Logic and databases, Mathematics of computing → Graph algorithms

Keywords and phrases Logic and Databases, Property Testing, Logical Meta-theorems, Bounded Degree Model, Sublinear Algorithms

Digital Object Identifier 10.4230/LIPIcs.STACS.2018.6

Acknowledgements We would like to thank Tomáš Gavenčiak for critical discussions.

1 Introduction

As technology advances, there is an increased need for new *extremely efficient* algorithms that deal with typical computational problems on ever larger databases. Approximate Query Processing [4] addresses this by seeking to provide approximate answers to queries at a fraction of the cost of traditional query execution, thus opening the path for revealing new insights into voluminous data sets. In this spirit, we study a relaxation of Boolean Query Evaluation on relational databases of bounded degree, including performance and (probabilistic) accuracy guarantees.

The problem of Boolean Query Evaluation asks, for a Boolean query Q and a relational database $\mathcal{D}$, whether $\mathcal{D}$ satisfies Q. Relaxing this problem, we want to distinguish with high probability correctly the case that $\mathcal{D}$ satisfies Q from the case that $\mathcal{D}$ is ϵ-*far* from satisfying Q, i. e. from the case that we need to modify (add / delete) more than an ϵ-fraction of the tuples in relations of $\mathcal{D}$ to make the database satisfy Q. For $\epsilon \in (0, 1]$, an ϵ-*tester* is a probabilistic algorithm that makes this distinction by only looking at a small number of small parts of the input database. More precisely, the ϵ-tester receives the size n of the input database and has oracle access to the database. For each given element of the domain, the tester can query the oracle for tuples (in any of the relations) containing the element. The *query complexity* $q(n)$ of the ϵ-tester is the maximum number of oracle queries performed, over all input databases on n elements.

35th Symposium on Theoretical Aspects of Computer Science (STACS 2018).
Editors: Rolf Niedermeier and Brigitte Vallée; Article No. 6; pp. 6:1–6:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


In this paper we only consider databases of degree bounded by a constant d, i. e. every element participates in at most d tuples in relations, and we assume that the oracle can answer queries in constant time. Due to the degree bound, a database $\mathcal{D}$ on n elements has at most dn tuples in relations. We say $\mathcal{D}$ is ϵ-*far* from satisfying query Q, if we need to delete or insert more than ϵdn tuples in $\mathcal{D}$ to satisfy Q. Given our interest in highly efficient algorithms, we study testers with *constant* query complexity q, i. e. q is independent of n (but may depend on ϵ and the degree d). Here, we view a *Boolean query* as an isomorphism closed class of relational databases. We use the term *property* instead of Boolean query. We say that a property Q is *testable*, if for every ϵ, there is an ϵ-tester for Q with constant (oracle) query complexity.

In [2], Alon, Krivelevich, Newman and Szegedy proposed a systematic study of the testability of logically defined properties. We obtain the first such *logical meta-theorem* for property testing in the bounded degree model, which relates logical definability to time efficient uniform testability. This is the main result of our paper. It can be thought of as testability-analogue of the well-known theorem Courcelle [6], which states that each property of relational databases which is definable in *monadic second-order logic with counting* (CMSO) is decidable in linear time on relational databases of bounded tree-width. We show that each such property is testable on structures of bounded degree and bounded tree-width.

Our model extends the bounded-degree model for property-testing of graphs, introduced by Goldreich and Ron [15], and the bidirectional model for directed graphs of [3, 9]. In [21], Newman and Sohler showed that every *hyperfinite* graph property is testable. This includes properties such as planarity and minor-closed graph classes. Their testers are *non-uniform* in the number n of vertices of the input graph. (Indeed, it is not hard to come up with a property of (edgeless) graphs that is undecidable but hyperfinite, and hence testable.) We generalise this result to databases of bounded degree over a fixed finite signature. We then study conditions that allow for *time efficient uniform* testers, and we characterise both non-uniform testability and uniform testability. We introduce a combinatorial criterion which characterises non-uniform testability with constant query complexity. This criterion which we call *locality* is similar to the concept of *Hanf locality* from mathematical logic which provides a combinatorial method for proving non-definability results for first-order logic and its counting extensions (cf. [17]). It captures the intuition that testability of a property with constant query complexity means that the property is determined by the distribution of substructures of a constant radius r (r-discs) in a structure. We also introduce a concept which we call *effective locality* and which characterises uniform testability. This pinpoints that uniform testability of a property is closely related to solving a relaxation of a *realisability problem* of r-disc distributions. This problem asks for a given distribution D and a natural number n whether there exists a structure on n elements satisfying the property and where the r-discs are distributed according to D.

Techniques. Similar to the evaluation of scale independent queries in [12], our algorithms only explore a constant size sub-database before outputting the answer. For this, we explicitly require the database to have bounded degree, whereas scale-independent queries use a degree restriction implicit in the access schema.

The proof of the CMSO-result involves a number of steps. By our characterisation of uniform testability it suffices to show that all CMSO-definable properties are effectively local. To prove this, we generalise the Local-Global Theorem [21, Thm. 3.1] from graphs to databases. We then show that the realisability problem can be solved efficiently. For this, we use the fact that many-sorted spectra of CMSO formulas on bounded tree-width are

semilinear, a result by Fischer and Makowsky [13], that is based on Parikh's Theorem [22]. This allows to encode the problem as an integer linear program (ILP) with a fixed number of variables and inequalities. Using a result of Lenstra [18] for solving such ILPs we obtain polylogarithmic running time.

Let us remark that some work goes into generalising tools from property testing on graphs to relational databases. For this we could reprove the relevant results in our slightly more general setting (this is not done in this extended abstract for space constraints). A different approach would be to encode relational structures as (coloured) incidence graphs, and apply the known results for graphs. This can be done, because bounded degree, bounded tree-width and hyperfiniteness carry over to incidence graphs. Nevertheless, it involves some technicalities, in particular when simulating testing algorithms on incidence graphs. We plan to take this approach in the journal version.

Structure of the paper. Section 2 introduces the notation and general definitions used throughout the paper. Section 3 is devoted to the discussion of our model, including illustrating examples. Section 4 contains our logical meta-theorem, together with the generalisations of known results about graphs to databases. Section 5 characterises testability by our notion of locality, and uniform testability by *effective locality*. We conclude in Section 6.

Proofs. Several proofs are omitted due to space contraints for this extended abstract. We indicate this by $(*)$.

2 Preliminaries

We let $\mathbb{N}$ denote the set of natural numbers including 0. For each $n \in \mathbb{N}$ with $n \geq 1$, we let $[n] := [1, n] \cap \mathbb{N}$. A *partition* of a set A is a collection P of non-empty, pairwise disjoint subsets of A whose union is A. For an element $a \in A$ we let $P[a]$ denote the unique member of P containing a.

Relational structures. In this paper, we consider relational databases as *finite relational structures* over *finite signatures*. A *signature* is a finite set $\sigma := \{R_1, \dots, R_\ell\}$ of *relation symbols*, each of which has an *arity*, $\mathrm{ar}(R_i) \in \mathbb{N}$. We let $\mathrm{ar}(\sigma)$ denote the maximum arity of the relation symbols contained in σ. A *σ-structure* is a tuple $\mathcal{A} := (A, R_1^{\mathcal{A}}, \dots, R_\ell^{\mathcal{A}})$ where A is a finite set, called the *universe* of $\mathcal{A}$, and $R_i^{\mathcal{A}}$ is a $\mathrm{ar}(R_i)$-ary relation on A. The members of the universe are the *elements* of $\mathcal{A}$. We let $|\mathcal{A}| := n$ denote the number n of elements of $\mathcal{A}$. We assume that all structures are linearly ordered in some way or, equivalently, that the universe of a structure with n elements is $[n]$. We extend the linear order on $\mathcal{A}$ to a linear order on each relation of $\mathcal{A}$ via lexicographic ordering. We say that a σ-structure $\mathcal{B}$ is a *substructure* of another σ-structure $\mathcal{A}$, if $\mathcal{B}$ can be obtained from $\mathcal{A}$ by deleting a (possibly empty) set of elements from $\mathcal{A}$ and a (possibly empty) set of tuples from the relations of $\mathcal{A}$. Note that we have to relabel the elements of $\mathcal{B}$ after deleting elements to be sure that the universe of $\mathcal{B}$ is still an initial segment of the natural numbers. A substructure of $\mathcal{A}$ is *induced* by a set $M \subseteq A$ if it can be obtained from $\mathcal{A}$ by deleting all elements of $A \setminus M$ (and all incident tuples). We denote such a structure by $\mathcal{A}[M]$. A *property* is a class of structures that is closed under isomorphism. Indeed, we only consider isomorphism closed classes of structures. For each class C of structures, we define $\mathrm{C}|n := \{\mathcal{A} \in \mathrm{C} : |A| = n\}$. The *degree* $\deg(a)$ of an element a in a structure $\mathcal{A}$ is the total number of tuples in all relations which

contain a. We define the *degree* $\deg(\mathcal{A})$ of a structure $\mathcal{A}$ as the maximum degree of its elements. A class C of structures has *bounded degree* d if $\deg(\mathcal{A}) \leq d$ for each $\mathcal{A} \in \mathrm{C}$. A class C is of structures is *closed under removing tuples*, if for every structure $\mathcal{A} \in \mathrm{C}$, the structure $\mathcal{A}'$ obtained from $\mathcal{A}$ by deleting a tuple from some relation of $\mathcal{A}$ is also a member of C.

A class C has *bounded degree* if there exists a number d such that C has bounded degree d. We write $\mathrm{C}_{\sigma,d}$ for the class of all σ-structures of degree at most d. Most of the time, σ will be fixed and we omit it from this notation. The *Gaifman graph* of a structure $\mathcal{A}$ is the undirected graph $\mathcal{G}(\mathcal{A}) = (A, E)$, where $\{x, y\} \in E$ if x and y occur together in a tuple belonging to some relation of $\mathcal{A}$. Observe that our notion of bounded degree classes coincides with the notion where the degree of $\mathcal{A}$ is defined as the (usual graph-theoretical) maximum degree of $\mathcal{G}(\mathcal{A})$.

We transfer the usual graph theoretic (shortest path) distance and related notions (e.g. radius, connectivity) from Gaifman graphs $\mathcal{G}(\mathcal{A})$ to structures $\mathcal{A}$. A non-empty induced substructure $\mathcal{B}$ of $\mathcal{A}$ is a *connected component* of $\mathcal{A}$, if $\mathcal{G}(\mathcal{B})$ is a connected component of $\mathcal{G}(\mathcal{A})$. For each $r \in \mathbb{N}$, an *r-disc* is a pair $(\mathcal{A}, a)$ where $\mathcal{A}$ is a relational structure of radius at most r and a is a central element of $\mathcal{A}$. Two r-discs $(\mathcal{A}, a)$ and $(\mathcal{B}, b)$ are *isomorphic* (written $(\mathcal{A}, a) \cong (\mathcal{B}, b)$) if there is an isomorphism of $\mathcal{A}$ and $\mathcal{B}$ which maps a to b. An *r-type* is an $\cong$-equivalence-class of r-discs. The number of *r-types* is bounded by a constant which depends only on the signature σ and the fixed degree bound d. We write $c(r)$ to denote this constant. For a given structure $\mathcal{A}$ and an element a of $\mathcal{A}$, the *r-neighbourhood of a in $\mathcal{A}$*, written $N_r^{\mathcal{A}}(a)$ is the set of all elements with distance at most r to a. The *r-disc around a in $\mathcal{A}$* is the structure $\mathcal{D}_r^{\mathcal{A}}(a) := (\mathcal{A}[N_r^{\mathcal{A}}(a)], a)$. We say that an element a *realises* an r-disc-type τ, if $\mathcal{D}_r^{\mathcal{A}}(a) \in \tau$. For each structure $\mathcal{A}$, the *r-histogram of $\mathcal{A}$* denotes the vector $\mathrm{h}_r(\mathcal{A})$ with $c(r)$ components, indexed by the r-disc types, where the component corresponding to type τ contains the number of elements of $\mathcal{A}$ which realise τ. More generally, we call a vector $\bar{v}$ an *r-histogram with n elements* if it has $c(r)$ components and if $n = \sum_{i \leq c(r)} \bar{v}[i]$. For every class of structures C and each $r \in \mathbb{N}$, we let $\mathrm{H}_r(\mathrm{C}) := \{\mathrm{h}_r(\mathcal{A}) : \mathcal{A} \in \mathrm{C}\}$. If there exists a structure $\mathcal{A} \in \mathrm{C}$ such that $\bar{v} = \mathrm{h}_r(\mathcal{A})$, we say that $\bar{v}$ is C*-realisable*. For each r-histogram $\bar{v}$ and each r-disc $\mathcal{D}$, we let $\bar{v}[\mathcal{D}] := \bar{v}[i]$ where τ_i is the unique r-disc type with $\mathcal{D} \in \tau_i$. For an r-histogram $\bar{v}$ with n elements, the vector $\bar{v}/n$ specifies a *distribution* of r-types.

3 The Model

Algorithms with direct access. Let σ be a signature. We consider algorithms which process σ-structures of bounded degree d. An algorithm that processes $\mathcal{A}$ does not obtain an encoding of $\mathcal{A}$ as a bit string in the usual way. Instead, it has direct access to $\mathcal{A}$ using an *oracle* which answers queries about the relations of $\mathcal{A}$ in constant time. The (usual) input of the algorithm consists of auxiliary information. In our case, this will always be the size n of the universe of $\mathcal{A}$ in binary representation. The oracle accepts queries of the form (R, i, j), for $R \in \sigma$, $i \leq n$, and $j \leq \deg(\mathcal{A})$, to which it responds with the j-th tuple of the relation $R^{\mathcal{A}}$ which contains the i-th element, or with $\perp$ if there are strictly less than j tuples in $R^{\mathcal{A}}$ which contain i. The *running time* of the algorithm is defined as usual, i.e. with respect to the auxiliary input n. As common in property testing, we use a uniform cost model, i. e. we assume that all basic arithmetic operations including random sampling can be performed in constant time, regardless of the size of the numbers involved.

Distance. For two graphs G and H, both with n vertices, $\mathrm{dist}(G, H)$ denotes the minimum number of edges that have to be inserted or removed from G and H to make G and H isomorphic. For two σ-structures $\mathcal{A}, \mathcal{B}$ with the same number n of elements, $\mathrm{dist}(\mathcal{A}, \mathcal{B})$

denotes the minimum number of tuples that have to be inserted or removed from $\mathcal{A}$ and $\mathcal{B}$ to make $\mathcal{A}$ and $\mathcal{B}$ isomorphic. Let $\epsilon \in [0,1]$ and $d \in \mathbb{N}$. If $\deg(\mathcal{A}), \deg(\mathcal{B}) \leq d$, we say that $\mathcal{A}$ and $\mathcal{B}$ are ϵ-*close* (with respect to d) if $\operatorname{dist}(\mathcal{A},\mathcal{B}) \leq \epsilon dn$. Usually, the number d is fixed and will not be mentioned. If $\mathcal{A},\mathcal{B}$ are not ϵ-*close*, then they are ϵ-*far*. We say that a structure $\mathcal{A}$ is ϵ-*close* to a property P if $\mathcal{A}$ is ϵ-close to some $\mathcal{B} \in \mathrm{P}$. Otherwise, $\mathcal{A}$ is ϵ-*far* from P. Given a class of structures C, we write ϵ-close$_\mathrm{C}$(P) and ϵ-far$_\mathrm{C}$(P) for the set of σ-structures from C which are ϵ-close and ϵ-far to P, respectively. Usually, we omit C from this notation if it can be inferred from the context. Note that since P is closed under isomorphism, both ϵ-close(P) and ϵ-far(P) are closed under isomorphism as well.

Degree bounds and distances carry over from structures to their Gaifman graphs as follows.

▶ **Lemma 1** (∗). *Let σ be a signature of maximum arity $\alpha := \operatorname{ar}(\sigma)$ and let $\mathcal{A},\mathcal{B}$ be σ-structures of degree at most $d \geq 1$. Then $\deg(\mathcal{G}(\mathcal{A})) \leq d\alpha$, and $\operatorname{dist}(\mathcal{G}(\mathcal{A}),\mathcal{G}(\mathcal{B})) \leq \binom{\alpha}{2} \operatorname{dist}(\mathcal{A},\mathcal{B})$.*

▶ **Definition 2** (ϵ-tester)**.** Let σ be a signature and let $d \in \mathbb{N}$. Let C be a class of σ-structures of bounded degree d and let $\mathrm{P} \subseteq \mathrm{C}$ be a property. An ϵ-*tester for* P *on* C is a probabilistic algorithm with direct access to σ-structures. Given oracle access to a σ-structure $\mathcal{A} \in \mathrm{C}$ and given $n := |A|$ as auxiliary input, the algorithm does the following:

1. If $\mathcal{A} \in \mathrm{P}$, then the tester accepts with probability at least $\frac{2}{3}$.
2. If $\mathcal{A} \in \epsilon$-far(P), then the tester rejects with probability at least $\frac{2}{3}$.

The *query complexity* of a tester is the maximum number of oracle queries made. A property $\mathrm{P} \subseteq \mathrm{C}$ is *uniformly testable in time $f(n)$ on* C, if for each $\epsilon \in (0,1]$ there is an ϵ-tester for P which has *constant query-complexity* (i. e. independent of n) and whose running time on structures with n elements is $f(n)$. Note that this tester must work for all n.

A property P is *non-uniformly testable*, if for each n, the class $\mathrm{P}|n$ is testable on $\mathrm{C}|n$, i.e. there may be a different tester for each input size. (Note that this definition of uniform testability is non-standard. Some authors define uniform testability as being uniform with respect to ϵ.)

Our definitions subsume the bounded degree model of graph property testing. Let $\sigma := \{E\}$, where E is a binary relation symbol, and let C denote the class of all undirected graphs (viewed as symmetric, irreflexive directed graphs) of degree at most d. Then, up to a constant factor in the query- and time complexity, all results about testability in the bounded degree graph model carry over to testability in our model. In the same way, our model generalises the bidirectional model for directed graphs of bounded degree introduced in [3]. We illustrate the model by two simple examples.

▶ **Example 3** (Keys)**.** A component of a relation is a *key* if there are no two tuples in the relation which contain the same value in this component. Let $\sigma := \{R\}$ be a signature. Consider the property Key containing all σ-structures $\mathcal{A}$ where the first component of $R^\mathcal{A}$ is a key. Let $d \in \mathbb{N}$. We show that on the class $\mathrm{C}_{\sigma,d}$, Key is uniformly testable with constant running time. Let $\epsilon \in (0,1]$. Given oracle access to a σ-structure $\mathcal{A}$ on n elements, the ϵ-tester proceeds as follows.

1. Sample $\alpha := \log_{1-\epsilon} \frac{1}{3}$ elements from $[n]$ uniformly and independently.
2. For each of these elements i, perform the queries $(R,i,1),\ldots,(R,i,d)$ to obtain all tuples of $R^\mathcal{A}$ which contain i.
3. If, in the previous step, two tuples with the same elements in their first components are found, the tester rejects $\mathcal{A}$. Otherwise, $\mathcal{A}$ is accepted.

Call a tuple belonging to $R^{\mathcal{A}}$ *bad* if $R^{\mathcal{A}}$ contains another tuple with the same first component. An element is *bad* if it occurs in the first component of a bad tuple. The tester clearly accepts $\mathcal{A}$ if $\mathcal{A} \in$ KEY, i.e. it contains no bad tuples. Suppose that $\mathcal{A}$ is ϵ-far from KEY. Then there are at least ϵdn bad tuples. Since $\mathcal{A}$ has degree at most d, at least ϵn different elements occur in the first components of these bad tuples. The probability that a uniformly random element of $\mathcal{A}$ is bad is hence at least ϵ. The probability that our sample of α independently selected elements contains no bad element is $(1-\epsilon)^{\alpha} \leq \frac{1}{3}$. Hence, with probability at least $\frac{2}{3}$ the algorithm samples a bad element i. Since i is bad, there are at least two tuples which contain i and hence the algorithm rejects $\mathcal{A}$. The running time is constant.

▶ **Example 4** (Symmetry)**.** A k-ary relation is *symmetric* if for each tuple $\bar{a} := (a_1, \ldots, a_k)$ of the relation and each permutation π of $[k]$, the tuple $\pi(\bar{a}) := (a_{\pi(1)}, \ldots, a_{\pi(k)})$ is also contained in the relation. Let σ be an arbitrary signature and let $R \in \sigma$ be a k-ary relation symbol. We show that the class SYM of σ-structures which interprete R by a symmetric relation is strongly uniformly testable on the class $\mathrm{C}_{\sigma,d}$, for each $d \in \mathbb{N}$. Let $\epsilon \in (0,1]$. Given oracle access to a σ-structure $\mathcal{A}$ on n elements, the tester proceeds as follows.

1. Sample $\alpha := \log_{1-\epsilon} \frac{1}{3}$ elements uniformly and independently from the universe $[n]$.
2. For each sampled element i, perform the queries $(R,i,1),\ldots,(R,i,d)$ to obtain all (at most d) tuples $\bar{a}$ of $R^{\mathcal{A}}$ which contain i, and for each such tuple $\bar{a} = (a_1, \ldots, a_k)$ and for each permutation π of $[k]$, check whether the tuple $\pi(\bar{a})$ is also present.
3. If this is true for all α elements, accept. Otherwise reject.

The tester clearly accepts $\mathcal{A}$ if $\mathcal{A} \in$ SYM. Suppose that $\mathcal{A}$ is ϵ-far from SYM. Then there are at least ϵdn bad tuples, i. e. tuples $\bar{a} \in R^{\mathcal{A}}$ such that $\pi(\bar{a}) \notin R^{\mathcal{A}}$ for some permutation π. Since each element of the universe is in at most d tuples, there are at least ϵn different elements that are contained in bad tuples. Hence the probability that a random element is in a bad tuple is at least $\epsilon n/n = \epsilon$. Hence the probability that none of the α sampled elements is in a bad tuple is at most $(1-\epsilon)^{\alpha} = 1/3$. Hence the algorithm accepts with probability at least $2/3$. The tester has query complexity $d \log_{1-\epsilon} \frac{1}{3}$ and constant running time.

4 Monadic second-order logic and bounded tree-width

We are interested in identifying general conditions which ensure the time efficient *uniform testability* of a wide range of properties on relational structures. In this section we consider properties which are definable by sentences of *monadic second order logic with counting* (CMSO) on classes of structures of bounded degree and bounded tree-width.

Before we state the main theorem of this section, we briefly introduce logic. We use the notation which is usual in finite model theory (cf. e.g. [19] for a general overview and [7] for an overview regarding CMSO and the notion of tree-width). Fix a signature σ. An *atomic formula*, is a formula of the form $x = y$ or $R(x_1, \ldots, x_r)$, where $R \in \sigma$ is an r-ary relation symbol and $x, y, x_1, \ldots, x_r$ 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* (MSO) 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* (denoted by small letters $x, y, z, x_1, \ldots$), which are interpreted by elements of a structure, and *set variables* (denoted by upper-case letters $X, Y, Z, X_1, \ldots$), which are interpreted by subsets of the universe of a structure. In addition to the atomic

first-order formulas, in MSO we also have atoms Xx saying that x is an element of set X. Furthermore, we have existential and universal quantification over both individual and set variables. CMSO extends MSO by first-order modular counting quantifiers $\exists^m$, for each integer m, where $\exists^m \varphi$ is true in a structure if the number of its elements for which φ is satisfied is divisible by m. 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 $\forall v$. A *sentence* is a formula without free variables. For a structure $\mathcal{A}$ and a sentence φ we write $\mathcal{A} \models \varphi$ to denote that $\mathcal{A}$ satisfies φ. Detailed introductions can be found in [10, 19].

By $\textsc{Mod}(\varphi)$ we denote the class of all structures satisfying φ. For an arbitrary formula φ and an assignment a in $\mathcal{A}$ to the free variables of φ, we write $(\mathcal{A}, a) \models \varphi$ to denote that $\mathcal{A}$ satisfies φ if the free variables are interpreted according to a. We say that a property P is *definable* in logic $\mathcal{L}$, if there is a sentence φ of $\mathcal{L}$ with $\mathrm{P} = \textsc{Mod}(\varphi)$. For a class C of structures, we say that $\textsc{Mod}(\varphi) \cap \mathrm{C}$ is the property *defined by* φ *on* C.

Recall that we have assumed that the universes of structures are initial segments of the natural numbers. This was used for the definitions surrounding testability. We stress that the linear orders on structures are *not* available in logical formulas.

▶ **Proviso.** *Let $d \in \mathbb{N}$, and fix a finite relational signature σ. From now on, all structures are σ-structures and have degree at most d, unless stated otherwise. Moreover, we let $\alpha := \mathrm{ar}(\sigma)$ and $\mathrm{C} \subseteq \mathrm{C}_{\sigma,d}$.*

We let $\mathrm{C}_t^{\mathrm{tw}}$ denote the class of all σ-structures of degree at most d and tree-width at most t. The main goal of this section is a proof of the following theorem.

▶ **Theorem 5.** *Each property* P *which is CMSO-definable on* $\mathrm{C}_t^{\mathrm{tw}}$ *is uniformly testable on* $\mathrm{C}_t^{\mathrm{tw}}$ *with polylogarithmic time complexity.*

For the proof, we introduce a notion of *locality*, which is based on the distributions of r-discs. In the next section, we will show that locality characterises non-uniform testability. Newman and Sohler's results [21] show that properties of hyperfinite graphs are non-uniformly testable and local. We generalise these to relational structures, and we use the fact that every class of relational structures of bounded tree-width is hyperfinite. This already implies non-uniform testability. We then use semilinearity of $H_r(\mathrm{P})$ for MSO-definable properties on bounded tree-width and a restricted form of ILP to establish polylogarithmic running time.

Since Hanf's paper [16], it is known that properties which are definable by formulas of first-order logic are *local*, in the sense that whether or not a structure has a first-order definable property depends only on the r-discs present in the structure. This is made precise by several notions of locality such as *Hanf locality* and *Gaifman locality* (cf. [17]). These yield a unified combinatorial method for showing that certain properties of sparse structures are not first-order definable. Our notion of (approximate) locality is of similar spirit.

We introduce some notation. We identify each r-histogram vector $\bar{v}$ with n components with a structure $\mathcal{A}$ on n elements over a signature $\sigma_r := \{P_1, \ldots, P_{c(r)}\}$ where each relation symbol P_i is unary. The unary relations of $\mathcal{A}$ are pairwise disjoint sets such that $|P_i^{\bar{v}}| := \bar{v}[i]$, for each $i \leq c(r)$. In this way, we can transfer all definitions for structures (e.g., distance, testability) to r-histograms. In particular, note that for all r-histograms $\bar{u}, \bar{v}$ with the same number n of elements, $\bar{u}$ and $\bar{v}$ are ϵ-close iff $\mathrm{dist}(\bar{u}, \bar{v}) \leq \epsilon n$, because histogram vectors are structures of degree $d = 1$. Recall that the ℓ_1-norm of a vector $\bar{v}$ on ℓ components is defined as $\|\bar{u}\|_1 := \sum_{i=1}^{\ell} |\bar{v}[i]|$.

▶ **Lemma 6** (∗)**.** $\mathrm{dist}(\bar{u}, \bar{v}) = \|\bar{u} - \bar{v}\|_1$*, for all r-histograms $\bar{u}, \bar{v}$ with the same number of elements.*

▶ **Definition 7** (Locality)**.** Let $\epsilon \in (0,1]$. A property $\mathrm{P} \subseteq \mathrm{C}$ is *ϵ-local* on C if there exist $r := r(\epsilon) \in \mathbb{N}$ and $\lambda := \lambda(\epsilon) \in (0,1]$ such that for each $\mathcal{A} \in \mathrm{P}$ and $\mathcal{B} \in \mathrm{C}$ with the same number of elements, if $\mathrm{h}_r(\mathcal{A})$ is λ-close to $\mathrm{h}_r(\mathcal{B})$, then $\mathcal{B} \in \epsilon$-close(P).

We call the parameters r and λ the *locality radius* and the *disc proximity* of P for ϵ, respectively. A property is *local* if it is ϵ-local for each $\epsilon \in (0,1]$.

The next example illustrates that locality and testability can indeed be established by very similar arguments.

▶ **Example 8** (Key is local)**.** Recall the definitions of Example 3. We show that Key is local on $\mathrm{C}_{\sigma,d}$. Let $\epsilon \in (0,1]$, let $r := 1$ and $\lambda := \epsilon$. Consider structures $\mathcal{A} \in$ Key and $\mathcal{B} \in \mathrm{C}$ with n elements each and suppose that $\mathrm{h}_r(\mathcal{A})$ and $\mathrm{h}_r(\mathcal{B})$ are λ-close. Each bad tuple of $R^{\mathcal{A}}$ belongs to the 1-disc of its first component. Since $\mathcal{A} \in$ Key, the relation $R^{\mathcal{A}}$ contains no bad tuples, and hence $\mathrm{h}_1(\mathcal{A})[\mathcal{D}] = 0$ for each 1-disc $\mathcal{D}$ such that $R^{\mathcal{D}}$ contains two tuples with the same first component. Since $\mathrm{h}_r(\mathcal{A})$ and $\mathrm{h}_r(\mathcal{B})$ are ϵ-close, the number of elements whose 1-discs contain bad tuples in $\mathcal{B}$ is $\leq \epsilon n$ and each such element is contained in $\leq d$ bad tuples. With $\leq \epsilon d n$ tuple deletions, we obtain a structure $\mathcal{B}' \in$ Key from $\mathcal{B}$. Hence, $\mathcal{B} \in \epsilon$-close(Key).

We now generalise the results of Newman and Sohler [21] to properties on arbitrary hyperfinite classes of relational structures of bounded degree. We begin with some definitions.

A substructure $\mathcal{P}$ of a structure $\mathcal{A}$ on n elements is a *k-partition* of $\mathcal{A}$, if $\mathcal{P}$ and $\mathcal{A}$ have the same universe (i. e. if $P = A$), every connected component of $\mathcal{P}$ contains at most k elements, and for each element $a \in A$, the component $\mathcal{P}[a]$ of a in $\mathcal{P}$ is the substructure of $\mathcal{A}$ induced by its universe $P[a] \subseteq A$. If, furthermore, $\mathrm{dist}(\mathcal{A}, \mathcal{P}) \leq \epsilon n$, we say that $\mathcal{P}$ is a (ϵ, k)-partition of $\mathcal{A}$. A class of structures $\mathrm{C} \subseteq \mathrm{D}$ is *ρ-hyperfinite* on D if for each $\epsilon \in (0,1]$ and each structure $\mathcal{A} \in \mathrm{C}$ there exists a $(\epsilon, \rho(\epsilon))$-partition $\mathcal{P}$ of $\mathcal{A}$ such that $\mathcal{P} \in \mathrm{D}$. We call C *hyperfinite* on D if there exists a function ρ for which C is ρ-hyperfinite on D. This definition generalises the notion of hyperfinite graph classes to general structures.

The proof of Theorem 5 makes use of the following theorem, which can be seen as the generalisation of [21, Theorem 3.1] from graphs to structures. After generalising all ingredients from graphs to relational structures, the proof of Theorem 9 can be put together as in [21, Theorem 3.1].

▶ **Theorem 9** (∗)(Local-Global Theorem)**.** *Let* C *be closed under removing tuples. If* $\mathrm{P} \subseteq \mathrm{C}$ *is hyperfinite on* C*, then* P *is local on* C*.*

We want to approximate the histogram vector of a structure that comes from a hyperfinite class of structures of bounded degree. For this we will make use of Lemma 5.1 in [21], which allows us to approximate the distribution of the r-discs of a graph by looking at a constant number of elements. This lemma easily translates to structures as follows. We write $\mathrm{EstimateFrequencies}_{r,s}$ to denote an algorithm that, given access to a σ-structure $\mathcal{A}$ of degree at most d, samples s elements in A uniformly and independently and explores their r-discs. The algorithm returns the distribution vector $\bar{v}$ of the r-disc-types of this sample.

▶ **Lemma 10.** *Let* $\lambda \in (0,1)$*,* $r \in \mathbb{N}$*. If* $s \geq \frac{c(r)^2}{\lambda^2} \cdot \ln(c(r) + 40)$*, with probability at least* 19/20 *the vector* $\bar{v}$ *returned by* $\mathrm{EstimateFrequencies}_{r,s}$ *on input* $\mathcal{A}$ *satisfies* $\|v - \mathrm{h}_r(\mathcal{A})/|\mathcal{A}|\,\|_1 \leq \lambda$*.*

Our approach to the proof of Theorem 5 can be summarised as follows. It is known that each class of graphs of bounded tree-width (and, more generally, any class of graphs which is minor-closed, cf. [21]) is hyperfinite. From this, it follows that each property P is hyperfinite

on $\mathrm{C}_t^{\mathrm{tw}}$. Hence, by Theorem 9, P is local on $\mathrm{C}_t^{\mathrm{tw}}$. Our aim is to show that a CMSO-definable property P is not only local, but also uniformly testable in polylogarithmic time.

To this end, we study the structure of the sets $\mathrm{H}_r(\mathrm{P})$ for CMSO-definable properties and arbitrary values of r. Using known results from logic, we show that these sets have a particularly simple shape (i.e., they are *semilinear*) if we consider only structures of bounded tree-width. Using this and a result about the complexity of integer linear programming (ILP) with a bounded number of constraints and variables from [11], we can show that there is an algorithm which, on input of an r-histogram on n elements *decides* in polylogarithmic time if $\bar{v}$ is λ-close to the r-histogram of a structure on n elements which belongs to P. This algorithm then, in particular, establishes the testability of $\mathrm{H}_r(\mathrm{P})$, finishing the proof.

The first ingredient to our proof is the following lemma, which carries over from graphs to structures.

▶ **Lemma 11** (∗). *Each property* $\mathrm{P} \subseteq \mathrm{C}_t^{\mathrm{tw}}$ *is hyperfinite on* $\mathrm{C}_t^{\mathrm{tw}}$.

Next, we consider the structure of $\mathrm{H}_r(\mathrm{P})$. For this, we need the following definition.

▶ **Definition 12** (semilinear sets). A set is *semilinear* if it is a finite union of linear sets. A set $M \subseteq \mathbb{N}^c$ is *linear* if $M = \{\bar{v}_0 + a_1\bar{v}_1 + \cdots + a_k\bar{v}_k : a_1, \ldots, a_k \in \mathbb{N}\}$, for $\bar{v}_0, \ldots, \bar{v}_k \in \mathbb{N}^c$.

▶ **Lemma 13.** *For each* $r \in \mathbb{N}$ *and each property* $\mathrm{P} \subseteq \mathrm{C}_t^{\mathrm{tw}}$ *which is CMSO-definable on* $\mathrm{C}_t^{\mathrm{tw}}$, *the set* $\mathrm{H}_r(P)$ *is semilinear.*

Lemma 13 is a corollary to a result of [13] about *many-sorted spectra* of CMSO-sentences.

▶ **Definition 14** (many-sorted spectrum). For every signature σ and each $\ell \in \mathbb{N}$, we let $\sigma_\ell := \sigma \cup \{P_1, \ldots, P_\ell\}$ where $P_1, \ldots P_\ell$ are unary relation symbols which do not occur in σ. A σ_ℓ-structure $\mathcal{A}$ is *ℓ-sorted* if $P_1^{\mathcal{A}}, \ldots, P_\ell^{\mathcal{A}}$ is a partition of A, i.e. $P_1 \cup \cdots \cup P_\ell = A$ and $P_i^{\mathcal{A}} \cap P_j^{\mathcal{A}} = \emptyset$ for all $1 \leq i < j \leq \ell$. We let $n(\mathcal{A}) := (|P_1^{\mathcal{A}}|, \ldots, |P_\ell^{\mathcal{A}}|)$. The *many-sorted spectrum* of a CMSO$[\sigma_\ell]$-sentence φ is the set

$$\mathrm{spec}(\varphi) := \{n(\mathcal{A}) : \mathcal{A} \text{ is a finite } \sigma_\ell\text{-structure}, \mathcal{A} \models \varphi\}.$$

▶ **Theorem 15** ([13]). *If the class defined by a CMSO$[\sigma_\ell]$-sentence φ on the class of all finite σ_ℓ-structures has bounded tree-width, then* $\mathrm{spec}(\varphi)$ *is semilinear.*

Now we can prove our lemma.

Proof of Lemma 13. Let φ be a CMSO$[\sigma]$-sentence defining P on $\mathrm{C}_t^{\mathrm{tw}}$. Let $\tau_1, \ldots, \tau_{c(r)}$ be an enumeration of all r-types according to the fixed ordered on r-types that was used in the definition of $\mathrm{H}_r(\mathrm{P})$. For each $1 \leq i \leq c(r)$, let $\psi_i(x)$ be an FO$[\sigma]$-formula such that for each $\mathcal{A} \in \mathrm{C}_t^{\mathrm{tw}}$ and $a \in A$, $(\mathcal{A}, a) \models \psi_i(x)$ if $(\mathcal{A}, a) \cong \tau_i$. There is an MSO-sentence $\psi_{\mathrm{C}_t^{\mathrm{tw}}}$ which defines the class $\mathrm{C}_t^{\mathrm{tw}}$ on the class of all finite σ-structures (using the forbidden minors for tree-width $\leq t$, see e. g. [8]). Consider the sentence

$$\xi \;:=\; \varphi \wedge \psi_{\mathrm{C}_t^{\mathrm{tw}}} \;\wedge\; \forall x \bigwedge_{1 \leq i \leq c(r)} \big(P_i(x) \leftrightarrow \psi_i(x)\big).$$

Note that $\mathrm{spec}(\xi) = \mathrm{H}_r(\mathrm{P})$ and that all models of ξ have tree-width at most t for some $p := p(t)$. By Theorem 15, we obtain that $\mathrm{H}_r(\mathrm{P})$ is semilinear. ◀

To finish the proof of Theorem 5 it remains to link the semilinearity of $\mathrm{H}_r(\mathrm{P})$ to the testability of P. The following lemma shows that semilinearity of $\mathrm{H}_r(\mathrm{P})$ implies decidability of ϵ-close$(\mathrm{H}_r(\mathrm{P}))$ with polylogarithmic running time.

▶ **Lemma 16** (∗) (Approximate realisability). *Let $r \in \mathbb{N}$ and $\epsilon \in \mathbb{Q} \cap (0,1]$, and let $M \subseteq \mathbb{N}^{c(r)}$ be a semilinear set of r-histograms. Then there is an algorithm that, given an r-histogram $\bar{u} \in \mathbb{N}^{c(r)}$ on n elements (in binary encoding), decides whether $\bar{u} \in \epsilon\text{-close}(M)$ in time polylogarithmic in n.*

For the proof of Lemma 16, we phrase the conditions for belonging to ϵ-close(M) as an ILP with a constant number of variables and constraints. Using results of [18], [11], we obtain the desired running time. It remains to prove the following lemma.

Proof of Theorem 5. Let P be a property defined by some fixed CMSO-formula on $\mathrm{C}_t^{\mathrm{tw}}$. Given ϵ, we construct an ϵ-tester for P for inputs $\mathcal{A} \in \mathrm{C}_t^{\mathrm{tw}}$ on n elements. By Lemma 11, $\mathrm{C}_t^{\mathrm{tw}}$ is hyperfinite, and hence, by Theorem 9, P is local on $\mathrm{C}_t^{\mathrm{tw}}$. Let $r = r(\epsilon)$ be the locality radius and $\lambda = \lambda(\epsilon)$ the disc proximity of P for ϵ, as in the definition of locality. Pick a sample size s for r and $\lambda/2$ as required for the algorithm $\mathrm{EstimateFrequencies}_{r,s}$ of Lemma 10, and run the algorithm to obtain an approximation $\bar{v}$ of the frequency vector of $\mathcal{A}$ with high probability. Accept, if $n \cdot \bar{v}$ is $\lambda/2$-close to $H_r(\mathrm{P})$, and reject otherwise (using the algorithm of Lemma 16). It is easy to see that the algorithm is correct. Since $\mathrm{EstimateFrequencies}_{r,s}$ runs in constant time, and the algorithm of Lemma 16 runs in time polylogarithmic in n, we obtain uniform testability with polylogarithmic running time. ◀

We remark that the vectors spanning the semilinear set $\mathrm{H}_r(\mathrm{P})$ (in the proof of Theorem 5) can be computed from the CMSO-definition of P. This is implicit in [13].

The same argument as in the proof of Theorem 5 can be used to show the following lemma.

▶ **Lemma 17** (∗). *If a property $\mathrm{P} \subseteq \mathrm{C}$ is local and ϵ-close$_{\mathrm{H}_r(\mathrm{C})}(\mathrm{H}_r(\mathrm{P}))$ is decidable in time* polylog(n)*, for each $\epsilon \in (0,1]$ and $r \in \mathbb{N}$, then P is uniformly testable on C in time* polylog(n)*.*

For the proof of Lemma 17, we first approximate the distribution of r-types by sampling. Then we accept if this distribution is sufficiently close to the distribution of a structure in P on the same number of elements as the input structure.

5 Locality and testability

In this section we characterise non-uniform testability by locality. Inspired by the proof for uniform testability of MSO on bounded tree-width, we introduce the notion of *effective locality*, which adds the requirement that a certain realisability problem for the neighbourhood distributions be solvable. We use effective locality to characterise uniform testabilty (on decidable classes). We believe that the characterisations are interesting, as they provide a purely structural criterion for testability and non-testability. While it is implicitely clear that non-uniform testability 'only depends on the local neighbourhoods', to our knowledge this has not been cast into a characerisation in the literature so far. Uniform testability has not been characterised before. Furthermore, they explain the role of uniformity which has been brought up by the general results on non-uniform testability of Newman, Sohler [21]. In this section, with a few exceptions, we disregard the running times of testers since we are interested in the structural properties of testability.

The first main theorem of this section shows that non-uniform testability is equivalent to locality.

▶ **Theorem 18** (Locality). *Then for every property $\mathrm{P} \subseteq \mathrm{C}$, P is non-uniformly testable on C if, and only if,* P *is local on* C*.*

We mention that from Theorems 18 and 9 it follows that every property of hyperfinite databases is testable. This is a generalisation of [21].

▶ **Corollary 19.** *Let* C *to be closed under removing tuples. If* C *is hyperfinite, then every property* $\mathrm{P} \subseteq \mathrm{C}$ *is non-uniformly testable on* C*.*

For example, Theorem 18 can be used for purely graph-theoretic proofs of non-testability (e.g. in the proof that bipartiteness is not testable with a constant number of queries [15]). Note that locality of a property P is not enough to ensure *uniform testability*. This can be easily shown since the halting problem for turing machines with empty input can be trivially encoded as a local property of graphs of degree 0. Intuitively, we would like to use the locality of P to construct a tester for P as follows: on input of a structure $\mathcal{A}$, (1) approximate the distribution of the r-types by random sampling and (2) accept $\mathcal{A}$ if this distribution is sufficiently close to the distribution of some structure from P. The notion of locality does not guarantee that step (2) can be implemented effectively, which motivates introducing *effective locality*.

We now introduce *effective locality*. Recall from above that our notion of testability applies, in particular, to properties of r-histograms which we treat as structures of degree 1.

▶ **Definition 20.** $\mathrm{P} \subseteq \mathrm{C}$ is *effectively local* on C if it is local on C and for each $\epsilon \in (0,1)$ and for the corresponding locality radius $r := r(\epsilon)$, the problem $\mathrm{H}_r(\mathrm{P})$ is uniformly testable on $\mathrm{H}_r(\mathrm{C})$. If the running time of the tester is $T(n)$, we say that P is $T(n)$*-effectively local*.

The *realisability problem* (cf. e. g. [1]) for a graph parameter f which maps graphs G to $f(G) \in \mathbb{N}^k$ and for a class C of graphs is the decision problem which, on input of a vector $\bar{v} \in \mathbb{N}^k$, asks if there exists a graph $G \in \mathrm{C}$ such that $f(G) = \bar{v}$. Hence, the problem $\mathrm{H}_r(\mathrm{P})$ can be viewed as a *realisability problem* for structures.

We show that effective locality characterises uniform testability. In the following theorem, we need the notion of a *promise problem*. Following e.g. [14], we define this as a pair of disjoint languages $(L_{\mathrm{YES}}, L_{\mathrm{NO}})$ of binary strings. We say that $(L_{\mathrm{YES}}, L_{\mathrm{NO}})$ is *solvable* if there exists an algorithm which accepts all inputs which belong to L_{YES} and rejects all inputs which belong to L_{NO}. Note that a brute-force derandomisation of an ϵ-tester for a property P yields a deterministic algorithm which solves the promise problem $(\mathrm{P}, \epsilon\text{-far}(\mathrm{P}))$. From a conceptual point of view, disregarding running times, it is more convenient to consider the promise problem.

The second main theorem of this section characterises uniform testability.

▶ **Theorem 21** (Effective Locality)**.** *Let* C *be a decidable class of structures, that is closed under removing tuples. For each property* $\mathrm{P} \subseteq \mathrm{C}$*, the following statements are equivalent.*

1. P *is uniformly testable on* C*.*
2. P *is effectively local on* C*.*
3. P *is local and the promise problem* $(\mathrm{P}, \epsilon\text{-far}(\mathrm{P}))$ *is solvable for each* $\epsilon \in (0,1]$*.*

We break the proofs of Theorem 18 and Theorem 21 into several lemmas.

▶ **Lemma 22** (∗)**.** *Let* C *be a decidable class of* σ*-structures. Let* $\mathrm{P} \subseteq \mathrm{C}$ *be a property such that the promise problem* $(\mathrm{P}, \epsilon\text{-far}(\mathrm{P}))$ *is solvable for each* $\epsilon \in (0,1]$*. Then the promise problem* $\big(\mathrm{H}_r(\mathrm{P}), \lambda\text{-far}(\mathrm{H}_r(\mathrm{P}))\big)$ *is solvable for each* $r \in \mathbb{N}$ *and* $\lambda \in (0,1]$*.*

▶ **Lemma 23** (∗)**.** *Let* C *be a decidable class of* σ*-structures. Let* $\mathrm{P} \subseteq \mathrm{C}$ *be a local property such that the promise problem* $\big(\mathrm{H}_r(\mathrm{P}), \lambda\text{-far}(\mathrm{H}_r(\mathrm{P}))\big)$ *is solvable for each* $r \in \mathbb{N}$ *and* $\lambda \in (0,1]$*, then* P *is uniformly testable.*

Without the solvability of the promise problem, we obtain non-uniform testability instead of uniform testability.

Now we prove the implication from effective locality to uniform testability of Theorem 21. The following lemma is stronger than what is needed here, because it also takes the running time of the tester into account. It can be seen as a generalisation of Lemma 17.

▶ **Lemma 24** (∗). *If a property* $\mathrm{P} \subseteq \mathrm{C}$ *is effectively* $\mathrm{polylog}(n)$*-local on* C*, then it is uniformly testable on* C *in time* $\mathrm{polylog}(n)$*.*

▶ **Corollary 25** (∗). *If a property* P *is local on* C*, then it is non-uniformly testable on* C*.*

Proof sketch. Since $\mathrm{P}|n$ is finite and local, it is effectively local and the result follows from Lemma 24. ◀

▶ **Lemma 26** (∗). *If a property* $\mathrm{P} \subseteq \mathrm{C}$ *is non-uniformly testable on* C*, then* P *is local on* C*.*

Proof idea. It is intuitively clear that a tester with constant query complexity can only inspect discs of a constant radius r. This is made precise by the Canonical Tester Lemma which was proved by Czumaj, Peng, Sohler in the context of property testing for directed graphs [9, Lemma 3.1]. This lemma extends to relational structures. It can be shown that, on structures with sufficiently close r-histograms, a tester for P will see the same r-discs with high probability. This is done in a similar way as in the proof of the "local versus global graph structure"-theorem of Newman, Sohler [21, Theorem 3.1]. Hence, the tester will not be able to distinguish between structures with close r-histograms, so if one of the structures is in P, the other has to be in ϵ-close(P). This yields locality of P. ◀

We can now finish the proofs of both characterisation theorems.

Proof of Theorem 18. The theorem follows from Corollary 25 and Lemma 26. ◀

Proof of Theorem 21. The implication from statement 1 (uniform testability) to statement 3 (locality and solvability of $(\mathrm{P}, \epsilon\text{-far}(\mathrm{P}))$) follows from Theorem 18 (locality) and by derandomisation (solvability). The implication from 3 to 2 (testability of $\mathrm{H}_r(\mathrm{P})$) is established by Lemma 22. The implication from 2 to 1 is established by Lemma 23. ◀

6 Conclusion

We introduced property testing for relational databases of bounded degree. Our main result is a logical meta-theorem proving testability of CMSO with constant query complexity and polylogarithmic running time for databases of bounded tree-width, and we provide characterisations of testability and uniform testability in the model. Our tester for CMSO has two-sided error (because it samples the distribution of the r-discs), and it would be interesting to know if a one-sided error can be achieved.

Since monadic second-order logic on words characterises the regular languages, Theorem 5 shows in particular, that regular languages are testable with a constant number of queries in our model. In [2] testability of regular languages was already shown for a more restrictive model, based on Hamming distance. Similarly, our result implies that regular (ranked) tree languages are testable with a constant number of queries. In [20], testability of tree languages was shown in a different model, using tree edit distance with an additional operation called *moves*. The question of [5] (explicitly stated in [20]) whether regular tree languages with Hamming distance are testable, remains open.

A further interesting question is whether all properties definable in first-order logic are (uniformly) testable is left open, and we are currently working on this. Furthermore, obtaining a logical characterisation of the (uniformly) testable properties is interesting and challenging open problem. It would also be interesting to determine the precise relation between our notion of locality and Hanf-locality. Being reminiscent of realisability of degree sequences of graphs, realisability of histograms seems worth studying in more detail.

References

1 Martin Aigner and Eberhard Triesch. Realizability and uniqueness in graphs. *Discrete Mathematics*, 136(1-3):3–20, 1994. doi:10.1016/0012-365X(94)00104-Q.

2 Noga Alon, Michael Krivelevich, Ilan Newman, and Mario Szegedy. Regular languages are testable with a constant number of queries. *SIAM J. Comput.*, 30(6):1842–1862, 2000. doi:10.1137/S0097539700366528.

3 Michael A. Bender and Dana Ron. Testing properties of directed graphs: acyclicity and connectivity. *Random Struct. Algorithms*, 20(2):184–205, 2002. doi:10.1002/rsa.10023.

4 Surajit Chaudhuri, Bolin Ding, and Srikanth Kandula. Approximate query processing: No silver bullet. In Semih Salihoglu, Wenchao Zhou, Rada Chirkova, Jun Yang, and Dan Suciu, editors, *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 511–519. ACM, 2017. doi:10.1145/3035918.3056097.

5 Hana Chockler and Orna Kupferman. w-regular languages are testable with a constant number of queries. *Theor. Comput. Sci.*, 329(1-3):71–92, 2004. doi:10.1016/j.tcs.2004.08.004.

6 B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science (Vol. B)*, pages 193–242. MIT Press, Cambridge, MA, USA, 1990. URL: http://dl.acm.org/citation.cfm?id=114891.114896.

7 B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2012.

8 B. Courcelle and J. Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2012.

9 Artur Czumaj, Pan Peng, and Christian Sohler. Relating two property testing models for bounded degree directed graphs. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1033–1045. ACM, 2016. doi:10.1145/2897518.2897575.

10 H.-D. Ebbinghaus and J. Flum. *Finite model theory*. Springer, 2005.

11 Friedrich Eisenbrand. Fast integer programming in fixed dimension. In Giuseppe Di Battista and Uri Zwick, editors, *Algorithms - ESA 2003, 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003, Proceedings*, volume 2832 of *Lecture Notes in Computer Science*, pages 196–207. Springer, 2003. doi:10.1007/978-3-540-39658-1_20.

12 Wenfei Fan, Floris Geerts, and Leonid Libkin. On scale independence for querying big data. In Richard Hull and Martin Grohe, editors, *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'14, Snowbird, UT, USA, June 22-27, 2014*, pages 51–62. ACM, 2014. doi:10.1145/2594538.2594551.

13 Eldar Fischer and Johann A. Makowsky. On spectra of sentences of monadic second order logic with counting. *J. Symb. Log.*, 69(3):617–640, 2004. doi:10.2178/jsl/1096901758.

14 Oded Goldreich. On promise problems: A survey. In Oded Goldreich, Arnold L. Rosenberg, and Alan L. Selman, editors, *Theoretical Computer Science, Essays in Memory of Shimon Even*, volume 3895 of *Lecture Notes in Computer Science*, pages 254–290. Springer, 2006. doi:10.1007/11685654_12.

15 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. doi:10.1007/s00453-001-0078-7.

16 W. Hanf. *The Theory of Models*, chapter Model-theoretic methods in the study of elementary logic, pages 132–145. North Holland, 1965.

17 L. Hella, L. Libkin, and J. Nurmonen. Notions of locality and their logical characterizations over finite models. *Journal of Symbolic Logic*, 64(4):1751–1773, 1999.

18 H. W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research.*, 8(4):538–548, 1983.

19 L. Libkin. *Elements Of Finite Model Theory (Texts in Theoretical Computer Science. An Eatcs Series)*. Springer, 2004.

20 Frédéric Magniez and Michel de Rougemont. Property testing of regular tree languages. *Algorithmica*, 49(2):127–146, 2007. doi:10.1007/s00453-007-9028-3.

21 I. Newman and C. Sohler. Every property of hyperfinite graphs is testable. *SIAM Journal on Computing*, 42(3):1095–1112, 2013. Conference version published at STOC 2011.

22 Rohit Parikh. On context-free languages. *J. ACM*, 13(4):570–581, 1966. doi:10.1145/321356.321364.